GOCE dynamical analysis and Drag Free Mode control

GOCE dynamical analysis and Drag Free Mode control

Willem-Jan Evers

DCT Report 2004.44

Graduate professor: Prof. dr. ir. Maarten Steinbuch
ESA/ESTEC supervisor: ir. Denis Fertin

European Space Agency (ESA)
European Space Research and Technology Centre (ESTEC)
Attitude, Navigation and Control Analysis Section (TOS-ECN)

In cooperation with:

Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology

April 2004

Samenvatting

GOCE, wat staat voor zwaartekracht en oceaan circulatie verkenner, zal omstreeks 2006 worden gelanceerd en zal met hoge nauwkeurigheid en resolutie het zwaartekrachtsveld van de aarde gaan meten.
Voor die hoge nauwkeurigheid worden er echter ook hoge eisen gesteld aan de regeling van de satelliet. Eisen die op dit moment nog niet volledig gehaal worden door de ontworpen regelaars.
In dit verslag zal het ontwerp van een nieuwe regelaar worden beschreven, welke als doel heeft een beter regel resultaat als de oude regelaars te verkrijgen, maar wel robuust is en makkelijk te implementeren, waarbij de variërende regelbaarheid van het systeem wordt meegenomen tijdens het ontwerp proces.
Om dit te bereiken is allereerst het dynamisch model van de satelliet afgeleid. Daarnaast is er ook nog gekeken naar de verstoringskrachten, waarbij het mogelijk bleek om de gravitatie torque in het state space model van de dynamica van de sateliet te implementeren. De andere verstoringen waren echter te complex om op die manier weer te geven. Het was echter wel mogelijk om ze mee te nemen in de GOCE *Simulink* simulatie.
Met het verkregen state space model is daarna een poging gedaan om de bestaande toestand schatter te verbeteren. Hiertoe zijn verschillende modellen gebruikt, maar uiteindelijk bleek toch dat de orginele schatter redelijk optimaal was. Toch is er voor gekozen met de nieuwe schatter verder te gaan, aangezien in het oude model de gravitatie torque niet was meegenomen, waardoor de werkelijkheid iets vertekend werd weergegeven.
Vervolgens is er een Linear Kwadratische Regulator (LQR) ontworpen. Deze kon, met zijn tijdsafhankelijke gedrag een veel betere performance halen dan de orginele regelaars. Verder bleek de LQR ook zeer robuust tegen de meeste model variaties, op een na. Variaties in de trend van het magnetisch veld bleken een zeer nadelige invloed te hebben op de stabiliteit van het systeem.
Daarnaast is er ook nog gekeken naar de mogelijkheden om deze regelaar onboard te implementeren en te gebruiken. Uit die analyse bleek dat het mogelijk was een stabiele LQR te implementeren als een tijds span van ruim 24 uur werd gebruikt voor de LQR berekening, aangezien dat de langste periode is van het pseudo-periodieke magnetisch veld in het Local Orbiting Reference Frame. Hierbij is het

echter wel noodzakelijk dat begin en eindpunt van de sateliet na die ruim 24 uur identiek is, wat in de practijk waarschijnlijk niet zo zal zijn.
Verder is er ook nog gekeken naar andere mogelijkheden om de LQR te implementeren met een kortere tijdspan, waarbij de LQR parameters zowel op een normale manier waren berekend (afhankelijk van de tijd), alswel dat een fit tussen de parameters van het magentisch veld en de LQR werd gebruikt. Beide manieren gaven echter geen stabiel resultaat, aangezien de gebruikte parameters teveel verschilden van de werkelijk optimale.
Als eind conclusie kwam daarom vooral naar voren dat, hoewel de LQR geweldige resultaten geeft, het voor zowel robuustheids redenen als implementatie redenen nodig is een manier te vinden om de optimale regel parameters *nauwkeurig* weer te kunnen geven als functie van de waarden van het magnetisch veld. Met die relatie is het duidelijk dat een LQR op alle fronten beter is dan de orginele regelaars. Echter, tot aan dat moment is een onboard LQR implementatie zeer discutabel.

Summary

GOCE, which stands for Gravity and Ocean Circulation Explorer, is scheduled for launch in 2006 and will measure the Earth's gravity field with high accuracy and resolution.
To obtain this high accuracy, there are stringent performance requirements for the GOCE controller. Requirements which are not fully met by the original controllers.
This report details the design of a new controller. The goal of the design is to enhance the performance, by taking into account the periodic variation of the controllability along the orbit. Furthermore, the design should have a reasonable robustness margin and a simple onboard implementation.
To accomplish this, the study started with deriving the dynamical model of the satellite. Furthermore, the size and shape of the disturbances were analyzed and it was found possible to add the gravitational torque to the state space equations of the satellite model. The other disturbances were to complex however, to add in this way. It was possible though, to implement them in the GOCE *Simulink* simulation.
After this, an effort was made to improve the state estimator. Even though different models were used, it was found that the original estimator was quite optimal. Still, it was chosen to continue with the newly tuned estimator, as the original model did not have the model of the gravitational torque implemented.
Subsequently, a Linear Quadratic Regulator (LQR) was designed. With its time-dependant nature, it is capable of obtaining a far better performance and is also highly robust to changes of the Inertia, disturbances and magnetic field magnitude. However, variations in the trend of the magnetic field proved to have a highly negative influence on the stability of the system.
Besides this, options for onboard implementation and use were also checked. From this analysis it became apparent that it is possible to implement a stable LQR if a time span of (a little over) 24 hours is chosen for the LQR gain predictions, as that is the longest period of the pseudo-periodic magnetic field in the Local Orbiting Reference Frame. To do this however, it is imperative that the starting- and ending point of the satellite after those 24 hours is identical, which will not be the case in practice.

Besides this, several other options of implementing the LQR with a shorter time span were checked. Hereby, the LQR was computed both in the normal way, as by using a fit between the magnetic field parameters and optimal LQR. Both ways proved to be instable however, as the used parameters varied to much from the optimal control strategy.
Hence, the final conclusion was, that however the LQR is capable of obtaining great tracking results, it is imperative for both robustness as implementation reasons, that a way is found to compute the optimal control parameters *accuratly*, as a function of the magnetic field. If such a relation is found, the LQR will give an overall better performance compared to the original controllers. However, till then an onboard LQR implementation is quite disputable.

Contents

Introduction

GOCE: Gravity and Ocean Circulation Explorer, is one of the core projects of the European Space Agency (ESA). It is scheduled for launch in 2006 and will improve the gravity field and geoid model for use in a wide range of applications (geodesy, oceanography, solid-Earth physics, etc.) with its high resolution and high accuracy measurements.

The GOCE program is managed by a consortium of European companies [1]: *Alenia Spazio*, the prime contractor, *Astrium GmbH*, *Alcatel Space Industries* and *Laben*.
The algorithms, needed for the Drag Free and Altitude Control (DFAC) were developed by *Alenia Spazio* in cooperation with the *Politechnico di Torino* and successfully passed the ESA examinations at the PDR in March 2002 [2].

The aim of the Drag Free Control (DFC) consists in making the spacecraft free from the environmental non-gravitational forces by generating control forces identical and opposite in sign to those forces. Acting in that way, the DFC makes the satellite fly along an orbit which characteristics only depend on initial position, velocity and the local gravitational fields.
The satellite will fly at an altitude between 240 and 270 km, where the thermosphere drag effects are still significant (very high temperatures) in a sun-synchronous polar orbit [3].
The accuracy demanded and achieved (simulation) by both *Alenia Spazio* as the *ESA preliminary study* in DFC, is given in table 1.

It can be seen that the original controller from *Alenia Spazio* does not meet al the requirements. For that reason *ESA* started a preliminary study in an effort to enhance the results, but as can be seen, the demands are still not fully met.

The objective of this study is to find a better controller which is sufficiently robust and simple to implement on board the satellite by taking into account the periodic roll/yaw exchange of actuation capability, induced by the magnetic field

Table 1: Tracking specifications and simulated control results

		Specification	*Alenia Spazio*	*ESA p.s.*
Roll	Maximum $[deg]$	10	4.48	4.32
	Maximum $[rad/\sqrt{Hz}]$	1.10^{-5}		
Pitch	Maximum $[deg]$	3		0.0133
	Maximum $[rad/\sqrt{Hz}]$	1.10^{-5}		
Yaw	Maximum $[deg]$	5	4.3	1.35
	Maximum $[rad/\sqrt{Hz}]$	1.10^{-5}		
Roll	Maximum $[rad/s]$	3.10^{-5}	2.10^{-4}	1.10^{-4}
velocity	Maximum $[rad/s/\sqrt{Hz}]$	5.10^{-7}	1.10^{-5}	1.10^{-5}
Pitch	Maximum $[rad/s]$	$1.2.10^{-3}$	1.10^{-3}	3.10^{-7}
velocity	Maximum $[rad/s/\sqrt{Hz}]$	5.10^{-7}	6.10^{-7}	1.10^{-6}
Yaw	Maximum $[rad/s]$	3.10^{-5}	6.10^{-5}	1.10^{-5}
velocity	Maximum $[rad/s/\sqrt{Hz}]$	5.10^{-7}	2.10^{-6}	1.10^{-6}
Roll	Maximum $[rad/s^2]$	1.10^{-6}	2.10^{-6}	3.10^{-7}
acceleration	Maximum $[rad/s^2/\sqrt{Hz}]$	7.10^{-8}	4.10^{-7}	2.10^{-7}
Pitch	Maximum $[rad/s^2]$	1.10^{-6}	8.10^{-8}	8.10^{-8}
acceleration	Maximum $[rad/s^2/\sqrt{Hz}]$	7.10^{-8}	2.10^{-8}	7.10^{-8}
Yaw	Maximum $[rad/s^2]$	1.10^{-6}	4.10^{-7}	1.10^{-7}
acceleration	Maximum $[rad/s^2/\sqrt{Hz}]$	7.10^{-8}	5.10^{-8}	7.10^{-8}

change along the orbit.

This report will start with the dynamical analysis of GOCE, followed by the redesign and validation of the estimator. After this, a new (LQR) controller shall be designed and validated.

Chapter 1

GOCE

To full fill the GOCE control objective, the controller system will not only need to minimize the angles and angular rotations, but also the angular accelerations. Therefore, the satellite will need both position and acceleration information. To obtain this information and give the spacecraft the proper actuation, it is equipped with a variety of sensors and actuators, which will be presented below.

1.1 Onboard Sensors

The GOCE has four sensors, which can be used during space flight.

First of all, there are 3 Magnetometers (MGM), which shall be used to determine the direction and magnitude of the Earth's magnetic field with respect to the satellite. The main problem with this type of sensor is that at further distances from the Earth, the residual magnetic field of the spacecraft will start to dominate the total magnetic field measurement. The GOCE shall be orbiting at a relatively low altitude though, so this will not be a problem.

Then there are the Sun Acquisition Sensor (SAS) for determining the direction of the sun and the Earth Sensor Assembly (ESA), which can give an estimate of the direction of the centre of the Earth. These two will not be used though during drag free mode, as the attitude information will be acquired using a Star Tracker.

For determining the spacecraft's orientation, with respect to known stars, a medium Field of View Star Tracker (FOV STR) will be used, primarily because of its high accuracy (arc-second range). Known problems of STR's are that they cannot

stand direct exposure to the Sun and are sensitive to bright light sources in general. Furthermore, they are also known to suffer from occultation (when a planet blocks the view of the stars).
The STR will give the attitude information in the form of a quaternion, see [4].

So, with the help of these sensors both the direction and strength of the magnetic field and the attitude of the satellite will be determined during Drag Free Mode. Information about the spacecraft's accelerations is also required though. Hereto, the special payload of the GOCE shall be used.

1.2 Payload Sensors

The payload of the GOCE consists of two special sensors, a Satellite to Satellite Instrument (SSTI) and an Electrostatic Gravity Gradiometer (EGG).

The SSTI (also known as GPS receiver) provides position and (longitudinal) velocity of the Center of Mass with respect to the Inertial Reference Frame (IRF), see section 2.1. Due to the fact that this data is not required for the attitude control in Drag Free Mode, no use shall be made of this instrument.

The EGG on the other hand is capable of determining all accelerations on the spacecraft, which shall be used to obtain the necessary attitude information. To obtain these accelerations, the EGG measures the force on a proof mass, maintained at the same position within a cage using a control loop.

1.3 Actuators

To control the GOCE, 2 types of actuators are available in Drag Free Mode.

The original GOCE design obtained a so called Micro Propulsion Assembly (MPA) containing several small thrusters. Due to its complexity and cost, this was replaced by the Magnetic Torquer Assembly (MTA).
The MTA consists of three Magnetic Torquers, all perpendicular to each other. Each of these Magnetic Torquers contains an electrified coil, supplied with a user defined constant current, to generate a magnetic moment. Due to the fact that the magnetic moment is within the Earth's magnetic field, a torque will be generated: $\overrightarrow{T} = \overrightarrow{M} \times \overrightarrow{B}$. The three torques generated this way, will be used for

attitude control.
There is a downside to magnetic control however. Every time the magnetic moment and magnetic field are aligned, no control torque can be generated. So using this type of control means having to deal with a fluctuating controllability.

The second actuator, which shall be used, is the Ion Propulsion Assembly (IPA). It will compensate the air drag during the measurement phase (Drag Free Mode) and increase the orbit altitude during orbit raise manoeuvres. The spacecraft has two thrusters mounted, one main and one redundant, both canted towards the Center of Mass. Both thrusters are actuated continuously. Notice that the IPA only controls the X-motion of the satellite; both Y- and Z-motion are left uncontrolled.

Chapter 2

Dynamical analysis

In this chapter the equations of motion will be derived for the satellite and the amplitude and frequency of the disturbances shall be examined. This will be used later for the simulation of the dynamics as well as the design of the estimator and controller.
Before it is possible to do this though, it is essential to define the reference frames which shall be used. For an explanation of the notation the reader is referred to [5]. Furthermore it should be noted that for clarity, time dependencies are omitted from notation in this chapter (so for instance $\underline{x}$ is used instead of $\underline{x}(t)$).

2.1 Reference frame definitions

For the derivation of the equations of motion, three reference frames will be used:

- Inertial Reference Frame (IRF) (Galilean);
- Local Orbiting Reference Frame (LORF) (not Galilean);
- Spacecraft Centered Reference Frame (SCRF) (not Galilean).

Each of these will now be shortly discussed.

2.1.1 Inertial Reference Frame (IRF)

The IRF is a fixed frame with respect to the Earth, see figure 2.1. The center of the Earth is used as origin. The $\overrightarrow{e}_3^{irf}$-axis is the direction of the Earth's rotation

axis and the $\overrightarrow{e}_1^{irf}$-axis lies in the direction of the vernal equinox (Υ). This is the direction from the Earth's center to the sun on the first day of spring. Finally, the $\overrightarrow{e}_2^{irf}$-axis is perpendicular to the other two giving a right-handed frame.
This reference frame is not truly inertial though, as the Earth's rotation axis is known to move slightly under the influence of the gravitational pull of both the Sun and the Moon. Therefore the frame is generally given a time-stamp, to really make it inertial.

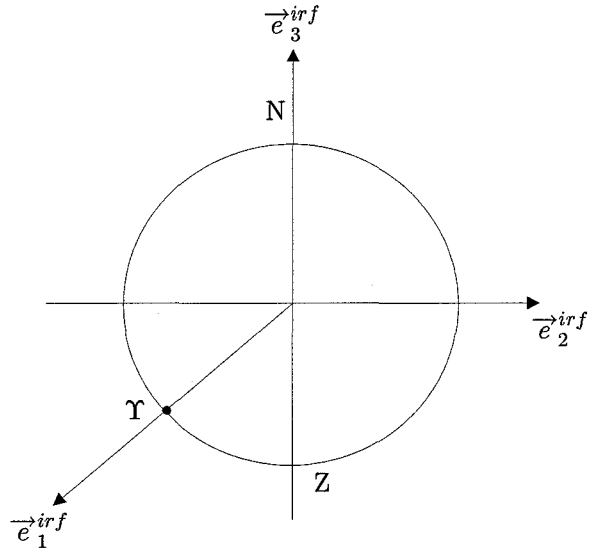


Figure 2.1: The Inertial Reference Frame with respect to the Earth.

2.1.2 Local Orbiting Reference Frame (LORF)

Another frame which is widely used is the LORF. This frame places the Center of Mass (CM) of the spacecraft (origin) with respect to the reference orbit, see figure 2.2.
The $\overrightarrow{e}_3^{lorf}$-axis is in the direction of the Earth's center and the $\overrightarrow{e}_2^{lorf}$-axis lies normal to the inertial orbiting plane, pointing opposite to the orbital momentum vector. Finally, the $\overrightarrow{e}_1^{lorf}$-axis is chosen perpendicular to the other two, giving a right-handed coördinate system.
Note that the the $\overrightarrow{e}_1^{lorf}$-axis lies tangent to the orbit incase of a circular orbit and that the orbital direction is from ascending to descending node, see [6].

For the LORF, special definitions have been made concerning the rotation angles.

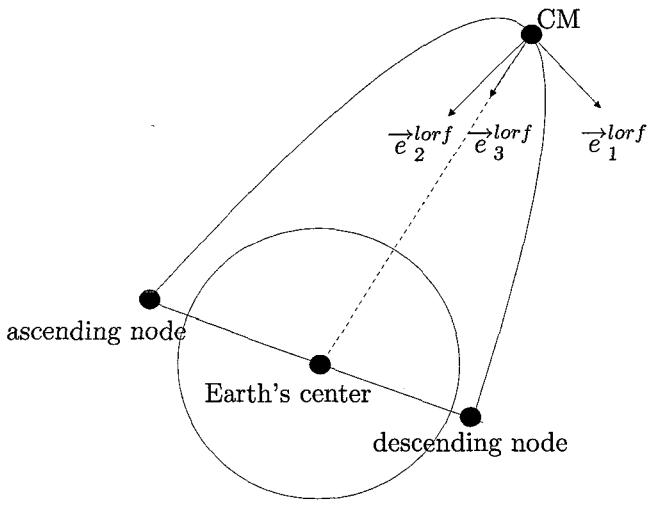


Figure 2.2: The Local Orbiting Reference Frame with respect to the Earth.

Rotation around the $\vec{e}_1^{lorf}$-axis is called roll (ϕ), around the $\vec{e}_2^{lorf}$-axis pitch (θ) and around the $\vec{e}_3^{lorf}$-axis yaw (ψ).

2.1.3 Spacecraft Centered Reference Frame (SCRF)

The SCRF is a body-fixed frame that, when the controller works perfectly, is aligned with the LORF. A graphical representation is given in figure 2.3.
For control purposes this will be considered the so-called control-frame. The control goal will be as said, to keep the control-frame as close as possible to the LORF and at the same time staying within the specifications for velocities and accelerations.

2.2 Derivation of the equations of motion

To derive the equations of motion, the Newton-Euler theorem, see [5], shall be applied with respect to the IRF (as it is Galilean). This consists of two parts: Balance of forces and balance of moments.
From the balance of forces it can be derived that:

$$m\frac{d^2}{dt^2_{IRF}}\vec{p} = \vec{F}_{drag} + \vec{F}_{ITA} + m\vec{g}, \tag{2.1}$$

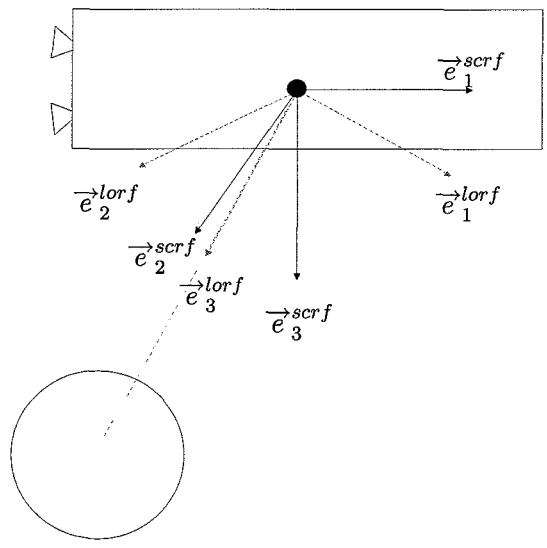


Figure 2.3: The Spacecraft Centered Reference Frame with respect to the Local Orbiting Reference Frame and the Earth.

with $\vec{p}$ the position vector. The only possible control action here, is executed by the ITA, which counters the air drag in $\vec{e}_1^{scrf}$-direction. Forces in $\vec{e}_2^{scrf}$-direction are considered small for small roll, pitch and yaw and forces in $\vec{e}_3^{scrf}$-direction are negligible in comparison to gravity.

Deriving the balance of moments is a bit more complex. The first step is deriving the angular momentum vector ($\vec{H}_{CM}$). The definition of this vector is given in equation 2.2.

$$\vec{H}_{CM} = {}_{CM}\mathbf{J} \cdot \vec{\omega}, \tag{2.2}$$

with $\vec{\omega}$ the angular velocity vector (of the SCRF with respect to the IRF) and ${}_{CM}\mathbf{J}$ the inertia tensor of the center of mass of the spacecraft.
After that, the equations of motion are given by:

$$\sum \vec{M}_{CM} = \dot{\vec{H}}_{CM}. \tag{2.3}$$

Using the additivity property of relative angular velocities [5], it can be found that the angular velocity vector of the SCRF with respect to the IRF can be written as given in equation 2.4

$${}^{scrf/irf}\vec{\omega} = {}^{lorf/irf}\vec{\omega} + {}^{scrf/lorf}\vec{\omega}. \tag{2.4}$$

When using a 3-2-1 sequence of three (single axis) elementary rotations, ${}^{scrf/lorf}\vec{\omega}$ can be written as:

$$\begin{aligned} {}^{scrf/lorf}\vec{\omega} = {}^{41}\vec{\omega} &= {}^{21}\vec{\omega} + {}^{32}\vec{\omega} + {}^{43}\vec{\omega} \\ &= \vec{\underline{e}}^{1^T} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \vec{\underline{e}}^{2^T} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \vec{\underline{e}}^{3^T} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \\ &= \vec{\underline{e}}^{4^T} \underline{A}^{41} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \vec{\underline{e}}^{4^T} \underline{A}^{42} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \vec{\underline{e}}^{4^T} \underline{A}^{43} \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}, \end{aligned} \tag{2.5}$$

where ϕ is the roll, θ the pitch and ψ the yaw.
Furthermore, the transformation matrices are given by:

$$\underline{A}^{43} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix},$$

$$\underline{A}^{42} = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ sin(\phi)sin(\theta) & cos(\phi) & sin(\phi)cos(\theta) \\ cos(\phi)sin(\theta) & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix},$$

$$\underline{A}^{41} = \begin{bmatrix} cos(\theta)cos(\psi) & cos(\theta)sin(\psi) & -sin(\theta) \\ sin(\phi)sin(\theta)cos(\psi) - cos(\phi)sin(\psi) & sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi) & sin(\phi)cos(\theta) \\ cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi) & cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi) & cos(\phi)cos(\theta) \end{bmatrix}. \tag{2.6}$$

Substitution of 2.6 in 2.5 gives:

$${}^{scrf/lorf}\vec{\omega} = \vec{\underline{e}}^{scrf^T} \begin{bmatrix} \dot{\phi} - \dot{\psi}sin(\theta) \\ \dot{\theta}cos(\phi) + \dot{\psi}sin(\phi)cos(\theta) \\ -\dot{\theta}sin(\phi) + cos(\phi)cos(\theta)\dot{\psi} \end{bmatrix}. \tag{2.7}$$

For the derivation of ${}^{lorf/irf}\vec{\omega}$ it is essential to note that the transformation from $\vec{\underline{e}}^{irf}$ to $\vec{\underline{e}}^{lorf}$ consists of a single rotation around the axis pointing towards Earth's center, with angular velocity $-\omega_0$. Therefore, ${}^{lorf/irf}\vec{\omega}$ may be written as:

$$\begin{aligned} {}^{lorf/irf}\vec{\omega} &= -\omega_0 \vec{e}_2^{lorf} \\ &= \vec{e}^{scrf^T} \underline{A}^{41} \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix} \\ &= \vec{e}^{scrf^T} \begin{bmatrix} -\omega_0 cos(\theta)sin(\psi) \\ -\omega_0(sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi)) \\ -\omega_0(cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi)) \end{bmatrix}. \end{aligned} \tag{2.8}$$

Using 2.4, 2.7 and 2.8, it can be found that:

$$ {}^{scrf/irf}\overrightarrow{\omega} = \begin{bmatrix} \dot{\phi} - \dot{\psi}sin(\theta) - \omega_0 cos(\theta)sin(\psi) \\ \dot{\theta}cos(\phi) + \dot{\psi}sin(\phi)cos(\theta) - \omega_0(sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi)) \\ -\dot{\theta}sin(\phi) + cos(\phi)cos(\theta)\dot{\psi} - \omega_0(cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi)) \end{bmatrix}. \tag{2.9} $$

For small roll, pitch and yaw (which may be assumed) this resolves to:

$$ {}^{scrf/irf}\overrightarrow{\omega} = \begin{bmatrix} \dot{\phi} - \omega_0\psi \\ \dot{\theta} - \omega_0 \\ \dot{\psi} + \omega_0\phi \end{bmatrix}. \tag{2.10} $$

Under the assumption that the inertia is given as:

$$ {}_{CM}\mathbf{J} = J_{xx}\overrightarrow{e}_1^{scrf}\overrightarrow{e}_1^{scrf} + J_{yy}\overrightarrow{e}_2^{scrf}\overrightarrow{e}_2^{scrf} + J_{zz}\overrightarrow{e}_3^{scrf}\overrightarrow{e}_3^{scrf}, \tag{2.11} $$

it is possible to substitute equations 2.11 and 2.10 in equation 2.2, to obtain the angular momentum vector. Using equation 2.3, with the knowledge that:

$$ \dot{\overrightarrow{H}}_{CM} = {}_{CM}\mathbf{J} \cdot {}^{scrf/irf}\dot{\overrightarrow{\omega}} + {}^{scrf/irf}\overrightarrow{\omega} \times ({}_{CM}\mathbf{J} \cdot {}^{scrf/irf}\overrightarrow{\omega}), \tag{2.12} $$

gives the equations of motion (for small angles) as given in equation 2.13.

$$ \begin{aligned} \ddot{\phi} &= -\frac{J_{yy} - J_{zz}}{J_{xx}}\omega_0^2\phi + (1 - \frac{J_{yy} - J_{xx}}{J_{xx}})\omega_0\dot{\psi} + \frac{M_x}{J_{xx}}, \\ \ddot{\theta} &= \frac{M_y}{J_{yy}}, \\ \ddot{\psi} &= \frac{J_{xx} - J_{yy}}{J_{zz}}\omega_0^2\psi - (1 + \frac{J_{xx} - J_{yy}}{J_{zz}})\omega_0\dot{\phi} + \frac{M_z}{J_z}. \end{aligned} \tag{2.13} $$

In equation 2.13, M_i represents torques applied on the center of mass. These consist of both actuation and disturbance torques. These will be discussed next.

2.3 Actuation and disturbance torques

In this section the characteristics of the forces working on the satellite will be discussed. With this information the dynamic model may be completed and compared to the one implemented in the Simulink simulation file.

2.3.1 Actuation

As said before, the main control loop is that of the attitude. Actuation of this loop is accomplished by three magnetic torquers, all orthogonal to each other, giving magnetic moments in the directions of the SCRF-axis. The resulting torque on the spacecraft depends on the size and direction of the Earth magnetic field. The resulting control torque on the satellite is given by:

$$\vec{T}_{act} = \vec{M}_{act} \times \vec{B}_{mag}. \tag{2.14}$$

To generate a magnetic moment, a control current has to be applied continuously to a magnetic torquer, see equation 2.15:

$$\overrightarrow{M_{act}} = NIA\vec{n}, \tag{2.15}$$

with N (-) the number of winds of the coil, I (A) the current, A (m^2) the enclosed area and $\vec{n}$ the unit vector normal to the plane of the loop.
Due to fluctuating voltages and resistance of the coil (temperature dependant), obtaining the intended magnetic moment might prove slightly difficult. This can be overcome by setting up a control loop for the coil current. This part shall not be modelled though. It shall be assumed that it is possible to obtain the intended magnetic moment, within the saturation bounds ± 400 (A/m^2).

The other control loop, the longitudinal drag compensation with the ITA shall not be taken into account due to its simplicity. It will be assumed that the ITA is capable of maintaining a constant commanded velocity.
There is a problem caused by the ITA though. Due to its misalignment and movement of the center of mass, errors will occur. These errors shall be considered to be disturbances which need to be countered just like the other disturbances.

2.3.2 Disturbances

Beside the misalignment induced disturbance of the actuators, there are three more disturbance torques which need to be considered: a magnetic disturbance torque; an aerodynamic torque and a gravitational torque. For each of these the order of magnitude needs to be determined and when possible, a dynamical model needs to be derived.

ITA misalignment

For the ITA, assume that the thrust application point is given by:

$$\underline{a}_{ita} = [-2.23 \quad 0 \quad \pm 0.1] \quad [m], \tag{2.16}$$

with respect to the SCRF and the thrust unit vector can be approximated by:

$$\underline{v}_{ita}\underline{\vec{e}}^{scrf} = [0.999 \quad 0.005 \quad \pm 0.044]\underline{\vec{e}}^{scrf}. \tag{2.17}$$

Furthermore, it is known from [1] that the maximum force of the ITA is 20.10^{-3} N.
Using all this, it can be found that the maximum disturbance torque caused by the ITA misalignment can be estimated to be:

$$\begin{aligned} \underline{T}_{ITA}^{scrf} &= \underline{a}_{ita} \times F_{ITA}^{max} v_{ita} \\ &= [1.10^{-5} \quad 4.10^{-5} \quad 2.10^{-4}] \ [Nm]. \end{aligned} \tag{2.18}$$

Note that this is a very rough estimation, to be used only to get an indication of the order of magnitude of the ITA induced disturbance torque.
Furthermore, the frequency content of the ITA misalignment torque will resemble that of the aerodynamic torque as it is the goal of the ITA to cancel it.

Residual magnetic field

Then there is a magnetic disturbance torque resulting from the interaction of the spacecraft's residual magnetic field and the Earth's magnetic field. This torque can be described by:

$$\vec{T}_m = \vec{M}_0 \times \vec{B}_{mag}, \tag{2.19}$$

where $\vec{M}_0$ is the residual magnetic dipole, which is considered to be constant within the SCRF.
Other sources that can produce a magnetic disturbance torque are Eddy currents and hysteresis [6]. These are considered to be small though and shall not be taken into account.
When assuming a residual dipole of: $\vec{M}_0 = [0 \quad 6 \quad 0]$ and the magnetic field values as shown in figure 2.7, it is possible to estimate the maximum disturbance torque induced by the residual magnetic dipole to be of the order:

$$\underline{T}_{mag}^{scrf} = [\pm 3.10^{-4} \quad 0 \quad \pm 1.8.10^{-4}] \ [Nm]. \tag{2.20}$$

The frequency content of the magnetic disturbance torque will resemble that of the magnetic field, consisting mainly of the frequencies ω_0 and $2\omega_0$.

Aerodynamic drag

Another disturbance source is the aerodynamic torque around the center of mass, induced by the air-drag of the upper atmosphere. This torque is very complex to model though, as it depends on the geometrical shape, the air flow and the air density.
From these, the density variations might be the most complex, as the subject of structural variations in the atmosphere is still not fully understood. Density variations may be divided in to six types [6]:

1. Diurnal, related to the local time;
2. 27-day, caused by the rotation around the sun;
3. seasonal-latitudinal, partially solar related and not well understood;
4. semi-annual, partially solar related and not well understood;
5. 11-year, 11-year cycle of solar activity;
6. geomagnetic, due to short term changes in solar activity.

In the thermosphere, the 11-year density variations are the largest, amounting to order of magnitude fluctuations at 350 km. The 27-day and semi-annual variations cause fluctuations of the density by a factor 2 or 3 and the others are smaller.

To get an estimation of the order of magnitude, lets assume that the drag force in a certain direction can be estimated by:

$$F_{drag} = \frac{1}{2} Cd \rho_{air} A V^2, \tag{2.21}$$

with A the satellite surface facing the wind in $[m^2]$, Cd the drag coefficient, and V the wind velocity in that direction in [m/s].

The surfaces can be assumed to be 0.9 $[m^2]$ frontal ($\overrightarrow{e}_1^{scrf}$), 12 $[m^2]$ sideways ($\overrightarrow{e}_2^{scrf}$) and 5 $[m^2]$ in $\overrightarrow{e}_3^{scrf}$-direction. Furthermore, the vector from Center of Mass to Center of pressure, is known to be of the order:
$\overrightarrow{v}_{CMCP} = \pm[0.1 \quad 0.01 \quad 0.01]\overrightarrow{\underline{e}}^{scrf}$ [m].
Using this and assuming that the drag coefficient is one, it is possible to estimate the order of magnitude of the aerodynamic torque to be:

$$\begin{aligned} T_{aero}^{scrf} &= \underline{v}_{CMCP}^{scrf} \times \underline{F}_{drag}^{scrf} \\ &= [\pm 3.10^{-7} \quad \pm 1.10^{-5} \quad \pm 1.10^{-5}]\ [Nm]. \end{aligned} \tag{2.22}$$

Furthermore, the frequency content of the air drag will resemble that of the air density ($2\omega_0$).

Similar to the magnetic disturbance torque, the aerodynamic disturbance torque is to complex to implement in the dynamical model of the satellite. The *Simulink* simulation which has been made available has it (with the magnetic field and ITA disturbances) implemented in a disturbances block as will be shown in section 2.4.

Gravitational torque

Finally, any non-symmetrical object of finite dimensions in orbit is subject to a gravitational torque because of the variations in the Earth's gravitational force over the object. From [6] it can be found that this torque may be written as:

$$\overrightarrow{T}_{gg} = 3\omega_0^2(\overrightarrow{Rs} \times ({}_{CM}\mathbf{J} \cdot \overrightarrow{Rs})), \tag{2.23}$$

with $\overrightarrow{Rs}$ the unit vector from the Earth's center to the geometric center of the satellite. For reasons of convenience it shall be assumed that this coincides with the center of mass. Even though this is not the case in reality, the error which is made is negligible.
Again using the assumption of small angles it can be found that $\overrightarrow{Rs}$ can be written as:

$$\overrightarrow{Rs} = \begin{bmatrix} \theta \\ -\phi \\ -1 \end{bmatrix} \overrightarrow{\underline{e}}^{scrf}. \tag{2.24}$$

Substitution of equation 2.24 and 2.11 in equation 2.23 gives:

$$\overrightarrow{T}_{gg} = 3\omega_0^2 \begin{bmatrix} (J_{zz} - J_{yy})\phi \\ (J_{zz} - J_{xx})\theta \\ 0 \end{bmatrix} \overrightarrow{\underline{e}}^{scrf}, \tag{2.25}$$

which gives an analytic model of the gravitational torque. This may be added to the equations of motion as described in 2.13, resulting in the final set of equations of motion which shall be used as dynamical model:

$$\begin{aligned}
\ddot{\phi} &= -\omega_0^2\phi\frac{J_{yy}-J_{zz}}{J_{xx}} + (1-\omega_0\dot{\psi}\frac{J_{yy}-J_{xx}}{J_{xx}}) + 3\phi\omega_0^2\frac{J_{zz}-J_{yy}}{J_{xx}} + \frac{M_x}{J_{xx}},\\
\ddot{\theta} &= 3\theta\omega_0^2\frac{J_{zz}-J_{xx}}{J_{xx}} + \frac{M_y}{J_{yy}},\\
\ddot{\psi} &= \omega_0^2\psi\frac{J_{xx}-J_{yy}}{J_{zz}} - \omega_0\dot{\phi}(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) + \frac{M_z}{J_z}.
\end{aligned} \tag{2.26}$$

When assuming that the pitch stays beneath ± 1 degree and the roll beneath ± 5 degrees, the order of magnitude of the gravitational torque can be estimated to be:

$$T_{gg}^{scrf} = [\pm 6.10^{-4} \qquad \pm 8.10^{-3} \qquad 0]\ [Nm]. \tag{2.27}$$

The main frequency of the gravity gradient torque will depend on the main frequency of the roll and pitch angles.

So, when summarizing the magnitude and frequency information of the disturbances it can be found that the gravitational torque will have a strong influence and that frequencies of ω_0 and $2\omega_0$ are expected to dominate the frequency content of the disturbance torques. The results have been sorted in table 2.1.

Table 2.1: Disturbance magnitude estimations

Disturbance: [Nm]	$\hat{T}_{dist}^x$	$\hat{T}_{dist}^y$	$\hat{T}_{dist}^z$	Frequencies [Hz]
Gravitational Torque	$\pm 6.10^{-4}$	$\pm 8.10^{-3}$	0	$f(\phi,\ \theta)$
Residual Magnetic Dipole Torque	$\pm 3.10^{-4}$	0	$\pm 8.10^{-4}$	$\omega_0,\ 2\omega_0$
ITA misalignment Torque	1.10^{-5}	4.10^{-5}	2.10^{-4}	$2\omega_0$
Aerodynamic Torque	$\pm 3.10^{-7}$	$\pm 1.10^{-5}$	$\pm 1.10^{-5}$	$2\omega_0$

2.4 Simulink implementation

Now that the equations of motion and the primary disturbances are known it is possible to check the results with the given simulation file *dfacs_mag.mdl*, which originated from the *ESA* preliminary study. A schematic representation of this *Simulink*-file is given in figure 2.4.
Note that orbit changes, caused by the firing of the ITA, are not taken along in

the simulation and that the linear approximations are used, which were derived in the previous sections.

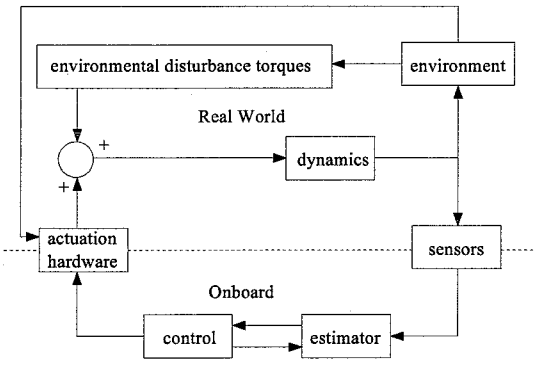


Figure 2.4: *Simulink* simulation representation.

The equations of motion (equation 2.26) are implemented in the *dynamics* block (to mimic the real world satellite) as a state-space model:

$$\begin{aligned} \dot{\underline{x}} &= A\underline{x} + B\underline{u} \\ \underline{y} &= [\phi \quad \theta \quad \psi \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad \ddot{\phi} \quad \ddot{\theta} \quad \ddot{\psi}]^T \\ &= C\underline{x} + D\underline{u}, \end{aligned} \tag{2.28}$$

with states:

$$\underline{x} = \left[\begin{array}{cccccc} \phi & \theta & \psi & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{array} \right]^T, \tag{2.29}$$

and dynamics described by:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4\omega_0^2(\frac{J_{yy}-J_{zz}}{J_{xx}}) & 0 & 0 & 0 & 0 & \omega_0(1-\frac{J_{yy}-J_{zz}}{J_{xx}}) \\ 0 & 3\omega_0^2(\frac{J_{zz}-J_{xx}}{J_{yy}}) & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_0^2(\frac{J_{xx}-J_{yy}}{J_{zz}}) & -\omega_0(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \tag{2.30}$$

and output matrices:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -4\omega_0^2(\frac{J_{yy}-J_{zz}}{J_{xx}}) & 0 & 0 & 0 & 0 & \omega_0(1-\frac{J_{yy}-J_{zz}}{J_{xx}}) \\ 0 & 3\omega_0^2(\frac{J_{zz}-J_{xx}}{J_{yy}}) & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_0^2(\frac{J_{xx}-J_{yy}}{J_{zz}}) & -\omega_0(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) & 0 & 0 \end{bmatrix}$$

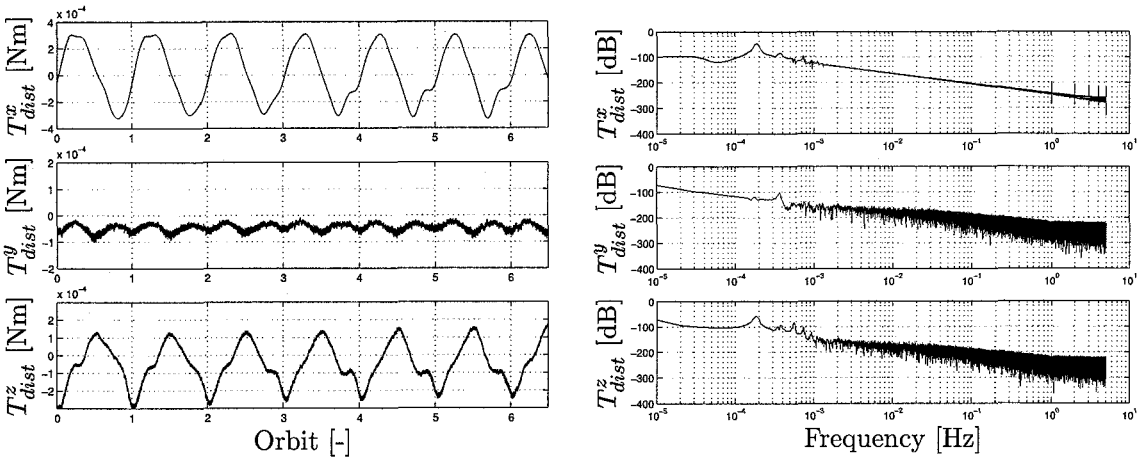
$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix}. \tag{2.31}$$

It must be noted that only the angles and the angular accelerations are measured by the sensors. Still, the velocities are needed as output for monitoring purposes.

The inputs $\underline{u}$ consist of $\underline{T}_{act} + \underline{T}_{dist}$, the actuated torques and the disturbance torques with respect to the SCRF. Herein, the disturbance torques include the magnetic- and aerodynamic-torque as well as the torque induced by the misalignment of the ITA.

Note that the definition of the C matrix differs from the originally implemented C-matrix, due to the fact that the original implementation did not include the gravity gradient torque.
Still, the original results offer good tuning references, because the flaw would only appear in the real world (its both in the dynamic model as the estimator model later on).

To get an indication of the size and shape of the disturbances as implemented, a simulation over 35.000 seconds has been done. The results are given in figure 2.5.



(a) (b)

Figure 2.5: Simulated disturbances (without T^{scrf}_{gg}): (a) time response; (b) power spectrum density.

It can be seen that the disturbances have a strong periodic nature (mainly due to the changes in the Earth's magnetic field over one orbit), with main periods ω_0 and $2\omega_0$, but also consist of a constant part.
Note that the gravity gradient torque is implemented in the dynamics block, so does not affect the output of the disturbances block as discussed here.

Information concerning the environment, needed to compute the environmental torques is loaded in the environment block. It outputs magnetic field data and wind- and air density information which were computed before the simulation to save time. The values of the air density and the wind velocity are given in figure 2.6.

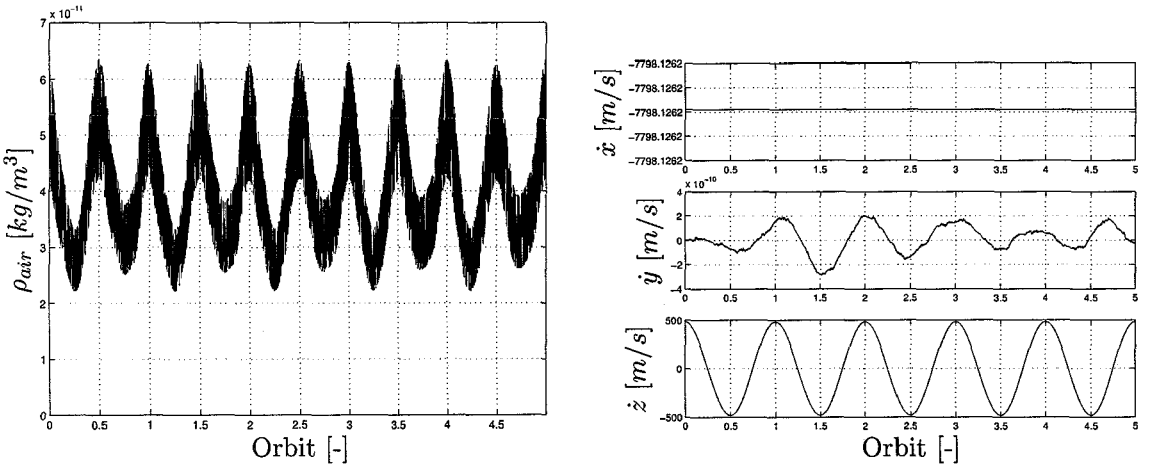


Figure 2.6: Environment: (a) Air density; (b) Wind velocity.

It can be seen that the air density indeed fluctuates heavily each orbit (section 2.3.2), caused by the periodical heating of the air by the sun.
Furthermore, assuming a perfect drag compensation in $\overrightarrow{e}_1^{lorf}$-direction by the ITA, it is possible to keep the velocity in that direction ($\dot{x}$) constant. For now, a forward velocity of 7.8 km/s will be used. The (side way) velocity, in $\overrightarrow{e}_2^{lorf}$-direction ($\dot{y}$), is almost zero, as it is assumed to vary only due to the lateral component of the air drag. The velocity in the planetary direction, $\overrightarrow{e}_3^{lorf}$ ($\dot{z}$), will have a periodic nature, as the gravity is modelled to be keplerian [6].

For the computation of the Earth's magnetic field, a model with variable complexity may be used. Initially, the original IGRF 2000 model of order 8 shall be used, for which the magnitudes in the LORF-directions are given in figure 2.7.

Like the wind velocity, the magnetic field is also specified with respect to the Local Orbiting Reference Frame. However, the Earth's magnetic field does not only vary due to the orbital motion, but also because of the daily rotation of the

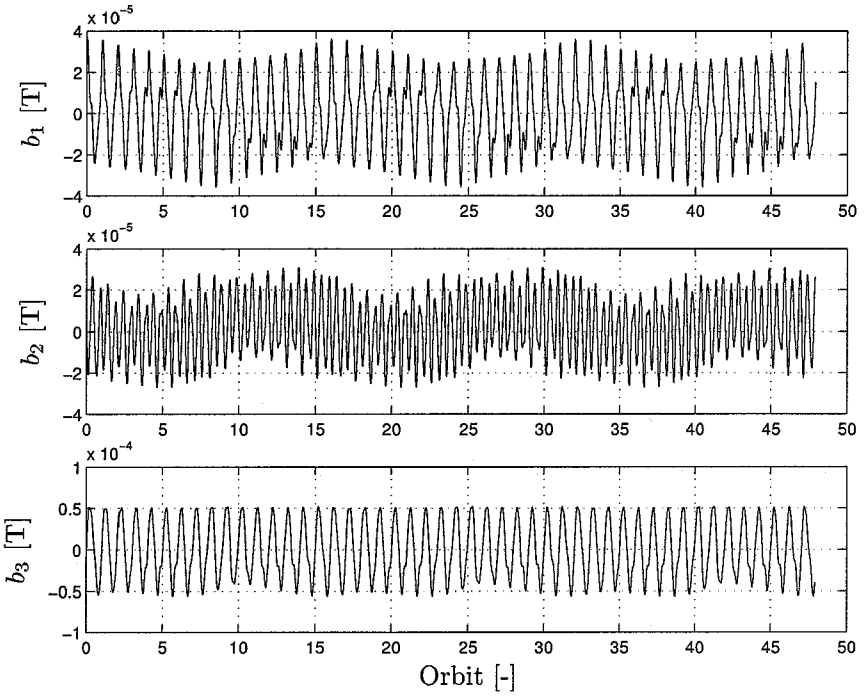


Figure 2.7: Environment: The Earth's magnetic field.

Earth around its own axis. So there are two important harmonics for the Earth magnetic field.

Finally, the control block consists of the *ESA* preliminary PD-controller (with disturbance feed forward), tuned with pole placement and the estimation block consists of a Kalman-filter with a sine-shaped disturbance model. Both controller and estimator are not tuned optimally though.
The steady state response of the satellites last three orbits from a 100.000 seconds simulation (aprox. 20 orbits), controlled with the original controller and the states estimated with the original 'sinusoidal-linear perturbation model'-estimator, is given in figure 2.8.

It can be seen that the controller seems to be stable, but is unable to suppress some remaining periodic errors.
The problem with this controller is that it computes the commanded torques ($\underline{T}_{cmd}^{scrf}$) to counter certain tracking errors instead of computing the commanded MTA moments ($\underline{M}_{cmd}^{scrf}$) directly.
The reason this is done is to prevent having to deal with a time dependant B-

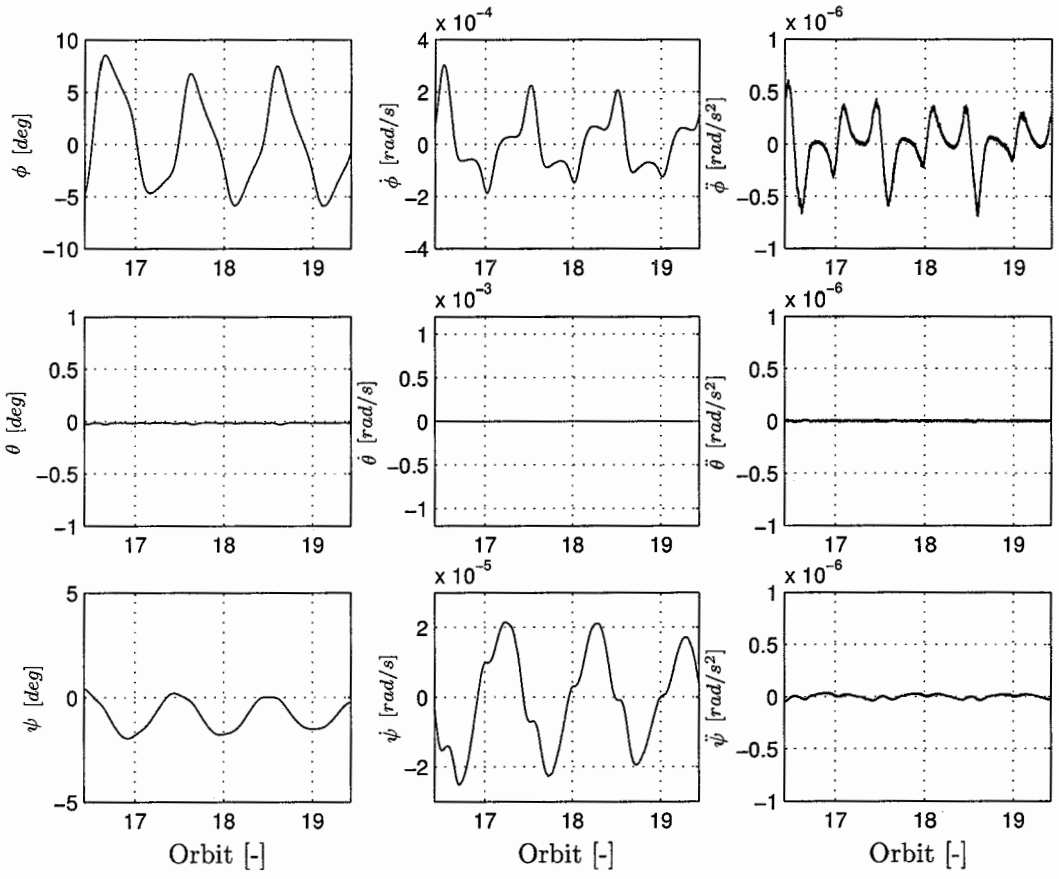


Figure 2.8: Tracking error obtained by original PD-controller, last 3 of 20 orbits.

matrix. Downside of this method is however, that in the end, the commanded torque and the actuated torque ($\underline{T}_{act}^{scrf}$), may differ a great deal. This shall be illustrated by simply writing down the equations.
Computation of the commanded magnetic moments is done using:

$$\vec{M}_{cmd}^{scrf} = \frac{\vec{B}_{mag}^{scrf} \times \vec{T}_{cmd}^{scrf}}{\| \vec{B}_{mag}^{scrf} \|^2}. \tag{2.32}$$

So, using equation 2.14, $\underline{T}_{cmd}^{scrf} = \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix}$ and $\underline{B}_{mag}^{scrf} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$, it can be found that:

$$\underline{T}_{act}^{scrf} = \frac{1}{(b_1^2+b_2^2+b_3^2)} \begin{bmatrix} T_1 b_3^2 + T_1 b_2^2 - b_1 b_3 T_3 - b_1 b_2 T_2 \\ T_2 b_1^2 + T_2 b_3^2 - b_1 b_2 T_1 - b_2 b_3 T_3 \\ T_3 b_1^2 + T_3 b_2^2 - b_1 b_3 T_1 - b_2 b_3 T_2 \end{bmatrix} \tag{2.33}$$
$$\underline{T}_{act}^{scrf} = B_m^{\star} \underline{T}_{cmd}^{scrf}.$$

The graphical representation is given in figure 2.9.

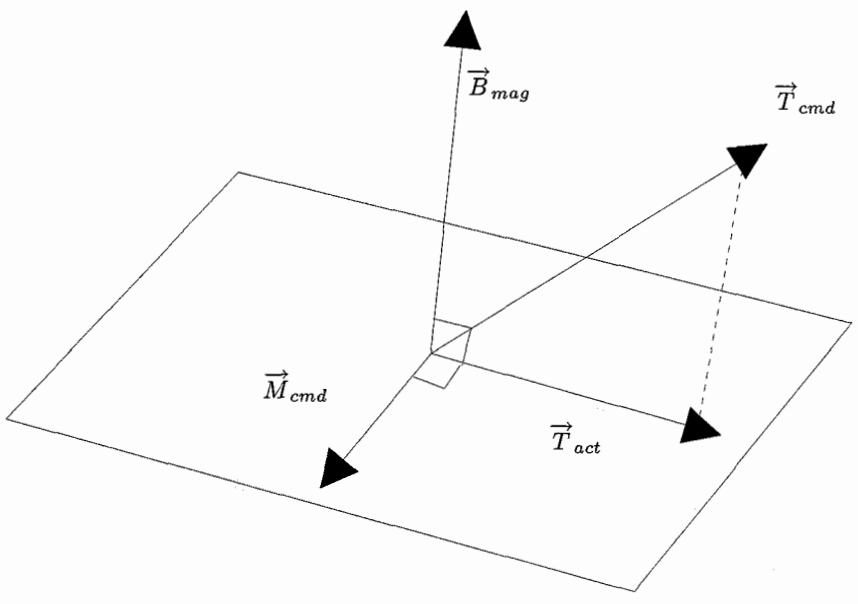


Figure 2.9: Difference commanded- and actual control torque.

Clearly, the commanded and actual torque will generally differ. The only reason this system is stable is due to the fact that the chosen bandwidth for roll and yaw is far beneath the orbital frequency, but still able to suppress influences of the pitch control action.

Another point of interest is the estimation. The maximum estimation errors with respect to the maximum tracking errors, are given in table 2.2. It can be seen that the maximum estimation errors are about ten percent (at its worst) of the maximum tracking errors for the original estimator and controller.

The state estimation shall be discussed in more detail in the next chapter.

Table 2.2: Original estimation errors.

Errors:	Estimation	Control
e_ϕ [deg]	3.10^{-3}	8
e_θ [deg]	9.10^{-3}	3.10^{-2}
e_ψ [deg]	3.10^{-3}	2
$e_{\dot{\phi}}$ [rad/s]	1.10^{-7}	2.10^{-4}
$e_{\dot{\theta}}$ [rad/s]	2.10^{-7}	1.10^{-6}
$e_{\dot{\psi}}$ [rad/s]	1.10^{-7}	2.10^{-5}
e^x_{Tdist} [Nm]	1.10^{-5}	3.10^{-4}
e^y_{Tdist} [Nm]	2.10^{-5}	1.10^{-4}
e^z_{Tdist} [Nm]	2.10^{-5}	3.10^{-4}

Chapter 3

State estimation

For the feedback controller it is essential to have good estimates of the angles and angular velocities. Without those, it will be impossible to get a stable high bandwidth closed-loop system.
Furthermore, for the disturbance feed forward it is also essential to have good estimates on the disturbance torques.

Still, obtaining these good estimates should be doable, considering that the sensors are really accurate and the disturbances have a reasonably predictable behavior. The only question that remains is what kind of models should be used for the attitude dynamics and disturbances in order to get an optimal performance of the estimator.

As can be seen from table 2.2, reasonably small estimation errors can already be obtained using a sinusoidal perturbation model. To get an impression however, of the quality of this estimator, it is important to take a step back. Lets start with an estimator without disturbances modelled.

3.1 Estimation using a model without disturbances

For the real world satellite, we know:

$$\dot{x}(t) = \underline{f}(\underline{x}, \underline{p}, t) + B\underline{u}(t), \tag{3.1}$$

with $\underline{f}(\underline{x}, \underline{p}, t)$ a non-linear function depending on $\underline{x}(t)$, the states: angles and angular velocities, $\underline{p}$ the position vector on the orbit, t the time and $\underline{u}(t)$ the sum

of torques.
We also know that for small angles this can be written in the form of equation 2.28. This will be the basis of the estimation model:

$$\begin{aligned} \dot{\underline{x}}(t) &= \hat{A}\underline{\hat{x}}(t) + \hat{B}\underline{\hat{u}}(t) + G\underline{w}(t) \\ \underline{y}(t) &= \hat{C}\underline{x}(t) + \hat{D}\underline{u}(t) + H\underline{w}(t) \\ \underline{z}(t) &= \underline{y}(t) + \underline{v}(t), \end{aligned} \tag{3.2}$$

with $\underline{z}(t)$ the measurements (angles and acceleration), $\underline{w}(t)$ the model noise with mean zero and covariance matrix Q and $\underline{v}(t)$ the measurement noise with covariance matrix R and mean zero.
Herein, the influence of the model noise $G\underline{w}(t)$ and $H\underline{w}(t)$, can be seen as:

$$\begin{aligned} \underline{\dot{x}} &= \dot{\underline{\hat{x}}}(t) + G\underline{w}(t) \\ \underline{y} &= \underline{\hat{y}}(t) + H\underline{w}(t). \end{aligned} \tag{3.3}$$

The model noise on the outputs ($H\underline{w}(t)$) shall be considered to be zero from this point on.

From the STR and EGG hardware description [7], it is known that:

$$R = \begin{bmatrix} 1.10^{-8} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.10^{-8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.10^{-8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5.10^{-16} & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.5.10^{-16} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.5.10^{-16} \end{bmatrix}. \tag{3.4}$$

Using a continuous, time invariant gain estimator, the estimation can be written as (see [8]):

$$\begin{aligned} \dot{\underline{\hat{x}}}(t) &= \hat{A}\underline{\hat{x}}(t) + \hat{B}\underline{\hat{u}}(t) + K[\underline{z}(t) - \hat{C}\underline{\hat{x}}(t) - \hat{D}\underline{\hat{u}}(t)] \\ &= (\hat{A} - K\hat{C})\underline{\hat{x}}(t) + (\hat{B} - K\hat{D})\underline{\hat{u}}(t) + K\underline{z}(t). \end{aligned} \tag{3.5}$$

The relationship given in equation 3.5 can be seen as a dynamical system, with state-space matrices:

$$\begin{aligned} A_{est} &= [\hat{A} - K\hat{C}] \\ B_{est} &= [\hat{B} - K\hat{D}, K] \\ C_{est} &= [I] \\ D_{est} &= [0]. \end{aligned} \tag{3.6}$$

A constant (time invariant) estimator gain (K) was chosen for computational simplicity. This was possible due to the fact that the matrices A,B,R,Q,C are constant. Even though the transitional phase of the estimation will not be optimal, the estimation will be optimal in steady state.
Computation of the optimal constant estimator gain (K) can be done using *Kalman* theory [8].

For the estimator without disturbances modelled the representation of equation 3.2 leads to:

$$\hat{A} = A$$

$$\hat{B} = B$$

$$\hat{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -4\omega_0^2(\frac{J_{yy}-J_{zz}}{J_{xx}}) & 0 & 0 & 0 & 0 & \omega_0(1-\frac{J_{yy}-J_{zz}}{J_{xx}}) \\ 0 & 3\omega_0^2(\frac{J_{zz}-J_{xx}}{J_{yy}}) & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_0^2(\frac{J_{xx}-J_{yy}}{J_{zz}}) & -\omega_0(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) & 0 & 0 \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix}, \tag{3.7}$$

where A and B are as given in equation 2.30 and the states contain the estimated angles and velocities.
Note that although the model of the real world outputs both angles, velocities and accelerations, there are only measurements available of the angles and accelerations, which explains the shape of $\hat{C}$ and $\hat{D}$.
Using this (and a quickly tuned Q-matrix, with model noise only on the accelerations), a simulation has been done for 20.000 seconds. The maximum angle and angular velocity estimation errors are given in table 3.1. It must be noted though that the controller showed huge (order 1.10^2 deg) angle errors due to the larger estimation errors and lack of disturbance feed forward. Clearly better estimates are necessary.

Table 3.1: Estimation errors: Original and model without disturbances.

	Original	Model without dist.
e_ϕ [deg]	3.10^{-3}	5.10^{-2}
e_θ [deg]	9.10^{-3}	2.10^{-2}
e_ψ [deg]	3.10^{-3}	1.10^{-1}
$e_{\dot{\phi}}$ [rad/s]	1.10^{-7}	2.10^{-4}
$e_{\dot{\theta}}$ [rad/s]	2.10^{-7}	2.10^{-5}
$e_{\dot{\psi}}$ [rad/s]	1.10^{-7}	1.10^{-4}
$e_{T^x_{dist}}$ [Nm]	1.10^{-5}	3.10^{-4}
$e_{T^y_{dist}}$ [Nm]	2.10^{-5}	1.10^{-4}
$e_{T^z_{dist}}$ [Nm]	2.10^{-5}	3.10^{-4}

3.2 Estimation using a model with disturbances

In the previous section it was concluded that it is necessary to use a disturbance model to obtain better estimation results. From section 2.3.2 and 2.4 it is known that the main disturbance torque frequency is ω_0 for T^x_{dist} and T^z_{dist} and $2\omega_0$ for T^y_{dist}.
Using sine-functions with those frequencies, it is possible to make a fit upon the measurements. This way the following disturbance model is found:

$$\underline{T}_{dist} = \begin{bmatrix} 3.10^{-4} sin(\omega_0 t) \\ -6.10^{-5} + 4.10^{-5} sin(2\omega_0 t - \frac{1}{2}\pi) \\ -1.10^{-4} + 2.10^{-4} sin(\omega_0 t - \frac{1}{2}\pi) \end{bmatrix}. \tag{3.8}$$

A graphical comparison of model and measurements is given in figure 3.1, with the model represented by the dashed line. As can be seen the model indeed gives a reasonable fit.

Note that this model is not generally applicable. The phase and values may differ in reality, only the period will remain the same. It is therefore necessary to include the disturbances in the state vector of the estimator. That way the disturbances will be estimated with the help of both the model and the measurements. So the estimated states will be:

$$\hat{\underline{x}}(t) = [\phi \quad \theta \quad \psi \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \quad T^x_{dist} \quad T^y_{dist} \quad T^z_{dist} \quad \dot{T}^x_{dist} \quad \dot{T}^y_{dist} \quad \dot{T}^z_{dist} \quad]. \tag{3.9}$$

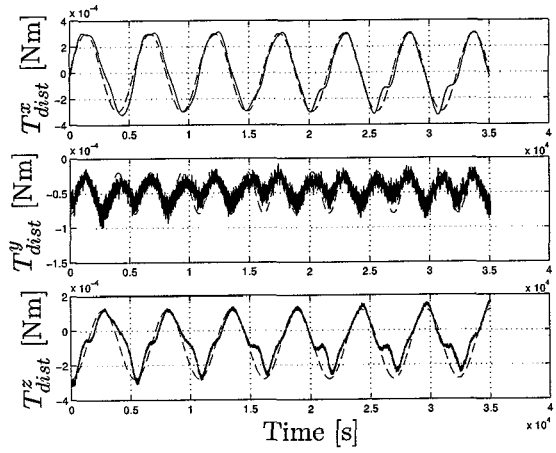


Figure 3.1: Difference measured (solid) and modelled (dashed) disturbance torques.

Using this, the dynamics become:

$$\hat{A} = \begin{bmatrix} 0(3) & I(3) & 0(3) & 0(3) \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} & 0(3) \\ 0(3) & 0(3) & 0(3) & I(3) \\ 0(3) & 0(3) & \hat{A}_{43} & 0(3) \end{bmatrix},$$

$$\hat{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.10}$$

with $I(3)$ a (3×3) identity-matrix, $0(3)$ a (3×3) zero-matrix and:

$$
\begin{aligned}
\hat{A}_{21} &= \begin{bmatrix} -4\omega_0^2(\frac{J_{yy}-J_{zz}}{J_{xx}}) & 0 & 0 \\ 0 & 3\omega_0^2(\frac{J_{zz}-J_{xx}}{J_{yy}}) & 0 \\ 0 & 0 & \omega_0^2(\frac{J_{xx}-J_{yy}}{J_{zz}}) \end{bmatrix} \\
\hat{A}_{22} &= \begin{bmatrix} 0 & 0 & \omega_0(1-\frac{J_{yy}-J_{zz}}{J_{xx}}) \\ 0 & 0 & 0 \\ -\omega_0(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) & 0 & 0 \end{bmatrix} \\
\hat{A}_{23} &= \begin{bmatrix} \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \\
\hat{A}_{43} &= \begin{bmatrix} -\omega_0^2 & 0 & 0 \\ 0 & -4\omega_0^2 & 0 \\ 0 & 0 & -\omega_0^2 \end{bmatrix}.
\end{aligned} \tag{3.11}
$$

Furthermore, for the outputs we find:

$$
\begin{aligned}
\hat{C} &= \begin{bmatrix} I(3) & 0(3) & 0(3) & 0(3) \\ \hat{C}_{21} & \hat{C}_{22} & \hat{C}_{23} & 0(3) \\ 0(3) & 0(3) & 0(3) & I(3) \\ 0(3) & 0(3) & \hat{C}_{43} & 0(3) \end{bmatrix} \\
\hat{D} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix},
\end{aligned} \tag{3.12}
$$

with:

$$
\begin{aligned}
\hat{C}_{21} &= \begin{bmatrix} -4\omega_0^2(\frac{J_{yy}-J_{zz}}{J_{xx}}) & 0 & 0 \\ 0 & 3\omega_0^2(\frac{J_{zz}-J_{xx}}{J_{yy}}) & 0 \\ 0 & 0 & \omega_0^2(\frac{J_{xx}-J_{yy}}{J_{zz}}) \end{bmatrix} \\
\hat{C}_{22} &= \begin{bmatrix} 0 & 0 & \omega_0(1-\frac{J_{yy}-J_{zz}}{J_{xx}}) \\ 0 & 0 & 0 \\ -\omega_0(1+\frac{J_{xx}-J_{yy}}{J_{zz}}) & 0 & 0 \end{bmatrix} \\
\hat{C}_{23} &= \begin{bmatrix} \frac{1}{J_{xx}} & 0 & 0 \\ 0 & \frac{1}{J_{yy}} & 0 \\ 0 & 0 & \frac{1}{J_{zz}} \end{bmatrix} \\
\hat{C}_{43} &= \begin{bmatrix} -\omega_0^2 & 0 & 0 \\ 0 & -4\omega_0^2 & 0 \\ 0 & 0 & -\omega_0^2 \end{bmatrix}.
\end{aligned}
\tag{3.13}
$$

Again it is possible to write the estimation in the dynamical state space representation as is given in equation 3.6, where the constant gain can be computed by using *Kalman* theory [8]. Due to the large matrices, the Q-matrix is not so easy to tune though.

3.2.1 Computation of Q

It would be nice to obtain an indication for the values of the Q-matrix, to have a stable starting point to start the tuning from. That is why it is necessary to analyze the nature of the model noise.
Recall the real world model (equation 3.1) and the estimation model (equation 3.2). Combining both gives:

$$f(\underline{x},\underline{p},t) + B\underline{u}(t) = \hat{A}\underline{\hat{x}}(t) + \hat{B}\underline{\hat{u}}(t) + G\underline{w}(t), \tag{3.14}$$

with $\underline{u}(t) = \underline{T}_{act}(t)$ and $\underline{\hat{u}}(t) = \underline{\hat{T}}_{act}(t)$, but since it is known that $\underline{f}(\underline{x},\underline{p},t)$ can be approximated by:

$$\underline{f}(\underline{x},\underline{p},t) = (\hat{A} + \delta A)\underline{x}(t) + \delta\underline{f}(\underline{x},\underline{p},t), \tag{3.15}$$

it is possible to write this as:

$$(\hat{A} + \delta A)\underline{x}(t) + \delta\underline{f}(\underline{x},\underline{p},t) + B\underline{u}(t) = \hat{A}\underline{\hat{x}}(t) + \hat{B}\underline{\hat{u}}(t) + G\underline{w}(t). \tag{3.16}$$

So, for the model noise this results to:

$$G\underline{w}(t) = \hat{A}(\underline{x}(t) - \hat{\underline{x}}(t)) + \delta A\underline{x}(t) + \delta\underline{f}(\underline{x}, \underline{p}, t) + B\underline{u}(t) - \hat{B}\underline{u}(t), \tag{3.17}$$

with:

- $\hat{A}(\underline{x}(t) - \hat{\underline{x}}(t))$, the influence of the estimation error;
- $\delta A\underline{x}(t)$, the uncertainty of the dynamics linearity;
- $\delta\underline{f}(\underline{x}, \underline{p}, t)$, un-modelled non-linear dynamics;
- $B\underline{u}(t) - \hat{B}\underline{u}(t)$, the uncertainty between actual torque and its estimation as there may be a discrepancy between the actual magnetic field and the measurement.

The covariance matrix Q has to be chosen with respect to all these uncertainties, in such a way that:

$$Q = E[G\underline{w}(t)(G\underline{w}(t))^T], \tag{3.18}$$

with E the mean of the function.

Each of these errors has been estimated with the help of the original simulation file and their contributions to the initial guess of the Q-matrix are given in table 3.2. Note that only Q's diagonal terms are given. The other terms are zero as there are no cross-correlations.

Using this Q-matrix a simulation can be done, to evaluate and tune the estimator.

3.2.2 Tuning the estimator

Despite the previously described analytic approach, the estimation proved instable. The reason is the overestimation of the error caused by the un-modelled non-linear dynamics. These only occur in the disturbance model and were determined by taking the difference between the real disturbances and the disturbance

Table 3.2: Contributions to the model noise.

Part:	Q-contribution:
$\hat{A}(\underline{x}(t) - \underline{\hat{x}}(t))$	$\begin{bmatrix} 1.10^{-13} & 1.10^{-13} & 2.10^{-14} & 9.10^{-15} & 1.10^{-17} & 2.10^{-17} & ... \\ ... & 2.10^{-17} & 3.10^{-16} & 6.10^{-17} & 1.10^{-18} & 2.10^{-19} & 9.10^{-20} \end{bmatrix}$
$\delta A\underline{x}(t)$	[0] (real world is linear in simulation)
$\delta \underline{f}(\underline{x}, \underline{p}, t)$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & ... \\ ... & 2.10^{-9} & 5.10^{-10} & 6.10^{-9} & 7.10^{-13} & 4.10^{-15} & 2.10^{-13} \end{bmatrix}$
$B\underline{u} - \hat{B}\underline{u}$	$\begin{bmatrix} 0 & 0 & 0 & 7.10^{-14} & 1.10^{-14} & 5.10^{-14} & ... \\ ... & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Initial Guess:	$\begin{bmatrix} 1.10^{-13} & 1.10^{-13} & 2.10^{-14} & 8.10^{-14} & 1.10^{-14} & 5.10^{-14} & ... \\ ... & 2.10^{-9} & 5.10^{-10} & 6.10^{-9} & 7.10^{-13} & 4.10^{-15} & 9.10^{-13} \end{bmatrix}$

model as is given in equation 3.8. In reality this difference is far smaller, because the model used by the filter only consists of sine-functions with frequencies ω_0 and $2\omega_0$ and is shaped by the measurements.
It is hard to get a better estimation though for the un-modelled non-linear dynamics, so for the new initial guess they will be discarded, assuming that there are no un-modelled non-linear dynamics.
Using:

$$Q_{intial}^{diag} = \begin{bmatrix} 1.10^{-13} & 1.10^{-13} & 2.10^{-14} & 8.10^{-14} & 1.10^{-14} & 5.10^{-14} & ... \\ ... & 2.10^{-17} & 3.10^{-16} & 6.10^{-17} & 1.10^{-18} & 2.10^{-19} & 9.10^{-20} \end{bmatrix}, \tag{3.19}$$

the estimation was found to converge. After some fine-tuning with respect to the singular values of the dynamical estimation system and the estimation errors, the following model noise covariance matrix was found:

$$Q_{final}^{diag} = \begin{bmatrix} 1.10^{-13} & 1.10^{-13} & 2.10^{-14} & 1.10^{-14} & 5.10^{-17} & 1.10^{-14} & ... \\ ... & 1.10^{-16} & 5.10^{-17} & 6.10^{-17} & 2.10^{-15} & 4.10^{-20} & 2.10^{-18} \end{bmatrix}. \tag{3.20}$$

A simulation using this Q-matrix, the sine-shaped-disturbance- model-estimator (SSDME) and the original PD-controller was done for a time-span of 140.000 seconds. The estimation results with respect to the original *sinlin*-estimator are given in table 3.3.

Considering that there is a small difference between the real-world measurement models, one can only conclude that the original estimator was reasonably well tuned. Overall, about the same estimation errors are found. A graphical repre-

Table 3.3: Maximum estimation errors original (*sinlin*) and SSDME.

Errors:	Original	SSDME
e_ϕ [deg]	3.10^{-3}	3.10^{-3}
e_θ [deg]	9.10^{-3}	1.10^{-2}
e_ψ [deg]	3.10^{-3}	3.10^{-3}
$e_{\dot{\phi}}$ [rad/s]	1.10^{-7}	2.10^{-7}
$e_{\dot{\theta}}$ [rad/s]	2.10^{-7}	2.10^{-7}
$e_{\dot{\psi}}$ [rad/s]	1.10^{-7}	$1.5.10^{-7}$
e^x_{Tdist} [Nm]	1.10^{-5}	7.10^{-6}
e^y_{Tdist} [Nm]	2.10^{-5}	$2.5.10^{-5}$
e^z_{Tdist} [Nm]	2.10^{-5}	2.10^{-5}

sentation of the estimation errors is given in figure 3.2.

It can be seen that the estimation error signal of the angles consists of a bias and noise, both coming from the STR. The noise level of the estimations is far smaller than that of the measurements though. This can be seen well in figure 3.3, which shows the measurement- and estimation error of the roll for the first orbit. Furthermore, it can also be seen that it takes about 2000 seconds for the estimator to converge.

The disturbance estimations consist of a harmonic and noise. The noise in this case is coming from the acceleration measurement and the harmonics are of the main frequencies not taken along by the model, so ω_0 for the pitch and $2\omega_0$ for the roll and yaw. Adding additional sines with those frequencies to the disturbance model did not improve the performance though.

With the tuned SSDME, it is possible to take a closer look at enhancing the control results.

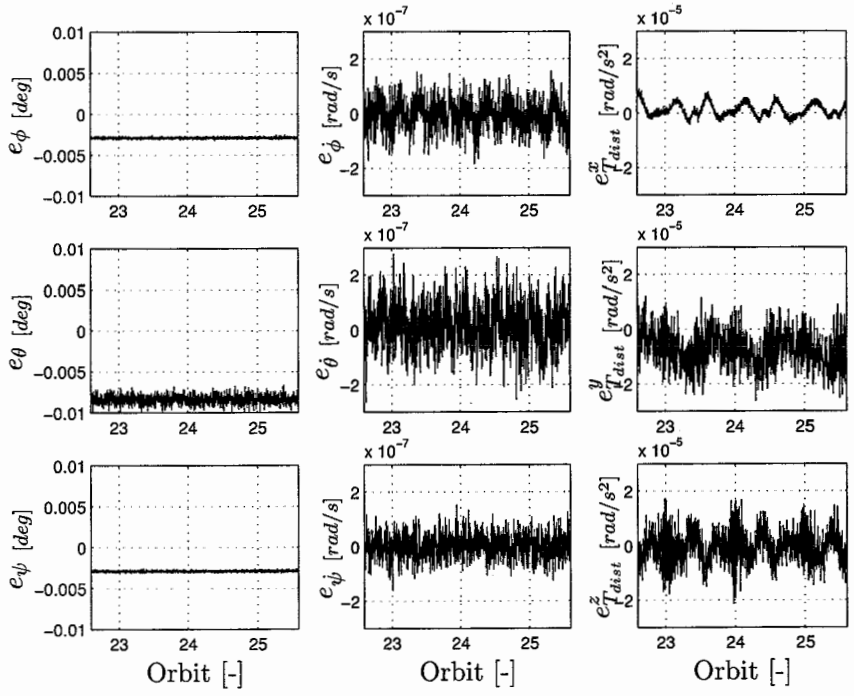


Figure 3.2: Estimation error obtained by SSDME, last 3 of 26 orbits.

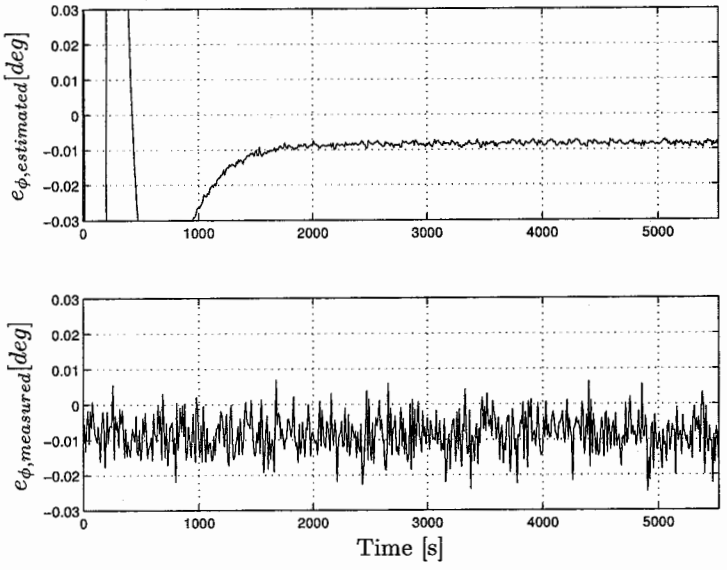


Figure 3.3: Difference roll estimation error and measurement error for the first orbit.

Chapter 4

Controller design

In this chapter new controllers will be designed and evaluated. It is hereby desired to be able to validate the quality of the new controllers with respect to the original, in view of tracking performance as well as stability and robustness. Simply looking at the tracking error will not give a clear impression of the last two, so another way has to be sought to determine these. This will be done in the next section.

4.1 Floquet Multipliers

Consider the continuous system described by equation 2.28, with the commanded control Torque:

$$\begin{aligned} \underline{T}_{cmd}(t) &= -L(t)\underline{q}(t) - L_{ff}(t)\underline{T}_{dist}(t) \\ \underline{q}(t) &= C_q\underline{x}(t) + D_q\underline{T}_{dist}(t), \end{aligned} \tag{4.1}$$

where $-L(t)\underline{T}_{dist}(t)$ is the feedback part of the controller and $-L_{ff}(t)\underline{T}_{dist}(t)$ the disturbance feed forward. For the original controller $L(t)$ was found using pole placement and the feed forward part was simply taken the same as the estimated disturbances with the opposite sign. In the original case the controlled states $\underline{q}(t)$ are the same as the dynamic states $\underline{x}(t)$, but this is not necessary.
From basic control theory it is known that (non-state dependant) feed forward has no influence on stability, only on the tracking performance. Therefore, it is possible to discard the term $-L_{ff}(t)\underline{T}_{dist}(t)$ for stability/robustness analysis. Using this, it is possible to substitute equation 4.1 in equation 2.28, resulting in:

$$\begin{aligned}\dot{\underline{x}}(t) &= \left[A - B(t)L(t)\left[I + D_q(t)L(t)\right]^{-1} C_q(t)\right]\underline{x}(t) \\ &= A_{cl}(t)\underline{x}(t) \\ \underline{y}(t) &= \left[C - D(t)L(t)\left[I + D_q(t)L(t)\right]^{-1} C_q\right]\underline{x}(t) \\ &= C_{cl}(t)\underline{x}(t).\end{aligned} \tag{4.2}$$

To obtain the Floquet Multipliers it is necessary to transform the continuous system to discrete and derive the eigenvalues of the closed-loop. One may wonder though, why it was not decided to use the eigenvalues of the continuous system for robustness/stability analysis as is normally done for time-invariant systems. To illustrate that doing so is essentially wrong, an example is presented from [9].

Example 1 *Consider the system:*

$$\dot{\underline{x}}(t) = A(t)\underline{x}(t), \tag{4.3}$$

with:

$$A(t) = \begin{bmatrix} -1 + \frac{1}{2}\cos^2 t & 1 - 1\frac{1}{2}\sin t\cos t \\ -1 - 1\frac{1}{2}\sin t\cos t & -1 + 1\frac{1}{2}\sin^2 t \end{bmatrix}. \tag{4.4}$$

The eigenvalues of this system can be found to be: $-\frac{1}{4} \pm \frac{1}{4}j\sqrt{7}$*, which seem to be stable and are not even time dependant!*
When computing the real response however, one will find that:

$$\underline{x}(t) = \begin{bmatrix} e^{\frac{1}{2}t}\cos t & e^{-t}\sin t \\ -e^{\frac{1}{2}t}\sin t & e^{-t}\cos t \end{bmatrix}\underline{x}_0. \tag{4.5}$$

So the real response is very instable and there is not even an indication of the earlier computed eigenvalues. This is the reason why one should not consider the eigenvalues of a time-dependant continuous system.

Using the *Tustin* method [10], the continuous closed-loop system, described by equation 4.2, is transformed to the discrete system:

$$\begin{aligned}\underline{x}_{n+1} &= A_n\underline{x}_n \\ \underline{y}_n &= C_n\underline{x}_n,\end{aligned} \tag{4.6}$$

with the same sample time as used in the *Simulink* simulations: 0.1 seconds.

Using equation 4.6, it can be found that any initial condition $\underline{x}_0$, will trigger a response in (discrete) time, given by:

$$\begin{aligned} \underline{x}_{n+1} &= \prod_{i=1}^{n}(A_i)\underline{x}_0 \\ &= S_n\underline{x}_0. \end{aligned} \tag{4.7}$$

The eigenvalues of S_n are called Floquet Multipliers (FM). If the FM remain within the unit circle (in the complex plane), the system will be globally stable. Furthermore, the FM in time can give an indication of both the rate of convergence as well as the robustness, by showing the proximity of the FM to the unit circle and how they change due to model deviations.

Testing the original PD-controller

Consider the reponse as given in figure 4.1, for the system with no disturbances (nor disturbance feed forward) and an exact match between commanded and actual torque:

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + BT_{act}(t) \\ &= A\underline{x}(t) + BT_{cmd}(t) \\ T_{cmd}(t) &= -L\underline{x}(t), \end{aligned} \tag{4.8}$$

but with an initial error of one degree on all states and the states estimated with the state-estimator of chapter 3.

Clearly, this system is asymptotically stable as the error converges to zero in about three orbits.
The same can be seen when computing the FM using equation 4.7, see also appendix C. The change of the FM in time is given in figure 4.2.

Furthermore, looking at the final eigenvectors it can be seen that there is a clear correlation between the FM and the angles:

$$\begin{aligned} N_1 = N_2 &= [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \\ N_3 = N_4 &= [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0] \\ N_5 = N_6 &= [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0], \end{aligned} \tag{4.9}$$

which also accounts for the similarity between the angles of figure 4.1 and the FM of figure 4.2. There are two differences though. The scattering of the FM, is

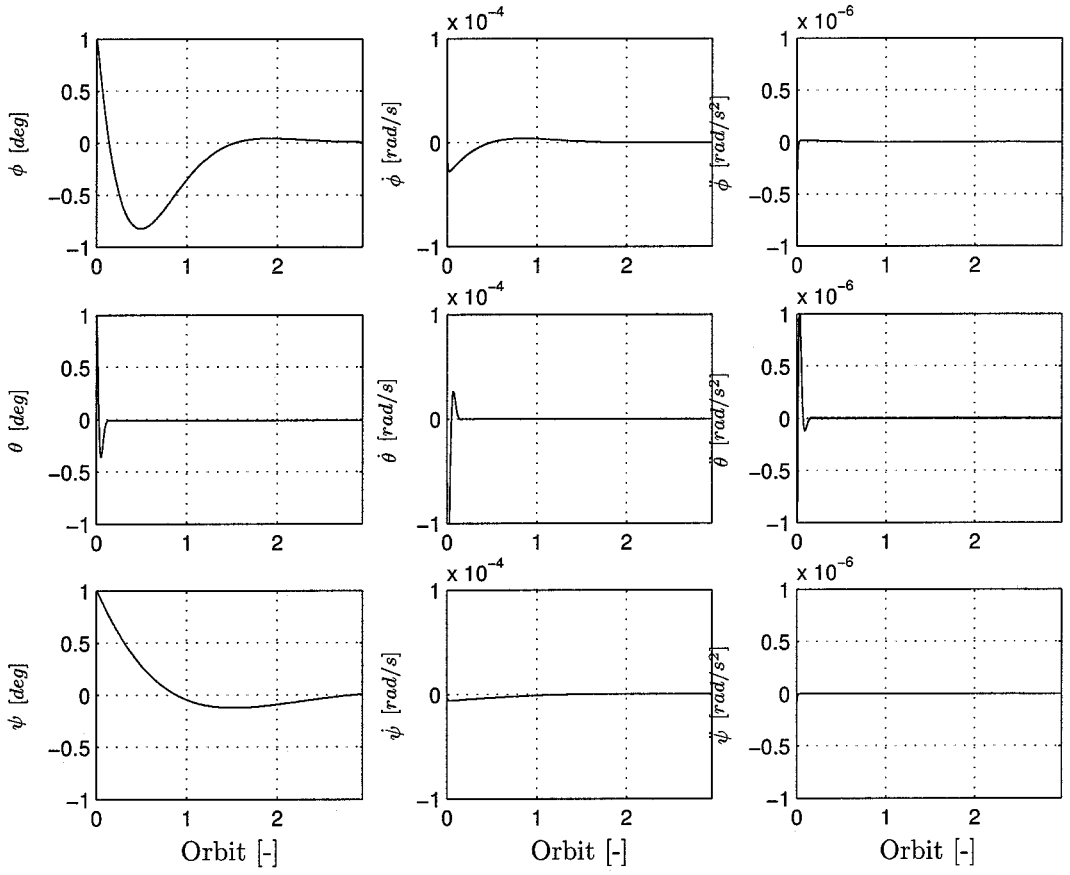


Figure 4.1: Response system with $T_{act} = T_{cmd}$ and 1 degree initial error.

caused by changes in the eigenvector representation by *Matlab* and the response difference is due to the fact that the influence of the estimator is not taken along in the FM computation.

Now lets take into account the real shape of the input matrix B(t), caused by the influence of the magnetic field.
Recall the state space representation:

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t), \tag{4.10}$$

with $u = T_{act}$ and:

$$\underline{T}_{act}^{scrf} = B_m^{\star}\underline{T}_{cmd}^{scrf}, \tag{4.11}$$

with $B_m^{\star}$ as given in equation 2.33.

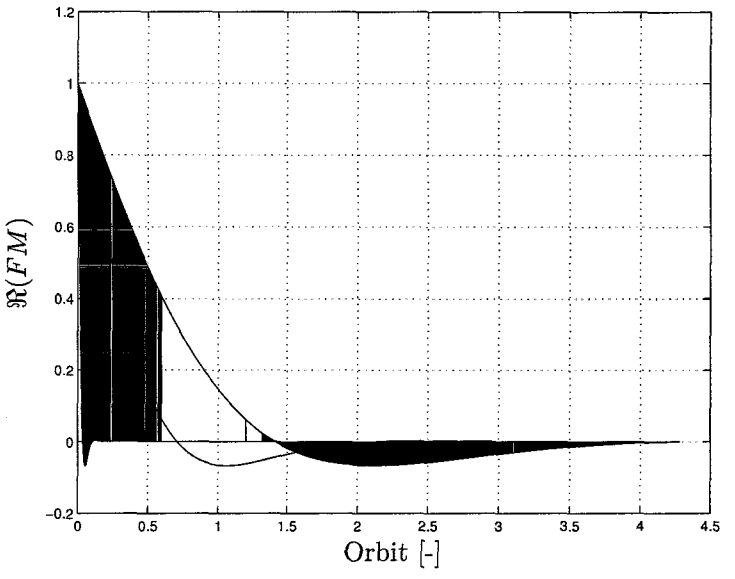


Figure 4.2: $\Re(FM)$: system with $T_{act} = T_{cmd}$ and 1 degree initial error.

Again a simulation has been done with no disturbances (or disturbance feed forward), but this time the differences between commanded and actual torque are taken along:

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + BT_{act}(t) \\ &= A\underline{x}(t) + BB_m^{\star}(t)T_{cmd}(t) \\ T_{cmd} &= -L\underline{x}(t), \end{aligned} \tag{4.12}$$

the response is given in figure 4.3.

It can be seen that for both the roll and yaw, the errors have increased after one orbit. Still, this is only the transient part of the response. On the long run the errors converge (slowly) to a near zero periodic solution.

The FM however, also show something else, see figure 4.3. Two of the FM become ± 1 once each orbit. This is due to the fact that each orbit there are two positions on which the satellite is not fully controllable, caused by the direction of the magnetic field.

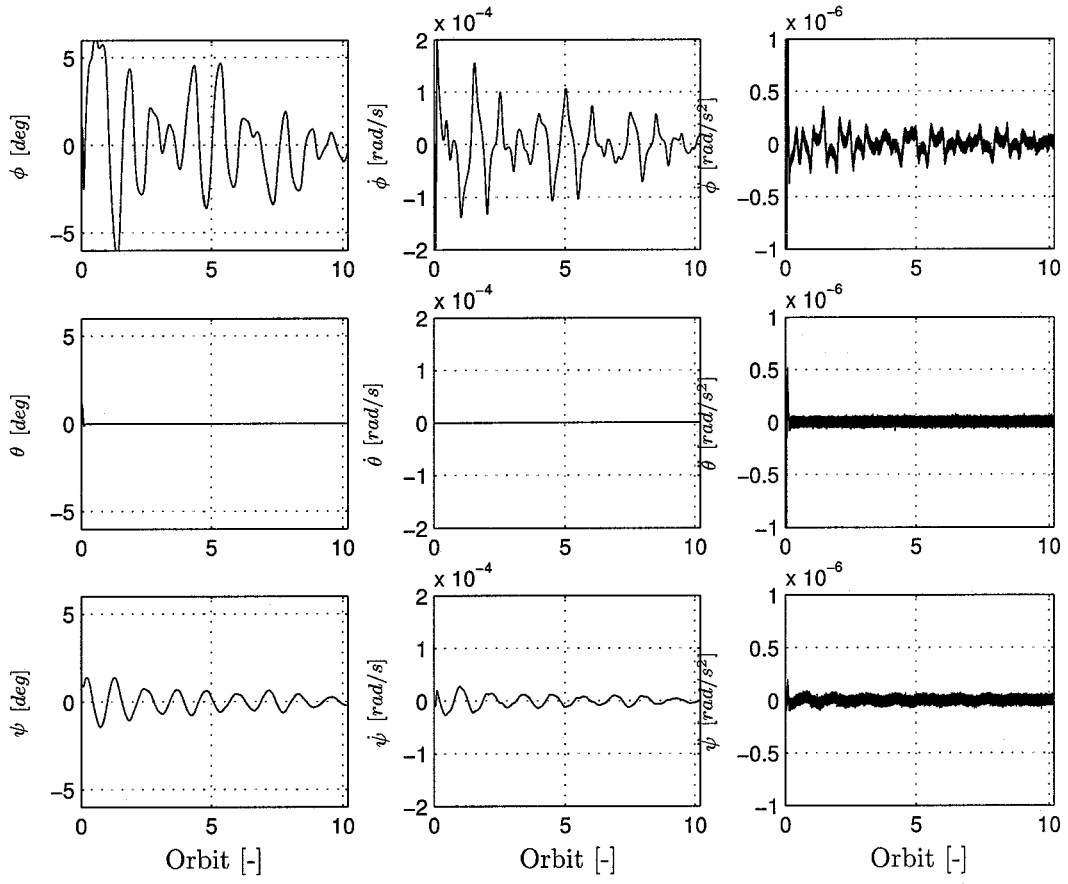


Figure 4.3: Response system with $T_{act}(t) = B_m^{\star}(t)T_{cmd}(t)$ and 1 degree initial error.

4.2 Enhancing the original PD-controller

As said before, the original controller consists of two parts: The feedback controller and a feed forward signal which consists of the estimated disturbance torques (with a minus sign). Lets forget about that feed forward signal for now and focus on the feedback controller.

From the step-response of the original system (*ESA* p.s. controller), without disturbances on a 1 degree error as given in figure 4.3, one can conclude that the poles of the closed-loop system differ greatly.
Due to the fact that the magnetic field influence was not taken along for its tuning (simplification), it was possible to split the original system in two sub-systems.

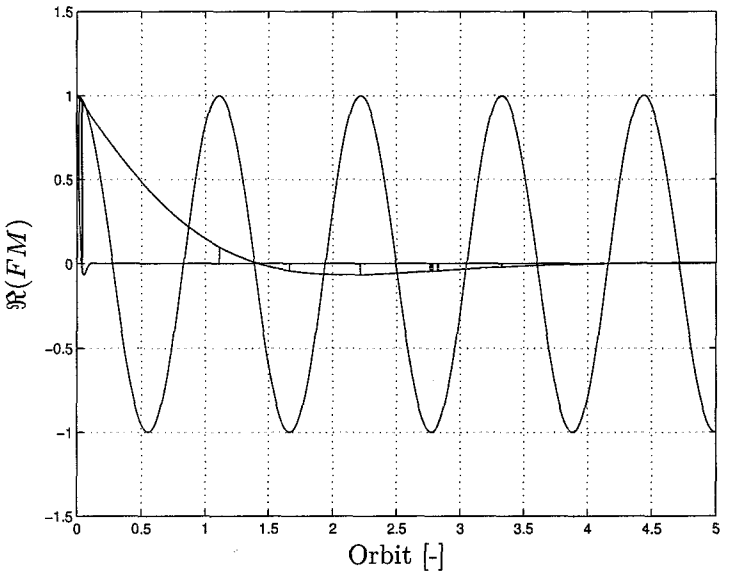


Figure 4.4: $\Re(FM)$: system with $T_{act}(t) = B_m^{\star}(t)T_{cmd}(t)$ and 1 degree initial error.

One SISO system containing the pitch equation and one MIMO system containing the roll and yaw equations, see also equation 2.26.
Tuning the pitch was relatively easy and the bandwidth of the SISO system could be placed as high as twenty times the orbital frequency without getting into any kind of trouble.
Tuning the roll and yaw was more difficult though, because of the strong connection between the two angles, as well as the un-modelled magnetic field influence, see equation 2.33. Therefore, the control engineers that build it were limited to placing the closed-loop poles of the MIMO-system at no more than half the orbital frequency, in order to keep the system stable.

It may be clear that under this restriction it is very hard to enhance the original controller (using the same techniques), as the closed-loop bandwidth is simply not enough to get results near the specifications, without using feed forward control. Feed forward however, is (generally speaking) not robust to non-modelled changes. Therefore, different control techniques need to be sought in order to increase the bandwidth.

4.3 Non-linear control: I/O-decoupling

The first thing to check for MIMO-systems like this one, is whether or not it is possible to decouple inputs and outputs. If so, the resulting control problem would be for 3 SISO-systems, which is easier to solve.
From [11] it is known that the I/O-decoupling problem has a solution when and only when the decoupling matrix $M(\underline{x})$ is non-singular. Herein, the decoupling matrix can be found by writing the system as:

$$\begin{bmatrix} y_1^{k_1} \\ \cdot \\ \cdot \\ y_m^{k_m} \end{bmatrix} = N(\underline{x}) + M(\underline{x})\underline{u}, \tag{4.13}$$

with k_i the relative degree of output i. So for the GOCE, this results in:

$$\begin{aligned} M(\underline{x},t)\underline{u}(t) &= J^{-1}(-B_{mag} \times M_{cmd}) \\ M(\underline{x},t) &= J^{-1}B_m, \end{aligned} \tag{4.14}$$

with:

$$B_m = \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix}, \tag{4.15}$$

or:

$$\begin{aligned} M(\underline{x},t)\underline{u}(t) &= J^{-1}B_m^{\star}\underline{T}_{cmd} \\ M(\underline{x},t) &= J^{-1}B_m^{\star}, \end{aligned} \tag{4.16}$$

with $B_m^{\star}$ as is given in equation 2.33, depending on the chosen control strategy. In both cases $M(\underline{x},t)$ will not be regular over a whole orbit, so I/O-decoupling is not possible.

4.4 Linear Quadratic Regulator (LQR)

The idea behind the LQR is that of using optimalisation techniques to find the most optimal controller, for each moment in time, over a certain time period. To be able to compute this optimal controller, it is absolutely necessary to have a model which accurately describes the real world over the chosen time-span.
Assuming the derived satellite model is accurate enough, an attempt shall be made to get the responses within the specification bounds, using a LQR. The LQR shall be computed using a continuous approach [12], as the dynamics model is also given in continuous form.

4.4.1 LQR computation

Again consider the continuous system described by equation 2.28, with the controlled outputs:

$$\underline{q}(t) = C_q\underline{x}(t) + D_q\underline{u}(t), \tag{4.17}$$

and $\underline{u}(t) = \underline{T}_{act}(t)$, where the controlled outputs are the angles, velocities and accelerations. Rewriting this to a magnetic moment based control system gives:

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + BB_m\underline{M}_{cmd}(t) \\ \underline{y}(t) &= C\underline{x}(t) + DB_m\underline{M}_{cmd}(t) \\ \underline{q}(t) &= C_q\underline{x}(t) + D_qB_m\underline{M}_{cmd}(t), \end{aligned} \tag{4.18}$$

with optimal control-action:

$$\underline{M}^o_{cmd}(t) = -L^o(t)\underline{x}(t). \tag{4.19}$$

Substituting this optimal control action in equation 4.18, it can be found that:

$$\begin{aligned} \underline{q}(t) &= C_q\underline{x}(t) - D_qB_mL^o(t)\underline{x}(t) \\ &= E(t)\underline{x}(t). \end{aligned} \tag{4.20}$$

Now consider the quadratic cost-function:

$$min\ J = \int_{t_0}^{t_e} [\underline{q}^T(t)W(t)\underline{q}(t) + \underline{M}^T_{cmd}(t)R(t)\underline{M}_{cmd}(t)]dt + \underline{q}^T(t_e)W_e\underline{q}(t_e)$$

$$\begin{aligned} &subject\ to: \\ &\quad dynamics\ equations\ 4.18\ and\ 4.19; \\ &\quad R(t)\ symmetric,\ possitive\ definite; \\ &\quad W(t),\ W_e\ symmetric,\ non-negative\ definite. \end{aligned} \tag{4.21}$$

Herein, $R(t)$, $W(t)$ and W_e are weighing matrices which determine the nature of the optimum. Although anything can be chosen for these matrices (within the constraints), constant diagonal matrices will be used to facilitate the tuning procedure later on.
Using this and equation 4.20, equation 4.21 resolves to:

$$
\begin{aligned}
&min\ J = \int_{t_0}^{t_e} [\underline{x}^T(t)Q(t)\underline{x}(t) + \underline{M}_{cmd}^T(t)R(t)\underline{M}_{cmd}(t)]dt + \underline{x}^T(t_e)P_e\underline{x}(t_e) \\
&subject\ to: \\
&\qquad dynamics\ equations\ 4.18\ and\ 4.19; \\
&\qquad R\ symmetric,\ possitive\ definite; \\
&\qquad Q(t) = E^T(t)WE(t)\ symmetric,\ non-negative\ definite; \\
&\qquad P_e = E^T(t_e)WE(t_e)\ symmetric,\ non-negative\ definite.
\end{aligned} \tag{4.22}
$$

From [12] it is known that when optimal control is applied (equation 4.19) the minimal cost over the time interval $(t,\ t_e)$ can be represented by:

$$J^o(t) = \underline{x}^T(t)P^o(t)\underline{x}(t). \tag{4.23}$$

Furthermore, after some terrible mathematics the (matrix) Riccati differential equation can be derived from equation 4.22. This derivation shall be omitted here, but for more information the reader is referred to [12]. The Riccati equation:

$$
\begin{aligned}
-\dot{P}^o(t) &= A^T P^o(t) + P^o(t)A + Q(t) - P^o(t)B^\star(t)R^{-1}B^{\star T}(t)P^o(t) \\
P^o(t_e) &= P_e,
\end{aligned} \tag{4.24}
$$

with $B^\star(t) = BB_m(t)$, can be solved using back-stepping after which the optimal control action can be computed by:

$$
\begin{aligned}
L^o(t) &= R^{-1}B^{\star T}(t)P^o(t) \\
M^o_{cmd}(t) &= -L^o(t)\underline{x}(t).
\end{aligned} \tag{4.25}
$$

4.4.2 LQR simulation and tuning

The implementation of the Riccati equation and the back-stepping procedure in *Matlab* is given in appendix A.
Due to the fact that the sample frequency of the *Simulink* simulation (and controller) is fixed at 0.1 seconds, the first thing to do is find the necessary time spacing for the $L^o(t)$-matrix. This was found to be 10 seconds.
At that interval rate any changes in the magnetic field are small enough to give only minimal changes in the optimal controller parameters. So the tracking error, when sampling with a 10 second interval, is almost the same as when sampling with the 0.1 second interval, which is certainly not the case when using a (for instance) 50 seconds controller time interval.
One may wonder why it was not decided to simply use a 0.1 seconds time interval

for the $L^o(t)$-matrix. The answer is that the computation of the back-stepping procedure with that kind of time-steps would take about one day in real time and make the control parameter matrix unnecessarily large.
Furthermore, it was chosen to use a time interval of 10 orbits, under the assumption that simulating 10 orbits would prove enough to visualize the steady-state controller behavior. For the onboard implementation however, a time interval of more than 24 hours needs to be taken, due to the nature of the Earth magnetic field, see section 2.4. More on this in section 5.2.

The next question is what should be chosen for the diagonal weighing matrices R and W. To get an initial guess for W, the specifications were used as norm, to equally weigh all the to be controlled parameters:

$$q_i^{spec^T} W_{ii} q_i^{spec} = 1, \tag{4.26}$$

which gives:

$$W = \begin{bmatrix} \frac{1}{10^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{5^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{(3.10^{-5})^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-3})^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{(3.10^{-5})^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-6})^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-6})^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-6})^2} \end{bmatrix}. \tag{4.27}$$

Furthermore, for R the identity matrix was used initially as it was unclear how heavily the inputs should be weight.
Using these initial settings the back-stepping procedure was found to converge to an optimal solution. This solution however, still had a reasonably slow step response. Therefore, the input weighting matrix R was tuned till the convergence boundary was reached. Further lowering R would result in a diverging back-stepping procedure so no optimal solution would be found.
The critical R was found to be:

$$R = \begin{bmatrix} \frac{1}{10^2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{3^2} \end{bmatrix}. \tag{4.28}$$

Nevertheless, even with these settings the specifications on the velocities could not be reached. Therefore it was necessary to give some extra weighing to the

velocities and angles and less on the accelerations, which were far below the specifications.
The resulting matrices are given in equations 4.29 and 4.30:

$$W = \begin{bmatrix} \frac{1}{(1.10^{-3})^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{(5.10^{-3})^2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{(5.10^{-3})^2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{(5.10^{-6})^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-5})^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{(5.10^{-5})^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(5.10^{-4})^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(3.10^{-4})^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{(1.10^{-3})^2} \end{bmatrix} \tag{4.29}$$

$$R = \begin{bmatrix} \frac{1}{10^2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2^2} \end{bmatrix}. \tag{4.30}$$

Using an implementation of this LQR-controller in *Simulink* (see appendix B), it was possible to check its step response on a 1 degree initial error, without disturbances (nor disturbance feed forward), as was done earlier for the original controller (recall figure 4.3). The results are given in figure 4.5.

As can be seen the step response of the system controlled by the LQR-controller is significantly better than that of the original controller. Note the different magnitude scales and time span.
The same can be concluded when looking at the Floquet Multipliers, see figure 4.6. It may be clear that convergence is much faster, compared to the FM given in figure 4.4 and the LQR reduces all FM to zero, despite the two positions of almost zero controllability.

Furthermore, when looking at the disturbance rejection of this controller, it can be seen that even without disturbance feed forward the tracking errors are within the specification bounds, see figure 4.7 (specs are the solid lines).
Due to the periodic nature of the errors however, caused by the periodic varying controllability, the results with respect to the frequency specifications are no better than those found by the original controllers. As that periodic nature of the response is very hard to change, it was decided not to pay to much attention to these specifications.

Note however, the slightly worse response at the end of the ten orbits. This is

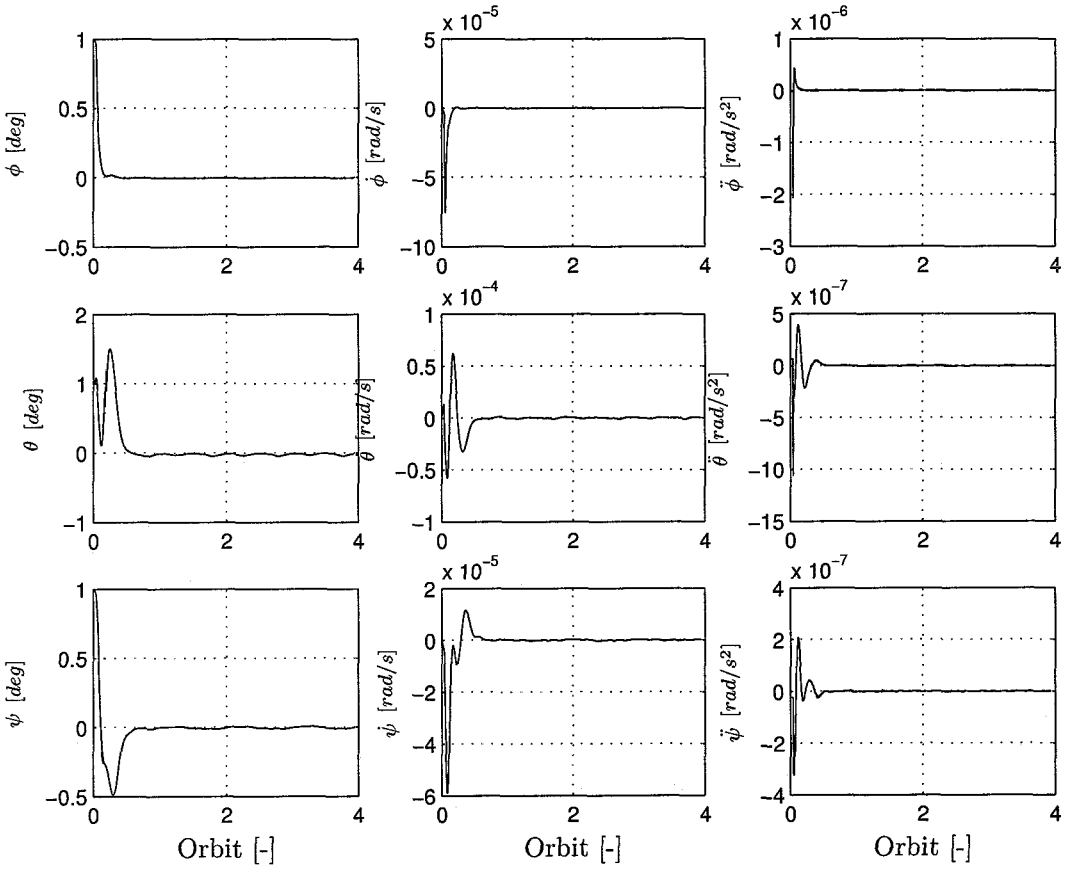


Figure 4.5: Response system with final LQR-controller and 1 degree initial error.

caused by the far from perfect estimation of P_e. It is known that the estimation of P_e is not the most optimal value of $P(t_e)$, but on the long run $P(t)$ will converge to its optimal value, as long as feasibility permits it. This process of conversion can be seen very well when looking at the diagonal values of $P(t)$ in time, see figure 4.8. This is also the reason why it is imperative to compute the optimal controller, using a longer time span than will be used in the real world.
Note that, to ensure that the found optimum is the global optimum (for these weighting matrices), several values of $P(t_e)$ have been tried.

In figure 4.8, it can be seen that the diagonal values of $P(t)$ show spikes with a frequency of $2\omega_0$. These spikes coincide with the points of lesser controllability, where the height of the spike is determined by both the chosen values for the control weighting matrix R, as the exact direction of the magnetic field.

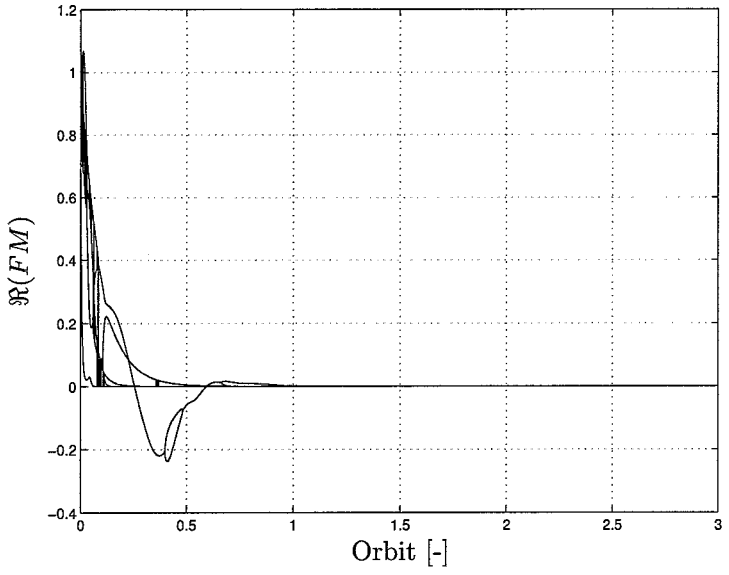


Figure 4.6: $\Re(FM)$: system controlled with LQR.

If the magnetic field direction would be exactly in e_1^{SCRF}-direction and there would be no input weighting, the spikes would go to infinity. The magnetic field will never be exactly in that direction however and there is input weighting, so the spikes will be limited to certain maximum values.
These spikes also appear in the control parameters, some of which are given in figure 4.9. This was to be expected though, as the only way to cope with a point of lesser controllability lies in adding additional control effort.

In an effort to further enhance the performance disturbance feed forward will now be added to the control action.

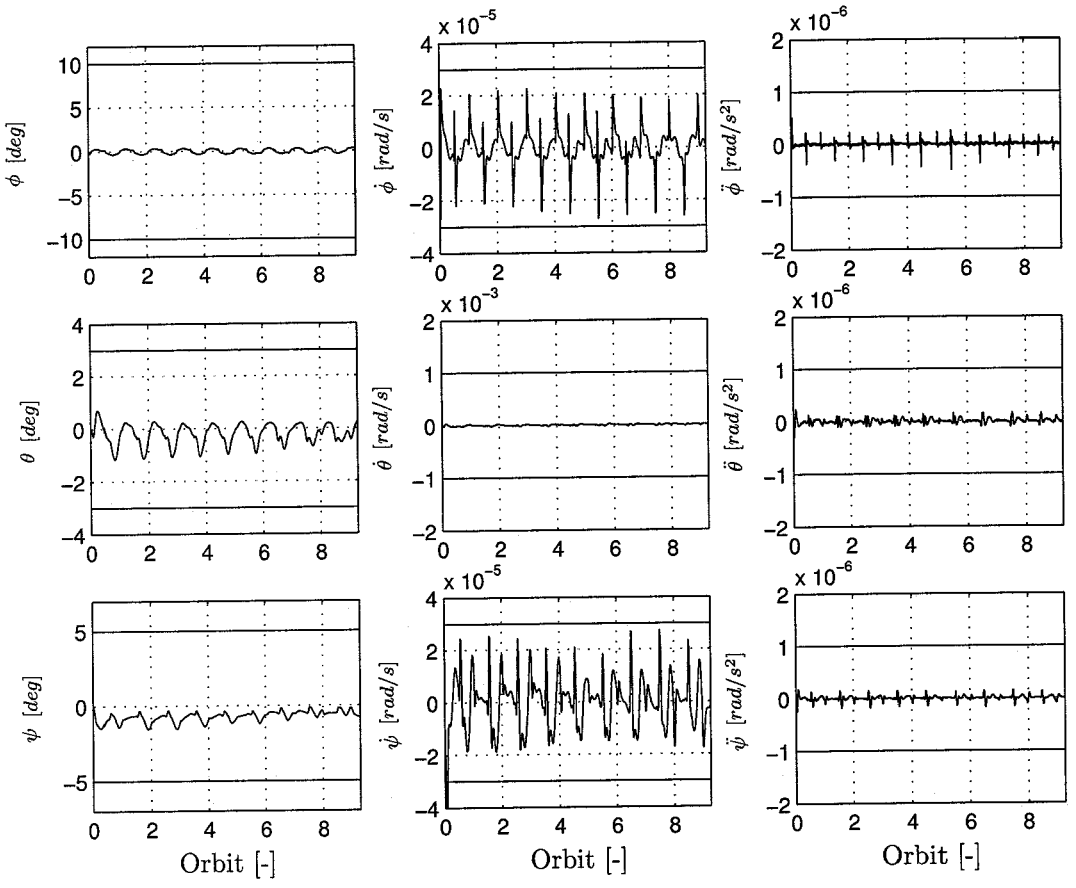


Figure 4.7: Tracking error obtained by LQR-controller.

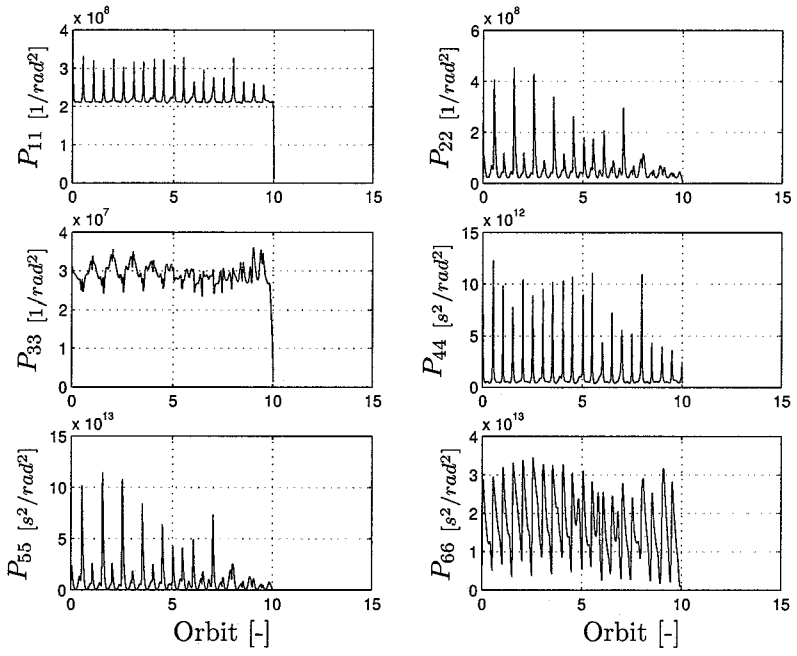


Figure 4.8: P(t) main diagonal components.

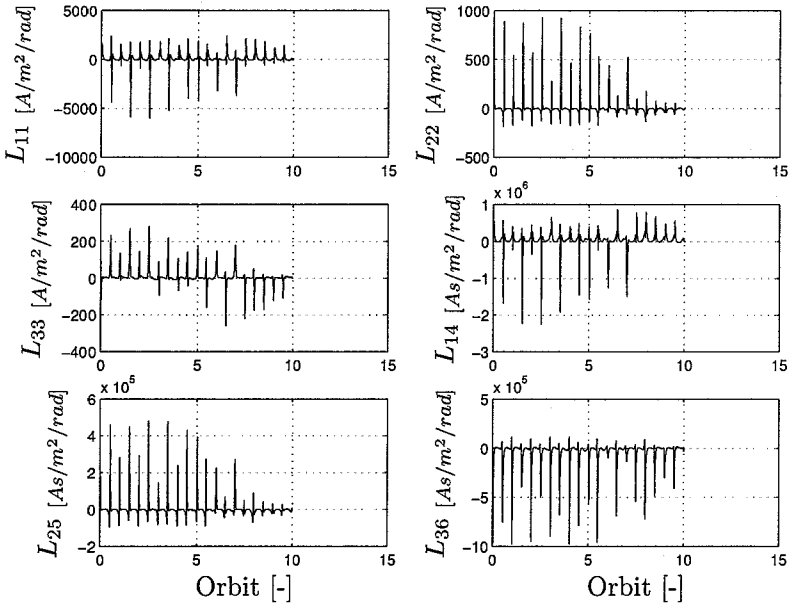


Figure 4.9: L(t) diagonal components.

4.5 Feed forward design

The original controller used a feed forward signal opposite to the estimated disturbance torques to greatly enhance the tracking results. Lets try the same, but now using the LQR-controller for feedback. The results are given in figure 4.10.

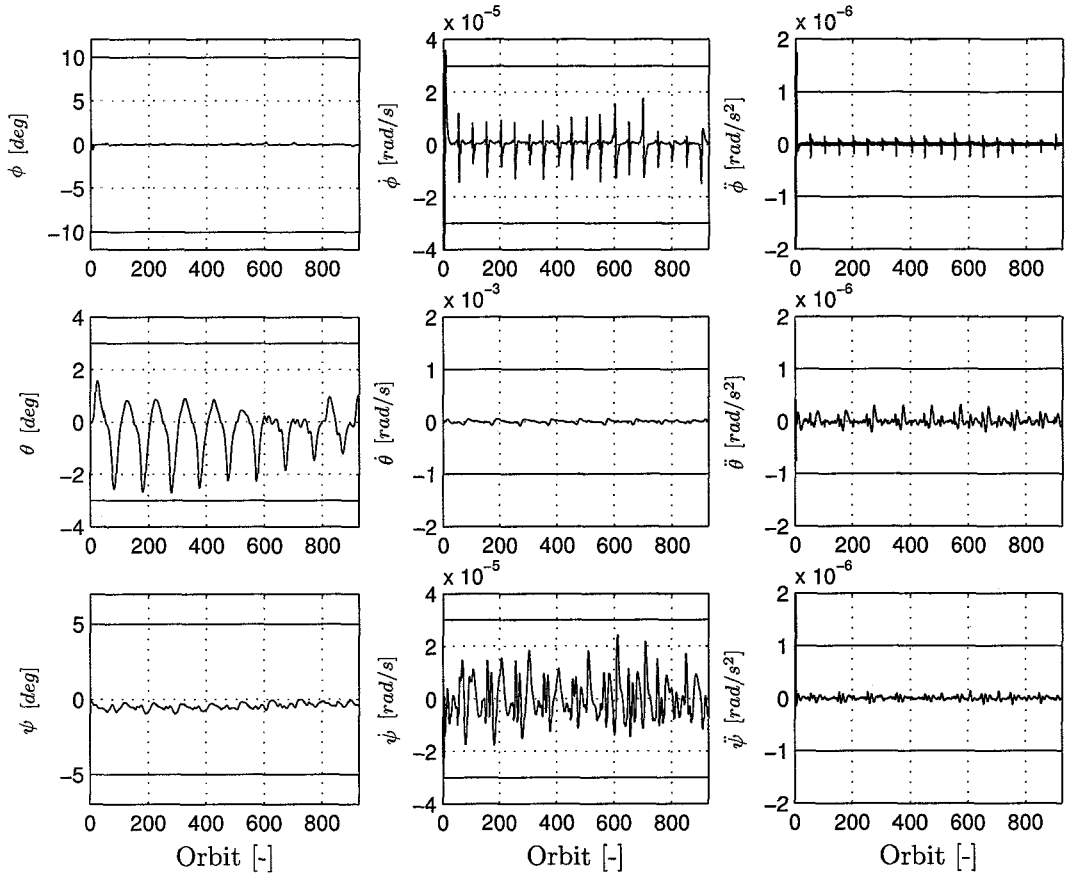


Figure 4.10: Tracking error obtained by LQR-controller with original feed forward.

It can be seen that, even though the result is enhanced (except for the pitch), the improvement is far less than what was obtained by adding the feed forward to the original controller. This is mainly caused by the fact that the LQR is capable of suppressing the (reasonably low-frequent) disturbances much better due to its higher bandwidth.
However, this feed forward is still not optimal, as the used feed forward signal is not exactly the same as the disturbance torques (recall equation 2.33). This is

especially clear at places with reduced controllability, where the error still spikes, but also explains the increase in pitch error.
Finding an optimal feed forward signal is not so easy however. One way of doing this is by implementing Iterative Learning Control (ILC) [13]. The problem with ILC however, is the fact that huge control spikes will appear when the system is not controllable, so those spots in the orbit should be left out. Doing so, the control result with ILC might enhance the control results even further, but the complexity of that controller will probably prevent it from being taken along in space. That is why the original feed forward will continue to be used.

Chapter 5

Controller validation

In the previous chapter it was shown that the LQR is capable of greatly enhancing the tracking results. However, the controller which will ultimately be implemented on the GOCE, must also be robust and implementable.
Whether or not the LQR is robust and easy to implement is going to be checked in this chapter.

5.1 LQR Robustness

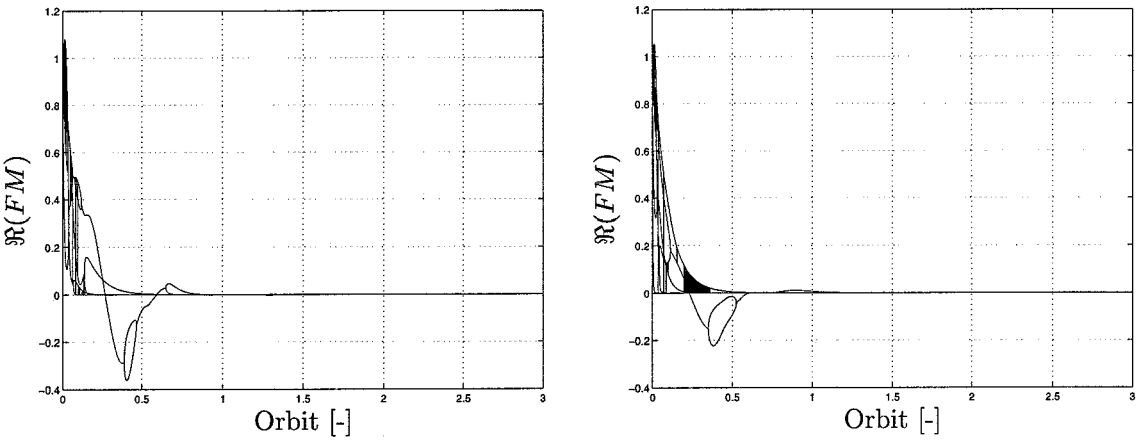
In this section the robustness of the LQR controller (with original feed forward) will be checked for three kind of structural changes:

1. Changes in the spacecraft Inertia;
2. Changes in the disturbance simulation;
3. Changes in the magnetic field (noise and trend moves).

5.1.1 Inertia deviations

Prior to launch, the satellites Inertia is reasonably well known. However, on its mission, GOCE will be using fuel which not only changes the mass, but also to Center of Mass. So, the Inertia matrix will change (slightly).
In order to analyze the robustness of to controller to these changes, the Floquet Multipliers have been computed for changes of the Inertia matrix of ± 50 percent (which of course, is huge compared to the real world changes). The results are given in figure 5.1. Recall that for the standard system, the FM were as shown in

figure 4.6.



(a) (b)

Figure 5.1: Inertia deviations: (a) +50%; (b) −50%.

As can be seen, a 50 percent Inertia variation will not endanger the stability of the controller, as it only gives a minimal change in the FM. This leads to the conclusion that the controller is quite robust to Inertia changes.

5.1.2 Disturbance deviations

Deviations of the simulated disturbances with respect to the real disturbances should not cause any problems for the feed back controller as long as the angles remain small.
There is a small catch here however due to the fact that the estimator uses a model of the disturbances which may be less accurate when the disturbances change. So, even though the feedback controller is not directly linked to the disturbances, the disturbances do influence the stability through the estimator.
This link has not been taken along for the computation of the Floquet Multipliers, so validating the robustness of the LQR to disturbance changes should be done by computing the time response.

Furthermore, it is written in [6] that: *For spacecraft below approximately* 400 *km, the aerodynamic torque is the dominant environmental disturbance torque.*
When comparing this to the initial disturbance estimations given in 2.1 which do

resemble the *Simulink* disturbance implementation, it may be concluded that the aerodynamic torque might be modelled a bit to small, even though the GOCE is designed to have a very low drag coefficient.
For the robustness analysis, the aerodynamic torque will be multiplied by a factor 100, making it the dominant torque of order 1.10^{-3} [Nm] in $\vec{e}_2^{scrf}$- and $\vec{e}_3^{scrf}$-direction. The resulting disturbance torque with respect to the SCRF is shown in figure 5.2 (compare with figure 2.5). Furthermore, the time response using this disturbance torque (for the first four orbits) is given in figure 5.3.

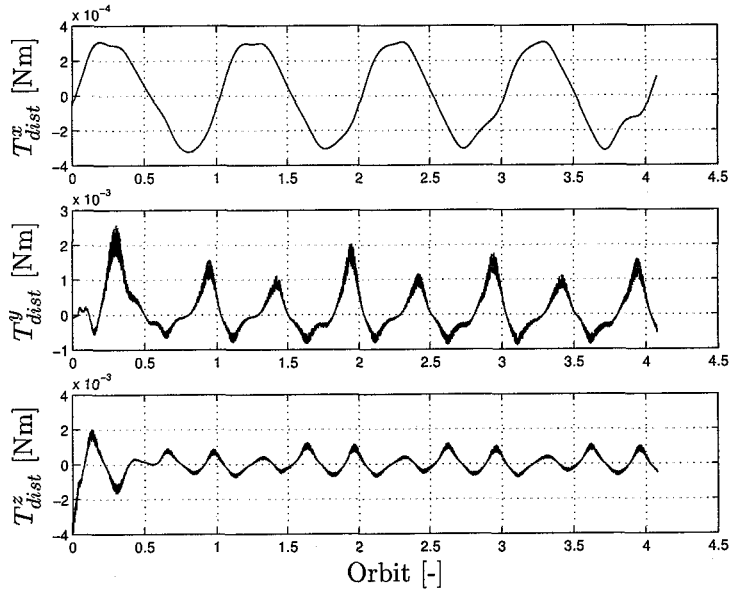


Figure 5.2: Simulated disturbances (without T_{gg}^{scrf}), with increased aero dynamic torque.

It can be seen that the controller has a bit more trouble suppressing the additional disturbance torque, but remains nicely stable. So, the LQR (with estimator) is also sufficiently robust to deviations in the disturbance model.
It should be noted though that, when doing the same with the system controlled by the *ESA* p.s. controller, the system goes instable, simply because the feedback part cannot suppress the huge addition to the disturbance torque.

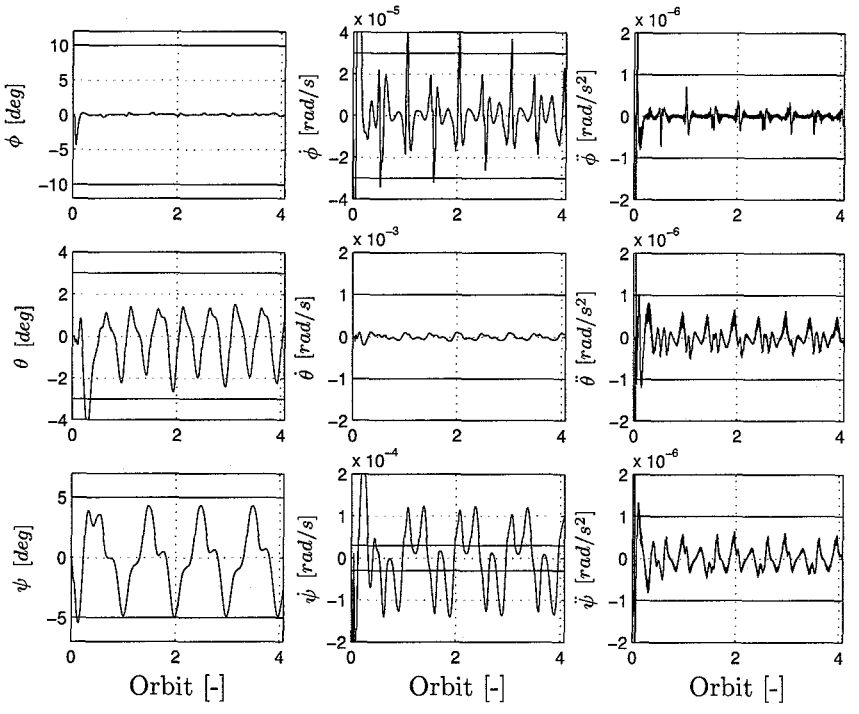
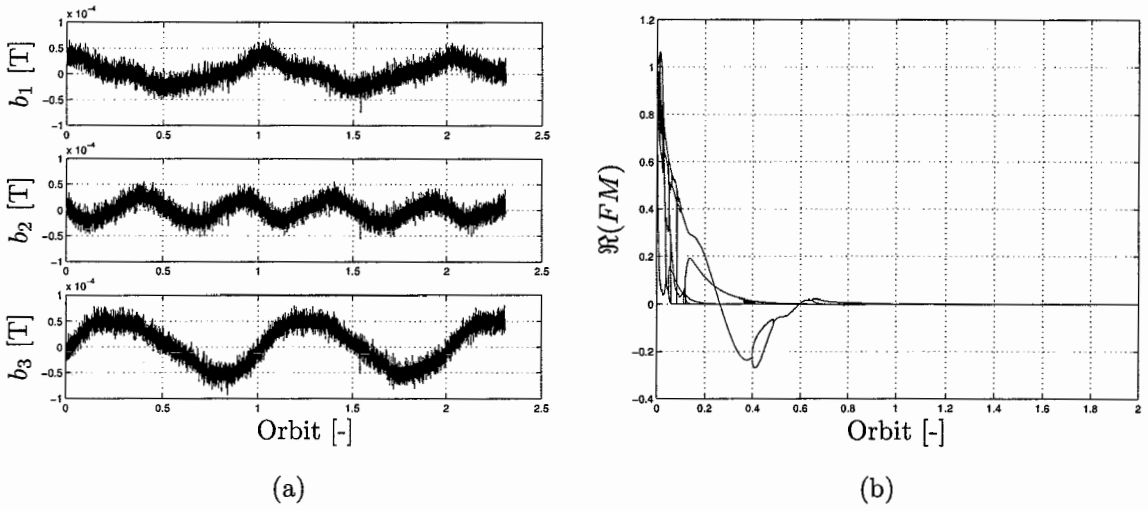


Figure 5.3: Tracking error obtained by LQR-controller with original feed forward and increased aerodynamic torque.

5.1.3 Deviations of the magnetic field

In the *Simulink* simulation there is no measurement noise on the magnetic field measurement, only two samples delay. One may wonder however, how stable the LQR controlled system is to measurement noise on the magnetic field measurement. Therefore, random noise has been added to the magnetic field measurement. The measurement of the magnetic field as well as the computed Floquet Multipliers are given in figure 5.4. It shows that despite the additional measurement noise, the system remains stable.

In chapter 2 it was said that an IGRF 2000 model of order eight was used as model for the Earth magnetic field.
In order to check the robustness too model deviations, the LQR parameters were recomputed with a magnetic field model of order 2. Using that controller, the FM were determined using a model of order 8 to generate the magnetic field measurements. The resulting FM are given in figure 5.5.



(a) (b)

Figure 5.4: Magnetic field with noise (a) and the corresponding FM (b).

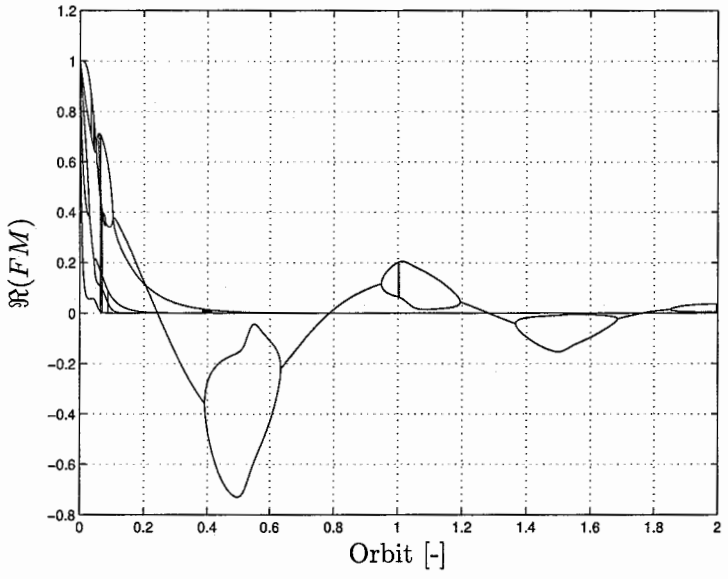
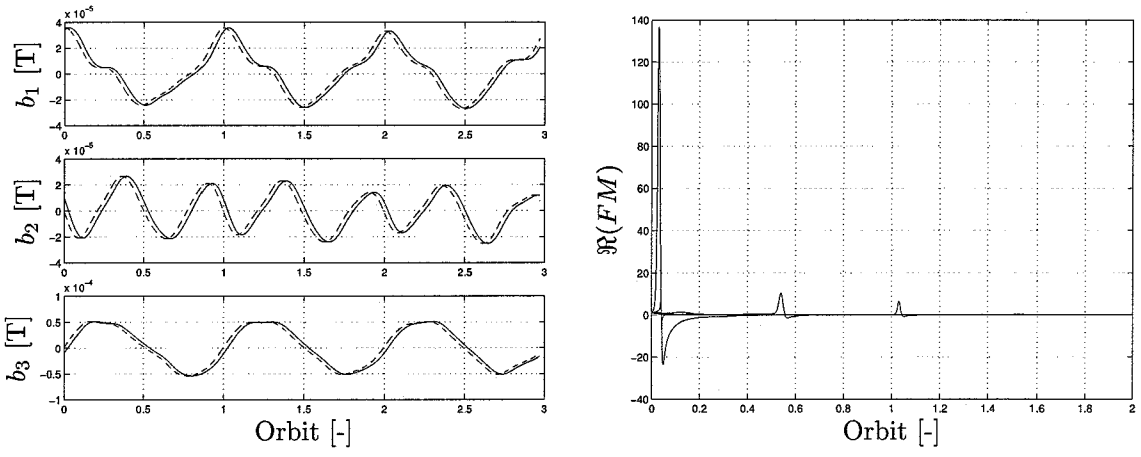


Figure 5.5: Magnetic model deviation: LQR with IGRF order 2, FM computation with order 8.

It can be seen that again the system remains nicely stable, although the oscillation is larger than before.

There is another possible deviation of the magnetic field though.
For the computation of the LQR a model of the magnetic field is used. Using this model, a value of the magnetic field is obtained for each point in time, giving the optimal controller for a small time period.
Doing so, it is assumed that the position of the satellite is exactly the same as the predicted position, which will not be the case in reality. Therefore, the LQR must also be robust to shifts in time of the real magnetic field with respect to the modelled one (phase differences between the trend of the modelled magnetic field and the actual one).
It should be clear that due to the large control spikes at the times of negligible controllability, this will not be the case. It was found that for trend differences below 150 seconds (on an orbit of 5400 seconds), the system remained stable. The Floquet Multipliers for this maximum trend difference are given in figure 5.6.



(a) (b)

Figure 5.6: Magnetic field with (max.) trend shift (a) and the corresponding FM (b).

It can be seen that the system is not globally stable, but still converges to a stable solution after some time.
So, it can be concluded that although the LQR is very robust to most deviations, it is reasonably sensitive to shifts in the trend of the magnetic field model with respect to the real world. As these deviations are mainly caused by differences between the predicted and actual position on the orbit, this LQR implementation needs (for stability) the additional constraint that the actual position and the predicted position remain within 1000 km, which will be the case in practice.
However, if a way would be found to compute the controller parameters as a function of the magnetic field values, instead of a function of time, this constraint

would disappear. More on this in the next section.

5.2 LQR Onboard implementation

So far, all simulations have been done with a LQR controller which was computed over a time span of 60.000 seconds. This sufficed as all simulations were limited to a time span of no more than 50.000 seconds. However, in the real world there will be no such limitation. It may be clear that it is therefore imperative to find a periodicity within the controller, as it is impossible to upload the controller settings for the complete mission on the satellite before departure. Furthermore, due to the obvious link between the magnetic field and the controller, it will be necessary to look at the magnetic field to find the needed period length.
From figure 2.7, it can be seen that the magnetic field's lowest harmonic, is a result of the daily rotation of the Earth. On top of that there is another harmonic caused by the rotation of the satellite around the Earth. So, under the assumption that the satellite will orbit the Earth in a round number of orbits each day (which will not be the case in practice), it will be possible to use this as a period length for the controller.
Doing so however, it will still be necessary to upload a matrix with $6 \times 3 \times 24 \times 360 = 155.520$ elements, each time the daily controller parameters change (for example when the round number of orbits is not reached).

Another possibility would be only to use the orbital period of the satellite for the controller length and reuse it for each new orbit, assuming the influence of the lowest harmonic (the rotation of the Earth) to be negligible.
This was attempted, but the system proved to be instable after 8 orbits, as the points of lesser controllability had shifted to much with respect to those modelled.

So, despite the dominant influence of the orbital frequency, one can not simply use it to obtain a stable set of control parameters which is periodically usable. However, the only reason this cannot be done, is because the unmodelled harmonics resemble a trend shift which surpasses the constraint (previous section). If however, the parameters of the controller were to be given as a function of the magnetic field, instead of the time, this problem would no longer exist.
In order to find this relationship several high polynomial fits have been made of the magnetic field on the control parameters, but none proved accurate enough to keep the system stable.

Recall figures 4.9 and 4.8. From these it can be seen that the control parameters

$L(t)$ fluctuate quite heavily and become positive as well as negative, but that the diagonal values of the Riccati-matrix $P(t)$ remain above zero and have a nice periodic behavior. Therefore, it might be a good idea to make a fit on the values of $P(t)$ instead of $L(t)$ and to use that fit to compute the values of $L(t)$.

Recall that:

$$L(t) = R^{-1}B^{\star^T}(t)P^o(t), \tag{5.1}$$

with:

$$R^{-1}B^{\star^T}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & rb_{12} & rb_{13} \\ 0 & 0 & 0 & rb_{21} & 0 & rb_{23} \\ 0 & 0 & 0 & rb_{31} & rb_{22} & 0 \end{bmatrix}. \tag{5.2}$$

Therefore, it is only necessary to make a fit on P_A and P_B when:

$$P = \begin{bmatrix} 0(3) & 0(3) \\ P_A & P_B \end{bmatrix}. \tag{5.3}$$

The relevant values of $P(t)$ are given in figure 5.7.

Several fits were made to obtain $P(Bmag)$, but again none proved accurate enough to obtain a stable set of control parameters.

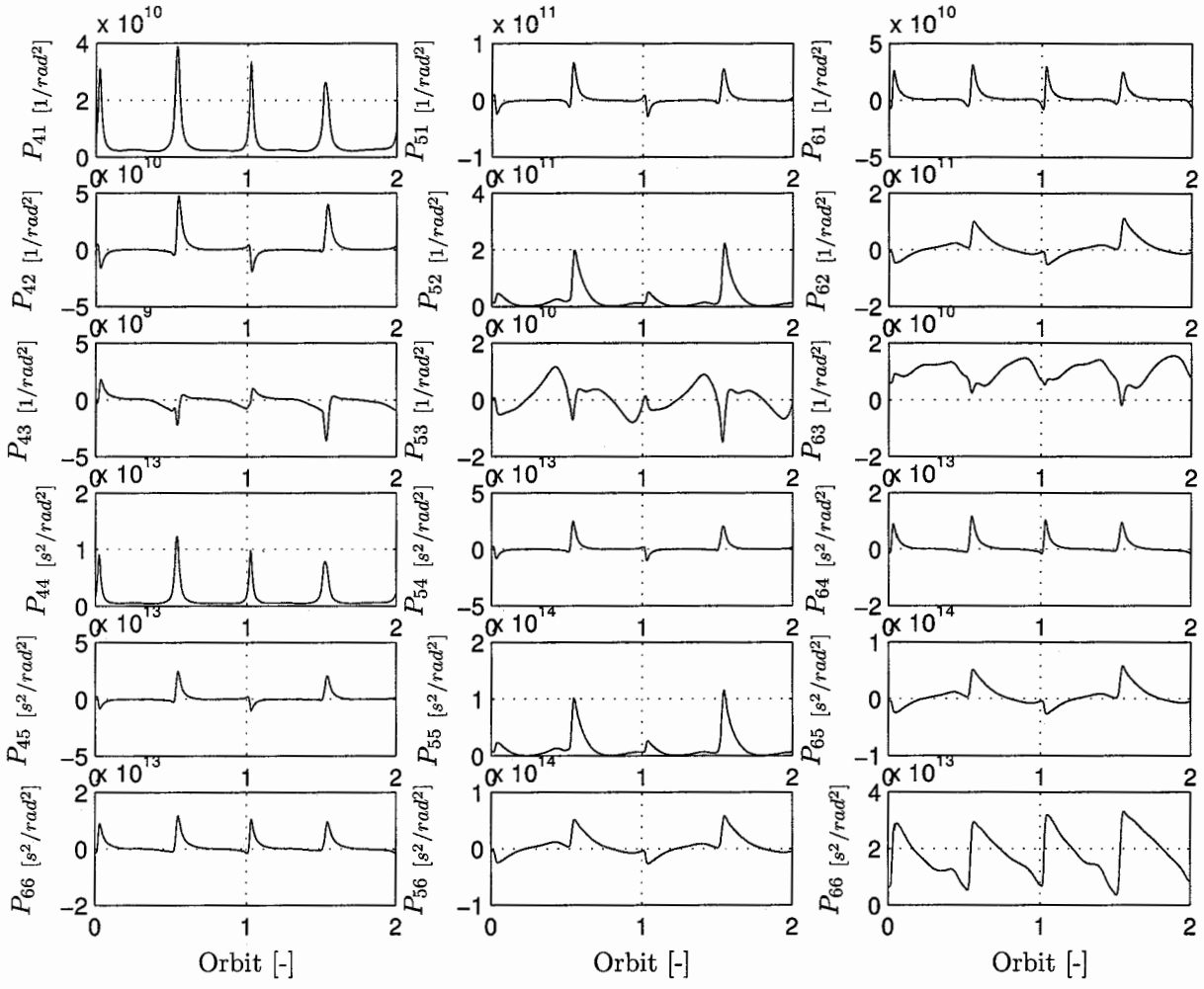


Figure 5.7: P(t) relevant fitting components.

Chapter 6

Conclusion and recommendations

In the previous chapters an analysis has been made of the satellite- and disturbance dynamics. Using this analysis, the estimator was redesigned. However, the final estimation error was of the same magnitude as that obtained by the original *ESA* preliminary study-estimator. Clearly, the original estimator was already quite optimal.
With the newly designed estimator, an effort was made to enhance the original controller, using the same techniques. The original controller consists of a time independent feedback part, which gives a very low system bandwidth and a very strong feed forward part, which is used to get near the specifications. It is impossible to raise the bandwidth of the feedback controller though, as that results in an instable system due to the system's periodic controllability variations.
At that stage, two possibilities for performance enhancement were available: Enhancement of the feed forward part and/or enhancement of the feedback part using other techniques. Due to the fact that pure feed forward controllers are not as robust as feedback controllers, it was decided to start with the second option. After determining that it was not possible to use input-output decoupling, a time dependant feedback controller was designed using LQR-techniques. Furthermore, the original feed forward action was also added to further enhance the performance, even though it became clear that the enhancement was far less than what was obtained for the original controller.
The LQR proved to be able to greatly enhance the performance and is also reasonably robust to most model deviations. The only exception however, are trend deviations of the magnetic field, which may result to instabilities when they grow to large. Unfortunately, it is also this sensitivity to trend deviations that complicates the onboard implementation of the controller. In order to remain stable, it was found necessary to use a time period of more than one day for the LQR computation, which is the lowest harmonic of the Earth magnetic field and have the same satellite position on orbit at the beginning and end of the daily period.

The resulting controller matrix was quite large however, making it hard to quickly upload new controller parameters, if that were to be needed.
In order to solve this problem, an attempt was made to find a relation between the magnetic field and the optimal control parameters, which would eliminate the trend sensitivity. However, no relation could be found that was accurate enough.

So, the objective of this study was to: *find a better controller which is sufficiently robust and simple to implement on board the satellite by taking into account the periodic roll/yaw exchange of actuation capability, induced by the magnetic field change along the orbit.*
It can be concluded that this controller has been found in the form of the LQR as designed an tested in chapters 4 and 5. This controller may be implemented as it is now, as it greatly enhances the performance of the original controllers. Still, this controller has a couple of down sides when it comes to robustness to trend changes in the magnetic field and implementation/operation simplicity.

In order to further enhance the designed LQR, it is recommended that an effort is made to find an *accurate* relationship between the magnetic field and the optimal controller parameters. Doing so, the controller may be given more robustness and made better updatable/usable in comparison to the original LQR.

Besides that, it is also recommended that the LQR controller designed in chapter 4 is tested in respect to the full non-linear model of the system (so far a linear approximation was used) and that the computational strain of the controller to the satellite is measured. Doing so, the quality of the controller and the possibilities for its onboard implementation may be further discovered.

Finally, it is recommended that it is analyzed how much the performance can be further enhanced by adding a more optimal feed forward action. One possibility to do this is by implementing Iterative Learning Control.

Bibliography

[1] E. Canuto P. Martella, G. Sechi. The goce drag free and attitude control design aspects and expected performance. Alenia Spazio, Turin Plant, Italy.

[2] Walter Fichter. Drag free and attitude control engineering file. Technical report, Astrium, September 2002.

[3] *Spacecraft Systems Engineering.* Chapter 5.3: Polar Leo/Remote Sensing Satellites.

[4] J.V. Carrol. The notation and use of quaternions for shuttle ascent steering. Technical report, Astrium, September 2002.

[5] Nathan van de Wouw. *Multibody Dynamics.* 2003. Lecture Notes.

[6] James R. Wertz. *Spacecraft attitude determination and controll.* D.Reindel Publishing Company, 1978.

[7] M. Palomba. Goce models databank. Technical report, Alenia Spazio, Turin Plant, Italy, December 2002.

[8] Technical staff. *Applied Optimal Estimation.* The Analytic Science Corporation, 1999.

[9] prof.dr. H. Nijmeijer. *Nonlinear Dynamics.* 2002. Lecture Notes.

[10] M. Steinbuch. *Digital Motion Control.* 2003. Lecture Notes.

[11] Shankar Sastry. *Nonlinear Systems: Analysis, Stability and Control,* volume 10. 1999.

[12] Jan J. Kok. *Control of Multivariable Systems.* 2001/2002. Lecture Notes.

[13] Rob Tousain, Eduard van der Meche, and Okko Bosgra. Design strategies for iterative learning control based on optimal control. Technical report, Delft Univeristy of Technology, 2001.

Appendix A

LQR filter in *Matlab*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation file computing LQR controller: lqr_c.m %
%                                                                                 %
% - Willem-Jan Evers 2004                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear all
close all
error=0;
load('PolesPlace_Ctrl_3.mat');  % Loading old controller, needed for getting estim

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parameters                                                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Orbit definition
a0              = (6378+270)*1.e+03;            % Semi-major axis [m]
mu_gravity      = 3.986e+14;                    % [m3/s2]
w0              = 1/sqrt(a0^3/mu_gravity);      % Orbit frequency

% Time
% P-matrix
dt      = 10;                                    % time spacing P and L
tend    = 10*(2*pi/w0+dt);                       % final time time interval control
time    =[0:dt:tend];

% Magnetic field parameters
```

```
load('data/MagField.mat');
bmag    = Bmag.signals.values(:,2:4);          % Sampled at 1 sec!!
bt      = Bmag.signals.values(:,1);            % time sample data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GOCE System Definition: Continuous
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
J = [128    0       0
    0      2067    0
    0      0       2041];
Inertie=J;

Jinv    = inv(J);

sig     = [ (J(2,2) - J(3,3))/J(1,1)
    (J(3,3) - J(1,1))/J(2,2)
    (J(1,1) - J(2,2))/J(3,3)];

A12     = eye(3,3);

A21     = w0^2*diag([-4*sig(1),3*sig(2),sig(3)]);

A22     = [ 0,              0,          w0*(1-sig(1))
            0,              0,          0
            -w0*(1+sig(3))  0,          0];

A    = [zeros(3)   A12,
        A21,       A22];

B    = [zeros(3)
        Jinv];

C    = eye(6);

D    = zeros(6,3);

% Exits you want controlled:
Cz    = [ eye(3),    zeros(3)
        zeros(3),  eye(3)
        A21,       A22];

Dz    = [zeros(6,3);Jinv];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LQR definitions                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% J = int[x.'*Q*x+u.'*Rs*u]dt+x.'*Pe*x

% Weight matrices (chosen constant):
W11 = [1/(0.001)^2 0 0
       0 1/(0.005)^2 0
       0 0 1/(0.005)^2];
W22 = [1/(5e-6)^2 0 0
     0 1/(1e-5)^2 0
     0 0 1/(5e-5)^2];
W33 = [1/(5e-4)^2 0 0
     0 1/(3e-4)^2 0
     0 0 1/(1e-3)^2];
W = [W11     zeros(3)    zeros(3)
     zeros(3)    W22    zeros(3)
     zeros(3)    zeros(3)    W33];
R = [1/(10)^2 0 0;
     0 1 0;
     0 0 1/(2)^2];


% resulting matrices
Rinv=inv(R); Ee  =   Cz-Dz*Bm*KcPD3;

Pe= Ee.'*W*Ee;                    % Vary to check validity of optimum!

% Initialisation
P=[Pe];
progress=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% discreet loop: solving P using backstepping -> P will be computed
% backward so postproccessing of P matrix will be required.
for i=1:length(time)
    if error == 0
        ta=time(length(time)-(i-1)); % actual time
        % New B computation with real Bmag
```

```
        ii = min(find(bt>=ta));
        Bm   = [0              bmag(ii,3)  -bmag(ii,2);
                -bmag(ii,3)    0           bmag(ii,1);
                bmag(ii,2)     -bmag(ii,1)    0];
        Bt=B*Bm;

        % Optimal controller
        L(:,:,i)=inv(R)*Bt.'*P(:,:,i);

        % Matrices computation
        E = Cz-Dz*Bm*L(:,:,i);
        Q = E.'*W*E;

        % Solving Riccati equation:
        dP          = -(A.'*P(:,:,i)+P(:,:,i)*A+Q-P(:,:,i)*Bt*inv(R)*Bt.'*P(:,:,i));
        P(:,:,i+1)  = P(:,:,i)-dt*dP;

        % Progress printing
        if i/length(time)*100>progress
            clc
            disp('Computing P(t):');
            progress=progress+1
        end
        if abs(trace(L(:,:,i)))>1e100
            disp('Optimization proces diverging!');
            i=length(time);
            error=1;
        end
    end
end
if error == 0
        % Spacing P and L chronologically
        i=0;progress=0;
        Pb=P;P=[];Lb=L;L=[];
        for i=1:length(time)
            P(:,:,i)=Pb(:,:,length(Pb)-(i-1));
            L(:,:,i)=Lb(:,:,length(Lb)-(i-1));
            % Progress printing
            if i/length(time)*100>progress
                clc
                disp('Spacing P(t) and L(t) chronologically:');
                progress=progress+1
```

```
            end
        end
    end
```

Appendix B

LQR *Simulink* implemention

```
function [sys,x0,str,ts] = LQR_comp(t,x,u,flag,dt,dt_c,L,Bmag)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation LQR implementation S-function: lqr_comp_long_ff.m %
%                                                                       %
% - Willem-Jan Evers 2004                                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage.
switch flag,

   case 0
     [sys,x0,str,ts] = mdlInitializeSizes(dt); % Initialization

   case 3
     sys = mdlOutputs(t,x,u,flag,dt_c,L,Bmag); % Calculate outputs

   case { 1, 2, 4, 9 }
     sys = []; % Unused flags

   otherwise
     error(['Unhandled flag = ',num2str(flag)]);% Error handling
end;
% End of function timestwo.

%=============================================================
% Function mdlInitializeSizes initializes the states, sample
```

```
% times, state ordering strings (str), and sizes structure.
%=============================================================
function [sys,x0,str,ts] = mdlInitializeSizes(dt)

% Call function simsizes to create the sizes structure.
sizes = simsizes;

% Load the sizes structure with the initialization information.
sizes.NumContStates= 0;
sizes.NumDiscStates= 0;
sizes.NumOutputs=    6;
sizes.NumInputs=     18;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;

% Load the sys vector with the sizes information.
sys = simsizes(sizes);
x0  = [];           % No continuous states
str = [];           % No state ordering
ts  = [dt 0];       % controller sample time
% End of mdlInitializeSizes.

%=============================================================
% Function mdlOutputs performs the calculations.
%=============================================================
function sys = mdlOutputs(t,x,u,flag,dt_c,L,Bmag)
                        % inputs: bmag, theta, w, wdot, Tdist, dTdist
bmag        = u(1:3);   % NB. Bmag is magnetic field data used for building L
theta(1:3,1)= u(4:6);
w(1:3,1)    = u(7:9);
wdot        = u(10:12);
Tdist       = u(13:15);
dTdist      = u(16:18);

xact        = [theta; w];

% ----------------------------------
% Computing commanded magnetic moments
% ----------------------------------

% Orbit definition
a0          = (6378+270)*1.e+03;    % Semi-major axis [m]
```

```
mu_gravity  = 3.986e+14;            % [m3/s2]
w0          = 1/sqrt(a0^3/mu_gravity);

% Finding right controller i for this time
i   = floor(t/dt_c)+1; Li  = L(:,:,i);

% computing feedback control magnetic moment Mcmd(t)
M   = -Li*xact;

% Used actual feedback torque estimate: T = M x Bmag = Bm*M
Bm = -[ 0           -bmag(3)    bmag(2)
        bmag(3)     0           -bmag(1)
        -bmag(2)    bmag(1)     0];
T  = Bm*M;

% Feedforward
%1. Tcmd_ff = -Tdist
b1 = bmag(1); b2 = bmag(2); b3 = bmag(3); bb = b1^2+b2^2+b3^2;
Bms= 1/bb*[(b2^2+b3^2)       -(b1*b2)     -(b1*b3)
           -(b1*b2)           (b1^2+b3^3)  -(b2*b3)
           -(b1*b3)           -(b2*b3)      (b1^2+b2^2)];
Bmstripe= -1/bb*Bm;

Mff=-Bmstripe*Tdist; Tff=-Bms*Tdist;

% Total control output
T = T+Tff; M = M+Mff;

sys =[T;M];                 % outputs: Tact_estimate, Mcmd

% End of mdlOutputs.
```

Appendix C

Floquet Multipliers in *Matlab*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation file computing FM: floquet_total.m %
%                                                %
% - Willem-Jan Evers 2004                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc
close all
clear all
format long e

load('data/MagField.mat');
bmag=Bmag.signals.values(:,2:4);        % magnetic field sampled at 1 hz
load lqr_long.mat                       % LQR
%load('PolesPlace_Ctrl_3.mat')          % PD controller

% Processing parameters
dt          = 1;                        % sampletime fm computation

% Parameters
a0          = (6378+270)*1.e+03;        % Semi-major axis [m]
mu_gravity  = 3.986e+14;                % [m3/s2]
w0          = 1/sqrt(a0^3/mu_gravity);

% Attitude Dynamics
%%%%%%%%%%%%%%%%%%%

nstate  = 6; ninputs = 3;
```

```
% Inertia matrix [kg*m2]

Inertie = [ 128         0.*(-25)    0.*(-2.9)
            0.*(-25)    2067        0.*(1.41)
            0.*(-2.9)   0.*(1.41)   2041];

Iinv    = inv(Inertie); Jinv    = Iinv;

sig     = [ (Inertie(2,2) - Inertie(3,3))/Inertie(1,1);...
            (Inertie(3,3) - Inertie(1,1))/Inertie(2,2);...
            (Inertie(1,1) - Inertie(2,2))/Inertie(3,3)];

A21     = w0^2*diag([-4*sig(1),3*sig(2),sig(3)]);

A22     = [ 0,              0,          w0*(1-sig(1));...
            0,              0,          0;...
            -w0*(1+sig(3))  0,          0];

Ac   = [ zeros(3)   eye(3),
         A21,       A22];

Bc   = [ zeros(3)
         Jinv];

Cc    = eye(6); Dc    = zeros(6,3);

clear A11 A12 A21 A22 sig nstate ninputs

% Time interval
t0  = 0; tend= floor(6*pi/w0); t   = [t0:dt:tend].';

% Computing all Floquet Multipliers
i = 0;m=0; Sold=eye(size(Ac));
istates=length(Ac);
fm=zeros(istates,length(t));            % Filling the matrix for faster computation

for i = 1:length(t)
    % Controller
    Li=[L(:,:,floor(i*dt/10)+1)];       % LQR
    %Li = KcPD3;                        % PD
```

```
    % Magfield data in bmag
    b1 = bmag(i,1);
    b2 = bmag(i,2);
    b3 = bmag(i,3);
    bb = b1^2+b2^2+b3^2;

    % Actuation-matrix for u = Tact and Tact = Bm*Mcmd = -Bm*L*x -> LQR
    Bm = [ 0            b3     -b2
        -b3          0       b1
        b2          -b1      0];

%    % Actuation-matrix for ycntr=Tcmd and udyn=Tact                -> PD
%    Bm = 1/bb*[ (b2^2+b3^2)       -(b1*b2)     -(b1*b3)
%        -(b1*b2)          (b1^2+b3^3)  -(b2*b3)
%        -(b1*b3)          -b2*b3       (b1^2+b2^2)];

        Acl=Ac-Bc*Bm*Li;
        Ccl=Cc;
        Bcl=zeros(6,3);
        Dcl=zeros(6,3);

        % Discretisation
        SYS=pck(Acl,Bcl,Ccl,Dcl);
        DSYS=tustin(SYS,dt);
        [A,B,C,D]=unpck(DSYS);       % Discrete A,B,C,D

        %Computation of S-matrix:
        S   = A*Sold;
        fm(:,i) = eig(S);
        Sold = S;

        % proces indicator
        if (i)/(length(t))*100>m
            clc
            m=m+1
        end
    end

    fm(:,end)


    % Graphical representation
```

```
% number of subplots vertical
n=ceil(sqrt(length(A)));

% unit circle
x=[-1:0.01:1];
y1=sqrt(1-x.^2);
y2=-y1;
% plot real(fm),im(fm) change in time
figure
title('Floquet multipliers in complex plane')
for j=1:istates
    subplot(n,istates/n,j)
    plot(real(fm(j,:)),imag(fm(j,:)),real(fm(j,end)),imag(fm(j,end)),'ro',...
    x,y1,'k:',x,y2,'k:')
    xlabel('Real(fm)')
    ylabel('Imag(fm)')
    grid
end
% plot real(fm) vs time
figure
plot(t,real(fm))
ylabel('Real(fm) [-]')
xlabel('Time [s]')
title('Floquet multipliers vs time')
grid
```