# Orbital Mechanics and Feedback Control

*A thesis submitted for the degree of Master of Science*

by

Kristin Johansson
Trondheim, June 15, 2005

Norwegian University of Science and Technology
Department of Engineering Cybernetics

**NTNU**
**Norges teknisk-naturvitenskapelige universitet**

**Fakultet for informasjonsteknologi, matematikk og elektroteknikk**
**Institutt for teknisk kybernetikk**

# MASTEROPPGAVE

Kandidatens navn: Kristin Johansson

Fag: Teknisk Kybernetikk

Oppgavens tittel (norsk): Regulering av banemekanikk.

Oppgavens tittel (engelsk): *Orbital Mechanics and feedback control.*

Oppgavens tekst:

ESMO er en mikrosatelitt under ESAs SSETI-prosjekt. Satelitten er tenkt å gå inn i bane rundt månen. Følgende skal utføres

- Gi en overikt over begreper og geometri for jord-baner og måne-baner og presenter trajektorer og tilhørende banemanøvre for å bringe ESMO i bane rundt månen
- Presenter 2-, 3-, og N-body problemene og studer numeriske integratorer som er egnet for å simulere disse.
- Presenter en dynamisk modell som er egnet for regulering av banene. Simulér en av de foreslåtte banene.
- Undersøk om bruk av tilbakekobling og PID-regulering kan forbedre banen til satellitten under påvirkning av pertubasjoner.
- Foreslå selv en regulator og undersøk stabilitetsegenskapene til denne.

Oppgaven gitt: 20/1-05

Besvarelsen leveres: 16/6-05

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Trondheim, den 20/1-05

Jan Tommy Gravdahl

Jan Tommy Gravdahl
Faglærer

# Preface

This report is the result of the work on the master's thesis 'Orbital Mechanics and Feedback Control' on the micro satellite The European Student Moon Orbiter (ESMO). The work was carried out at the Norwegian University of Science and Technology under the Department of Engineering Cybernetics in cooperation with the Student Space Exploration Technology Initiative (SSETI), a project supported by the Education Office of the European Space Agency (ESA). I want to thank my supervisor, Associate Professor Jan Tommy Gravdahl, for support and motivation throughout the project. I also want to thank my fellow students in the ESMO project for a good working environment and fellow student Hege Sande for proofreading.

Trondheim, June 15, 2005

Kristin Johansson

# Contents

# Abstract

In this master's thesis, trajectories and orbit control for the micro satellite ESMO (The European Student Moon Orbiter) are studied. The ESMO satellite is a project of the Student Space Exploration Technology Initiative (SSETI) that works together with the European Space Agency (ESA).

For better understanding, an overview of concepts and geometry for Earth and Moon orbits is given. Then three transfer methods are presented; the Hohmann transfer, the Ballistic Capture Transfer (BCT) and the Patched Conic Approximation (PCA). The latter method is studied closer in this thesis.

Orbit control is studied, and two controllers are suggested as a means of keeping the satellite in orbit despite perturbations. The first is the traditional PID controller and the second a nonlinear controller derived from Lyapunov control theory.

The MatLab/Simulink environment is chosen for simulations. To make it perform its best, different solvers are tested. The PCA trajectory is simulated and used as the reference trajectory. The Moon is added as a perturbation. The two mentioned controllers are simulated on this system to make the satellite follow the reference trajectory even with the perturbation. Stability is studied for the nonlinear controller. The corresponding Lyapunov function is also simulated.

# Chapter 1

# Introduction

In SSETI students from many different European countries participate. They are all united by their desire to 'launch the dream'. ESA's Education Office gives them full support. There are four planned launches. The SSETI project started with the vision to create and build a micro-satellite and should be completed with the development of a Moon Rover in the third mission. The launch described in this thesis is the micro satellite The European Student Moon Orbiter, (ESMO). The goal of this launch is to make the ESMO satellite orbit the Moon.

One of the teams working on the ESMO project is the Attitude Determination and Control System (ADCS) team. They work on control of the attitude and the orbit of the ESMO satellite.

To add control to the orbit of the satellite, forces acting on the satellite need to be described. Here there are many options on what to include. The number of celestial bodies has to be decided on, making the problem a two-, three- or four-body problem depending on the number of celestial bodies included. These will form the largest forces, but other forces such as atmospheric drag and solar radiation pressure can also be included. Also, there are many possibilities to choose from amongst possible trajectories to get to the Moon. Some are more fuel-efficient than others, but these often use longer time. But no matter which is used, there will always be perturbations to consider.

To deal with the perturbations, there are two main options; to try to model most of them and include them in the model describing the environment of the satellite, or to just take into account the most fundamental forces in the environment and add a controller to make the satellite keep its desired trajectory. The latter is chosen for this thesis to get large errors in the trajectory to give a practilcal environment to test orbit controllers.

# Chapter 2

# Orbital Mechanics

The main part of this chapter is taken from (Johansson [2004]). It gives an overview of concepts and geometry for orbital mechanics.

## 2.1 Dynamics

Satellite orbits are results of basic nature laws, such as Newton's and Kepler's laws.

### 2.1.1 Newton's laws

Newton's three laws of motion are given as (Sidi [2000]):

1. *Every object in a state of uniform motion tends to remain in that state of motion unless an external force is applied to it.*
2. *The relationship between an object's mass $m$, its acceleration $a$, and the applied force $\vec{F}$ is*

$$\vec{F} = m\vec{a}. \tag{2.1}$$

3. *For every action there is an equal and opposite reaction,*

$$\vec{F}_{ab} = -\vec{F}_{ba}. \tag{2.2}$$

There is also the law of gravitational attraction;

*Any two objects with masses $m_1$ and $m_2$ exert a gravitational force $\vec{F}$ of attraction on each other given by*

$$\vec{F} = \frac{Gm_1m_2\vec{r}}{r^3} \tag{2.3}$$

*where $G = 6.67259 \times 10^{-11} m^3 kg^{-1} s^{-2}$ is the gravitational constant and $\vec{r}$ is the vector with magnitude $r$ along the line connecting the two masses. The direction of the force is along this line joing the objects.*

### 2.1.2 Kepler's laws

Kepler gave three main laws of orbital mechanics (Sidi [2000]):

1. *All the planets orbit the Sun in an elliptic orbit with the Sun at one focus*
2. *For any planet orbiting the Sun the line joining them sweeps out equal areas in equal intervals of time*
3. *The square of the sideral period of an orbiting planet is directly proportional to the cube of the orbit's semimajor axis*

### 2.1.3 The general n-body problem

In a system consisting of $n$ bodies, the sum of forces acting on the $i$th body is (Sidi [2000])

$$\mathbf{F}_i = G\sum_{j=1}^{j=n} \frac{m_i m_j}{r_{ij}^3}(\mathbf{r}_j - \mathbf{r}_i),\ i \neq j. \tag{2.4}$$

It follows from Newton's second law of motion, equation (2.1), that

$$\frac{d^2\mathbf{r}_i}{dt^2} = G\sum_{j=1}^{j=n} \frac{m_i}{r_{ij}^3}(\mathbf{r}_j - \mathbf{r}_i),\ i \neq j. \tag{2.5}$$

### 2.1.4 The two-body problem

The simplest of the n-body problems is the two-body problem. Here, only two masses is considered at a time. Let the masses be denoted as $m_1$ and $m_2$. The n-body equation (2.4) becomes

$$\mathbf{F}_1 = m_1\ddot{\mathbf{r}}_1 = Gm_1m_2\frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|^3}, \tag{2.6}$$

$$\mathbf{F}_2 = m_2\ddot{\mathbf{r}}_2 = Gm_1m_2\frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} = -\mathbf{F}_1. \tag{2.7}$$

Combining these two equations gives (Sidi [2000])

$$\ddot{\mathbf{r}}_2 - \ddot{\mathbf{r}}_1 = -G(m_1 + m_2)\frac{\mathbf{r}_2 - \mathbf{r}_1}{r^3}, \tag{2.8}$$

and with $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$,

$$\ddot{\mathbf{r}} + G(m_1 + m_2)\frac{\mathbf{r}}{r^3} = 0. \tag{2.9}$$

Equation (2.9) is the basic equation of motion for the two-body problem.

### 2.1.5 The three-body problem

The restricted three-body problem is a good way to describe forces between Earth, the Moon and a satellite. It consists of a system that includes three masses moving in a plane. Let Earth have mass $m_1$, the Moon mass $m_2$ and the satellite mass $m_3$, (Egeland and Gravdahl [2002]). Mass $m_3$ is a lot smaller than $m_1$ and $m_2$, so it can be neglected.

The law of gravitation gives gravity force $\vec{F}_1$ on Earth from the Moon and gravity force $\vec{F}_2$ the opposite way. They are given by

$$\vec{F}_1 = -\vec{F}_2 = k^2\frac{m_1m_2}{L^2}\vec{b}_1 \tag{2.10}$$

where $k$ is the Gaussian constant of gravitation, $L$ is the distance between body 1 and 2 and $\vec{b}_1$ is the unit vector along the axis from Earth to the Moon. The vector $\vec{b}_3$ is along the axis of rotation of the Earth-Moon system.

The vector from the centre of Earth to the centre of the Moon rotates with an angular velocity $\vec{\omega} = \omega\vec{b}_3$. Earth has position $\vec{R}_1 = -x_1\vec{b}_1$ and the Moon has position $\vec{R}_2 = -x_2\vec{b}_1$. $L$ is therefore given by $L = x_1 + x_2$. The accelerations become (Egeland and Gravdahl [2002])

$$\vec{a}_1 = \vec{\omega} \times (\vec{\omega} \times \vec{R}_1) = \omega^2 x_1 \vec{b}_1 \tag{2.11}$$

$$\vec{a}_2 = \vec{\omega} \times (\vec{\omega} \times \vec{R}_2) = -\omega^2 x_2 \vec{b}_1. \tag{2.12}$$

The gravitational and centrifugal forces are in balance. This gives

$$k^2 \frac{m_1 m_2}{L^2} = m_1 x_1 \omega^2 = m_2 x_2 \omega^2, \tag{2.13}$$

and from this Kepler's third law is found as

$$\omega^2 = \frac{k^2 M}{L^3}, \tag{2.14}$$

where $M = m_1 + m_2$. The position of the satellite is

$$\vec{r} = x\vec{b}_1 + y\vec{b}_2, \tag{2.15}$$

the velocity is

$$\vec{v} = \frac{d}{dt}\vec{r} + \vec{\omega_{ib}} \times \vec{r} = \dot{x}\vec{b}_1 + \dot{y}\vec{b}_2 + \omega(x\vec{b}_2 - y\vec{b}_1) \tag{2.16}$$

and the acceleration becomes

$$\begin{aligned} \vec{a} &= \frac{d^2}{dt^2}\vec{r} + 2\vec{\omega_{ib}} \times \frac{d}{dt}\vec{r} + \vec{\alpha_{ib}} \times \vec{r} + \vec{\omega_{ib}} \times (\vec{\omega_{ib}} \times \vec{r}) \\ &= \ddot{x}\vec{b}_1 + \ddot{y}\vec{b}_2 + 2\omega(\dot{x}\vec{b}_2 - \dot{y}\vec{b}_1) - \omega^2(x\vec{b}_1 + y\vec{b}_2). \end{aligned} \tag{2.17}$$

Using equation(2.4), the motion of the satellite can be described as

$$\vec{F}_3 = -k^2 \frac{m_1 m_3}{r_1^3}\left[(x + x_1)\vec{b}_1 + y\vec{b}_2\right] - k^2 \frac{m_2 m_3}{r_2^3}\left[(x - x_2)\vec{b}_1 + y\vec{b}_2\right], \tag{2.18}$$

where

$$\vec{r}_1 = \sqrt{(x+x_1)^2 + y^2},\ \vec{r}_2 = \sqrt{(x-x_2)^2 + y^2}. \tag{2.19}$$

In the $x$ and $y$ direction this results in

$$\ddot{x} - 2\omega\dot{y} - \omega^2 x = -k^2\left[\frac{m_1}{r_1^3}((x+x_1) + \frac{m_2}{r_2^3}(x-x_2)\right] \tag{2.20}$$

$$\ddot{y} + 2\omega\dot{x} - \omega^2 y = -k^2\left(\frac{m_1}{r_1^3} + \frac{m_2}{r_2^3}\right). \tag{2.21}$$

The model is usually presented in normalized form where the distances are divided by $L$ and $\tau = \omega t$.

This is just one way of presenting the three-body problem. It can be done more easily by just using equation (2.5) for $n = 3$ bodies. It is however more time consuming to do computations on.

### 2.1.6 Dynamics of orbits

When orbital mechanics is to be described, there are many different types of coordinate systems to choose from. It is quite easily expressed in polar coordinates. The plane polar coordinates are $(r, \theta)$ and the unit vectors are $(\vec{e_r}, \vec{e_\theta})$ as shown in Figure 2.1. The velocity vector is

$$\vec{v} = \dot{r}\vec{e_r} + r\dot{\theta}\vec{e_\theta}, \tag{2.22}$$

and acceleration vector is

$$\vec{a} = (\ddot{r} - r\dot{\theta}^2)\vec{e_r} + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\vec{e_\theta}. \tag{2.23}$$

The equations of motion can be divided up into radial and transverse direction from equation (2.23).

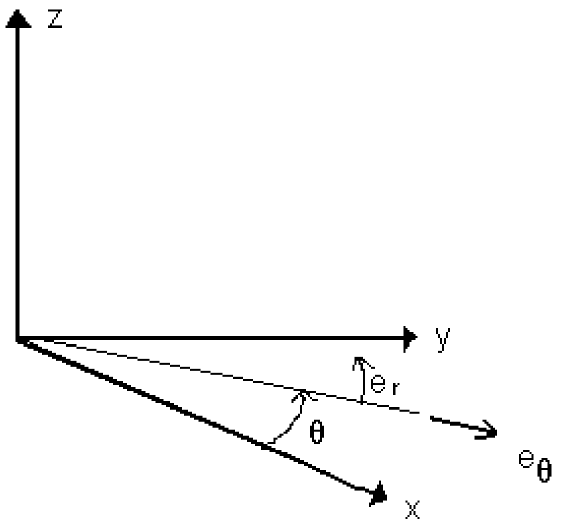



Figure 2.1: Polar coordinates

In the radial direction the equation of motion is

$$\ddot{r} - r\dot{\theta}^2 = -\frac{\mu}{r^2}, \tag{2.24}$$

where $\mu = Gm$ ($G$ - gravitational constant, $m$ - mass of spacecraft) and the whole expression on the right hand side is gravity. This is the only acceleration that works in radial direction.

In transverse direction the equation of motion is

$$r\ddot{\theta} + 2\dot{r}\dot{\theta} = 0, \tag{2.25}$$

which can be restated as

$$\frac{1}{r}\frac{d}{dt}(r^2\dot{\theta}) = 0. \tag{2.26}$$

From equation (2.26) it can be seen that $r^2\dot{\theta}$ is constant. $r^2\dot{\theta}$ is equal to the angular momentum per unit mass; $h$. By substiting $\dot{\theta} = \frac{h}{r^2}$ into equation

(2.24)

$$\ddot{r} - \frac{h^2}{r^3} = -\frac{\mu}{r^2} \tag{2.27}$$

is found. From equation (2.27), the following equation can be derived

$$r(\theta) = \frac{\frac{h^2}{\mu}}{1 + \frac{Ah^2}{\mu}\cos(\theta - \theta_0)}, \tag{2.28}$$

where $A$ and $\theta_0$ are constants. This is the equation of an ellipse in polar coordinates, (McInnes [2004]).

## 2.2 Geometry of orbits

The simplest orbits follow basic geometry of conic sections. Conic sections are different intersections of a plane and a cone. Much of the material in this section is from (Bate et al. [1971]) and (Sidi [2000]).

The circle intersects the cone horizontally, and the ellipse intersects the cone with a tilt, see Figure 2.2. Both are closed curves. The hyperbola intersects the cone resulting in an open curve. There is yet another basic conic section; the parabola. The parabola is the single curve which divides the closed ellipse from the open hyperbola. Here the plane is parallell to the side of the cone.

There are two points of particular interest on the orbits; the pericentre[1] and the apocentre[2]. The pericentre is the point where a spacecraft will be closest to the object it is orbiting, and the apocentre is the point furthest away, see Figure 2.3.

In Figure 2.3, $b$ is the semi-minor axis, $a$ is the semi-major axis and $ae$ is the distance from the center to the focal point. The distance from the center to the focal point is determined by the conic sections's eccentricity; $e$. The eccentricity determines the type of orbit obtained. Table 2.1 gives the orbits with the corresponding eccentricities.

[1] also referred to as perigee or periapsis
[2] also referred to as apogee or apoapsis

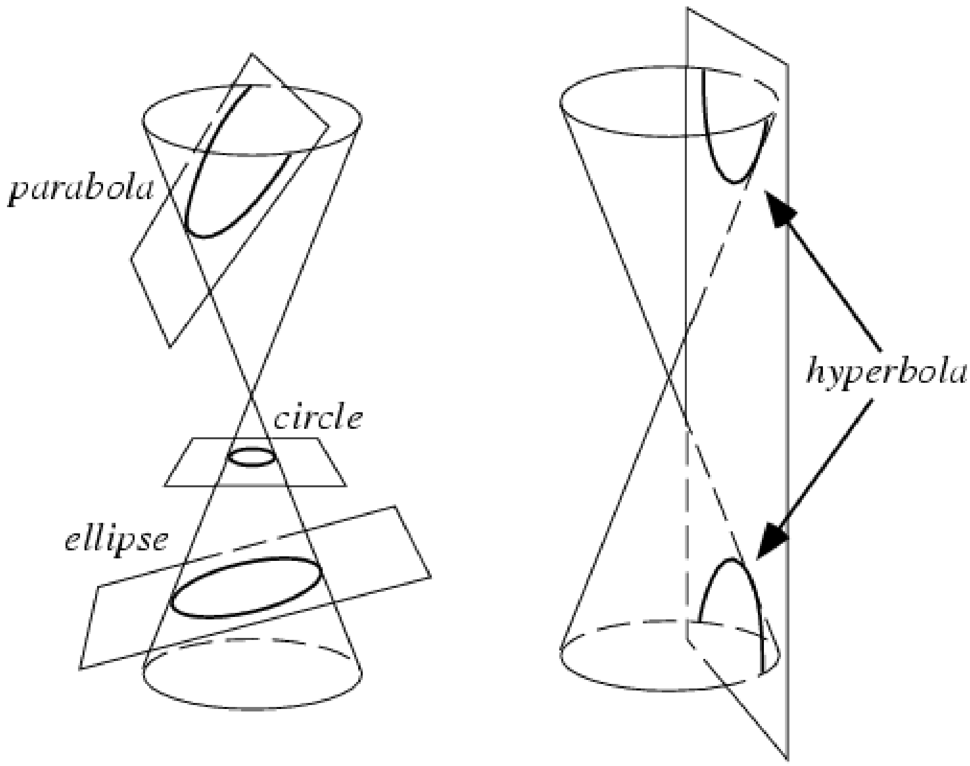



Figure 2.2: Conic sections

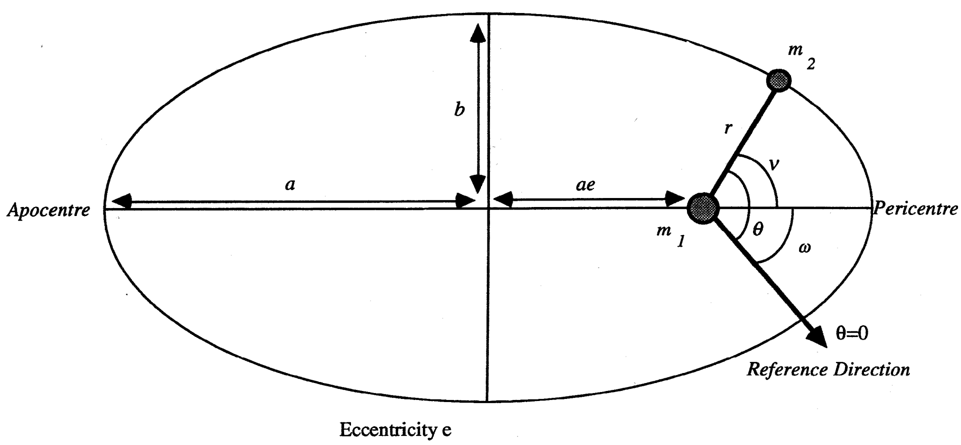



Figure 2.3: Orbit parameters

| *Eccentricity* | *Orbit* |
|---|---|
| $e = 0$ | Circular Orbit |
| $0 < e < 1$ | Elliptical Orbit |
| $e = 1$ | Parabolic Orbit |
| $e > 1$ | Hyperbolic Orbit |

Table 2.1: Orbits with corresponding eccentricities

To describe an orbit accurately, more parameters are needed. The six required elements to fully define an orbit are described in Table 2.2, see also Figure 2.3 and 2.4.

| *Element* | *Name* | *Description* |
|---|---|---|
| $a$ | Semi-major axis | See Figure 2.3 |
| $e$ | Eccentricity | When mulitplied with a, it gives the distance from the centre of the orbit to the focal point |
| $i$ | Inclination | The angle between the equator and the orbit |
| $\Omega$ | Longitude of ascending node | The point where the satellite crosses equator moving south to north |
| $\omega$ | Argument of pericentre | Describes the orientation of the orbit |
| $\nu$ | True anomaly | Location of the satellite with respect to perigee |

Table 2.2: Description of Keplerian Elements

These parameters are called Keplerian elements. They are also referred to as classical orbital elements (COE).

Another parameter that is often used is the mean anomaly $M$. It can be used instead of the true anomaly $\nu$. $M$ is defined by

$$M = \epsilon - e \sin \epsilon, \tag{2.29}$$

where $\epsilon$ is the eccentric anomaly and is given by

$$\cos \epsilon = \frac{e + \cos \nu}{1 + e \cos \nu}. \tag{2.30}$$

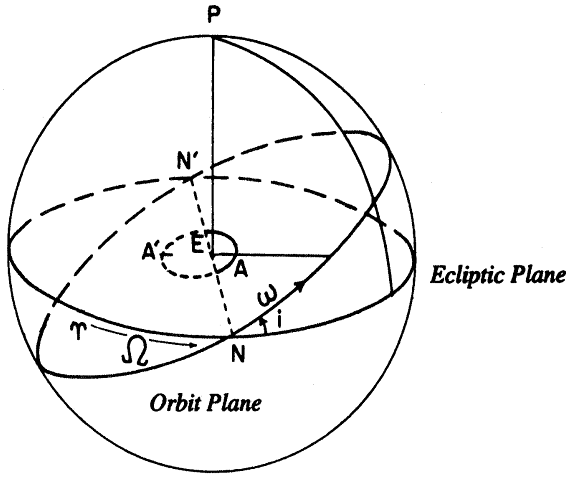



Figure 2.4: Keplerian elements

To convert a satellite position given in COEs to usual cartesian coordiates, the convertion(Hegrenæs [2004])

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = r \times \begin{pmatrix} \cos(\nu+\omega)\cos\Omega - \sin(\nu+\omega)\sin\Omega\cos i \\ \cos(\nu+\omega)\sin\Omega + \sin(\nu+\omega)\cos\Omega\cos i \\ \sin(\nu+\omega)\cos i \end{pmatrix} \tag{2.31}$$

is used, where $r$ is calculated as

$$r = \frac{a(1+e^2)}{1+e\cos\nu}. \tag{2.32}$$

The velocity of the satellite can be calculated as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \frac{na}{r} \begin{pmatrix} bl_2\cos\epsilon - al_1\sin\epsilon \\ bm_2\cos\epsilon - am_1\sin\epsilon \\ bn_2\cos\epsilon - an_1\sin\epsilon \end{pmatrix}, \tag{2.33}$$

where $n$ is the mean motion which is found from

$$n = \sqrt{\frac{\mu}{a^3}}, \tag{2.34}$$

and

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

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

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

$n_1 = \sin\omega\sin i$

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

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

$n_2 = \cos\omega\sin i.$

The time a satellite uses to get to a point on the orbit from the pericentre, is called time of flight. It can be calculated using Kepler equation

$$t_e = \frac{\epsilon - e\sin\epsilon}{n}. \tag{2.35}$$

### 2.2.1 Elliptical Orbits

The polar equation of an ellipse is found to be (Sidi [2000])

$$r(\nu) = \frac{a(1 - e^2)}{1 + e\cos\nu}, \tag{2.36}$$

where $r$ is the radius from the centre of the object that is orbited to the spacecraft, $a$ is the semi-major axis, $e$ is the eccentricity and $\nu$ is the true anomally. Equation (2.36) is simplified at the pericentre and at the apocentre. At the pericentre $\nu = 0$ and

$$r_p = a(1 - e). \tag{2.37}$$

At the apocentre $\nu = \pi$ and

$$r_a = a(1 + e). \tag{2.38}$$

The orbit is found by identifying

$$h^2 = \mu a(1 - e^2) \tag{2.39}$$

from equation (2.36) and (2.28). At the pericentre, angular momentum per unit mass, $h$, is $r_p v_p$. As $h$ is conserved, equation (2.39) becomes $a^2(1 - e)^2 v_p^2 = \mu a(1 - e^2)$. This results in the equation for pericentre speed

$$v_p = \sqrt{\frac{\mu}{a}\frac{1+e}{1-e}}. \tag{2.40}$$

Similarly, speed in the apocentre is found to be

$$v_a = \sqrt{\frac{\mu}{a}\frac{1-e}{1+e}}. \tag{2.41}$$

Total energy is expressed as $E = K + P$ where $K$ is kinetic energy and $P$ is potensial energy. The energy per unit mass is

$$E = \frac{1}{2}v^2 - \frac{\mu}{r}. \tag{2.42}$$

At the pericentre

$$E = \frac{1}{2}(\frac{\mu}{a}\frac{1+e}{1-e}) - \frac{\mu}{a(1-e)}, \tag{2.43}$$

which can be restated as

$$E = -\frac{\mu}{2a}. \tag{2.44}$$

Comparing equation (2.42) and (2.44) gives

$$\frac{1}{2}v^2 - \frac{\mu}{r} = -\frac{\mu}{2a}, \tag{2.45}$$

which can be rearranged to the equation of velocity on an elliptical orbit

$$v^2 = \mu(\frac{2}{r} - \frac{1}{a}). \tag{2.46}$$

The orbit period can be calculated from the equation for the area of an ellipse, the definition of an orbit period and equation (2.39), and results in

$$T = 2\pi\sqrt{\frac{a^3}{\mu}}. \tag{2.47}$$

### 2.2.2 Circular, Parabolic and Hyperbolic Orbits

In circular orbit, the eccentricity is zero, which means that the radius is constant; $R$. This results in the following velocity and orbit period equations (Sidi [2000])

$$v = \sqrt{\frac{\mu}{R}} \tag{2.48}$$

and

$$T = 2\pi\sqrt{\frac{R^3}{\mu}}. \tag{2.49}$$

In a parabolic orbit the eccentricity is one. This results in the velocity equation

$$v = \sqrt{\frac{2\mu}{r}}. \tag{2.50}$$

The orbit period $T \rightarrow \infty$ since $a \rightarrow \infty$.

In hyperbolic orbit the eccentricity is greater than one. The velocity equation is then

$$v^2 = \frac{2\mu}{r} + V_\infty^2, \tag{2.51}$$

where $V_\infty$ is the hyperbolic excess speed expressed as

$$V_\infty = \sqrt{\frac{\mu}{a}}. \tag{2.52}$$

## 2.3 Perturbations of Orbits

### 2.3.1 Lunar and Solar gravity

The Sun and the Moon can influence the trajectory of a satellite by their gravity. When one of them is taken into account with the satellite and Earth model, we must study a three-body problem. The following equation determines forces that are acting on the $i$th body in a system of $n$ bodies (Sidi [2000])

$$\vec{F}_i = G \sum_{j=1}^{j=n} \frac{m_i m_j}{r_{ij}^3} (\vec{r}_j - \vec{r}_i). \tag{2.53}$$

It follows that the acceleration becomes

$$\frac{d^2 r_i}{dt^2} = G \sum_{j=1}^{j=n} \frac{m_j}{r_{ij}^3} (\vec{r}_j - \vec{r}_i). \tag{2.54}$$

The mass of Earth, $m_e$, and the mass of the satellite, $m_s$, can be extracted from the summation. The used radii are definied as

$\vec{r} = \vec{r}_2 - \vec{r}_1 = \vec{r}_{12}$

$\vec{r}_{2j} = \vec{\rho}_j$

$\vec{r}_{1j} = \vec{r}_{pj}$ .

Placing the origin of the inertial frame at the center of Earth gives

$$\frac{d^2 r_i}{dt^2} = \sum_{j=3}^{j=n} \mu_{pj} \left[ \frac{\vec{\rho}_j}{\rho_j^3} - \frac{\vec{r}_{pj}}{r_{pj}^3} \right] \quad , \tag{2.55}$$

where $\mu_{pj} = Gm_{pj}$. Perturbations coming from these forces are very relevant for the ESMO satellite as it is travelling to the Moon. Naturally, as the satellite gets closer to the Moon, the more the gravity of the Moon will influence the orbit of the satellite.

### 2.3.2 The flattening of the Earth

Earth is in everyday life thought of as being a perfect sphere. But this is not entirely true. Earth is slightly flattened at top and bottom.

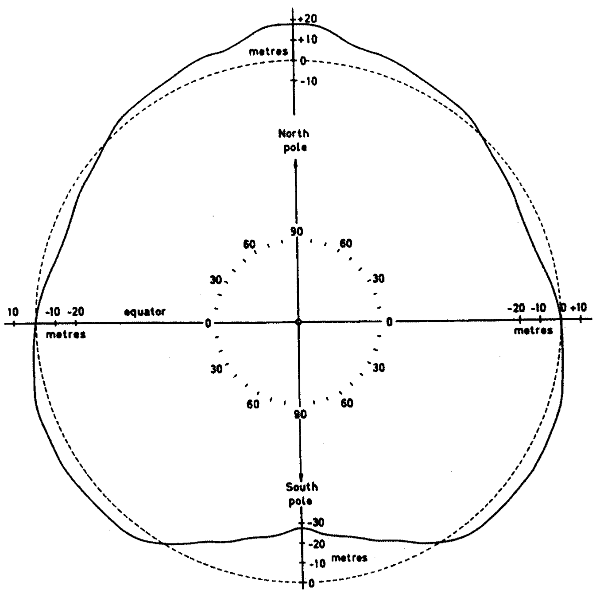



Figure 2.5: Earth's nonhomogenity

Besides being flat at top and bottom, Earth has a bulge on Equator. It is not important to take this effect into account for Low Earth Orbits (LEOs) as it will avarage out after many revolutions, but it should be taken into account when determining orbits for Geosynchronous Earth Orbits (GEOs). As the ESMO satellite will keep a high altitude orbit around Earth before being launched into Moon orbit, it is relevant.

The equations for this perturbation are found in (Sidi [2000]). When only $J_2$ zonal harmonic coefficient is considered, see (Sidi [2000]), the result is

$$\vec{a}_{Ex} = GA_{J2}(15\frac{xz^2}{r^7} - 3\frac{x}{r^5}) \tag{2.56}$$

$$\vec{a}_{Ey} = GA_{J2}(15\frac{yz^2}{r^7} - 3\frac{y}{r^5}) \tag{2.57}$$

$$\vec{a}_{Ez} = GA_{J2}(15\frac{z^3}{r^7} - 3\frac{z}{r^5}), \tag{2.58}$$

where $A_{J2} = \frac{1}{2}J_2R_e^2$ and $R_e$ is the mean equatorial radius of Earth and $J_2$ the zonal harmonic coefficient of order 0. The radius from the satellite to Earth is $r = \sqrt{x^2 + y^2 + z^2}$.

### 2.3.3 Atmospheric drag

Atomospheric drag is a breaking force and therefore dissipates energy from the satellite in orbit. Orbital height of the satellite will decrease slightly (Sidi [2000]). Atmospheric drag force is dependent on air density. Air density decreases with altitude. Therefore, atmospheric drag force is inversely proportional with altitude. It can be one of the main perturbations if working on a satellite in LEO. The ESMO satellite will only be in an orbit where atmospheric drag is relevant when it is in its parking orbit around Earth and possibly in the start of the transfer orbit.

Acceleration from atmospheric drag force can be expressed as (Rao et al. [2002])

$$\vec{a}_d = \frac{1}{2}\frac{\rho C_d A v^2}{m_{Sat}}, \tag{2.59}$$

where $\rho$ is the density of the air, $C_d$ is the drag coefficient, $A$ is the reference area and $v$ is the velocity of the satellite relative to the air. $A$ is often set to $\frac{1}{2}$.

### 2.3.4 Solar radiation pressure

The Sun's radiation causes a small force on the spacecraft that is exposed to it. This is because the Sun emits photons that are either absorbed or reflected by the satellite. Therefore, the force experienced by the satellite depends upon the surface area of the satellite. The acceleration can be expressed as (Rao et al. [2002])

$$\vec{a}_{sp} = P_s \nu r_{se}^2 C_r \frac{A}{m_{Sat}} \frac{\vec{r_{ss}}}{r_{ss}^3}, \tag{2.60}$$

where $\nu$ is the eclipse factor, $r_{se}$ is the distance between the Sun and Earth, $C_r = 1 + \eta$ where $\eta$ is the reflectivity of the surface of the satellite, $P_s$ is the solar radiation pressure constant and $A$ is the cross-section area. To make the equation easier to read, $\nu C_r r_{se}^2 \approx K$ .

### 2.3.5 Onboard thruster system

Not only celestial bodies can make perturbations to the satellite's trajectory. Also the satellite's own thrusters can change the orbit unintentionally. During a long thrust orbital manoeuvre, the mass of the satellite will change during the burn as propellant is consumed. A simple, constant thrust model is however often sufficient to describe the motion of a spacecraft during thrust arcs (Montenbruck [2000]). When a propulsion system ejects a mass $|dm| = |\dot{m}|dt$ per time interval $dt$ at a velocity $v_e$, the spacecratft of mass $m$ experiences a thrust $F$ which results in the acceleration

$$\vec{a}_t = \frac{F}{m} = \frac{|\dot{m}|}{m} v_e. \tag{2.61}$$

### 2.3.6 Bad thruster impulses

The satellite can cause itself to miss the desired trajectory if one fireing of the thrusters is not properly performed. This can be seen as a kind of perturbation as it makes the satellite follow another trajectory than the desired one.

### 2.3.7 Impact on the satellite motion equation

Recall equation (2.3), where gravitational force is expressed. By dividing by the mass of the satellite, the acceleration of the satellite is found and expressed as

$$\vec{a} = \frac{GM\vec{r}}{r^3}. \tag{2.62}$$

With perturbations, the expression becomes more complex (Rao et al. [2002])

$$\vec{a} = \frac{GM\vec{r}}{r^3} + \vec{k}_s, \tag{2.63}$$

where

$$\vec{k}_s = \vec{a}_m + \vec{a}_s + \vec{a}_E + \vec{a}_d + \vec{a}_{sp} + \vec{a}_t, \tag{2.64}$$

and $\vec{a}_m$ and $\vec{a}_s$ are due to Lunar and Solar gravity, $\vec{a}_E$ is due to the flattening of Earth, $\vec{a}_d$ is due to atmospheric drag, $\vec{a}_{sp}$ is due to solar radiation pressure and $\vec{a}_t$ is due to the thrusters.

# Chapter 3

# Trajectories and Orbital manoeuvres

There are many possible trajectories and orbital manoeuvres to choose between. The choice depends on the satellite that finally is deceided on. Different kinds of propulsion will make a big difference in weight of the satellite. The best trajectory should be carefully chosen in terms of applicability, reliability, simplicity and cost.

## 3.1 Orbital manoeuvres

To get the satellite to follow a trajectory, different orbital manoeuvres have to be performed. These are performed by consumption of propellant. The less propellant the mission needs to use, the lighter the satellite can be and the less the mission will cost. Therefore, propellant consumption is a crucial factor in orbital manoeuvres.

Adjustment of orbits can be made by single or multiple impulses. Only a few orbital manoeuvres can be obtained by a single impulse, but by using multiple impulses, any desired orbit can be obtained. Also, when making sure that the correct orbit is obtained, multiple impulses have to be used, (Sidi [2000]).

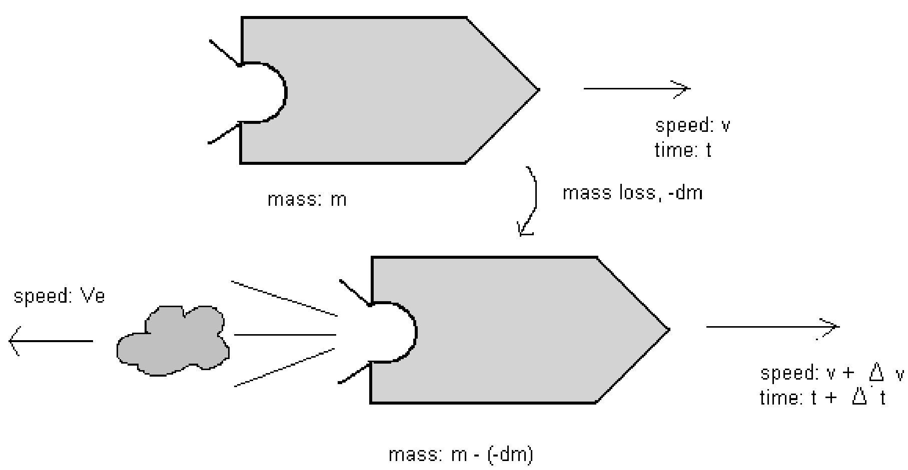



Figure 3.1: Spacecraft using propellant

### 3.1.1 The Rocket Equation

The rocket equation relates change in the velocity of the spacecraft, $\Delta V$ to the change in mass, that is propellant used. The initial condition of the spacecraft is mass $m$ and velocity $V$ at time $t$. Then the thrusters let out an exhaust gas element, $-dm$, that exits with exhaust speed $V_e$. The new velocity of the spacecraft is $V + \Delta V$ and the mass is $m - (-dm)$ at time $t + \Delta t$, see Figure 3.1. The linear momentum is conserved so that (McInnes [2004])

$$\begin{aligned} mV &= (m - (-dm))(V + dV) - (-dm)(V_e - V) \\ &= mV + mdv + Vdm + dmdV + V_e dm + Vdm, \end{aligned} \tag{3.1}$$

where $dmdV \approx 0$. If $V_e$ is constant, equation (3.1) can be written as

$$\int_{V_1}^{V_2} dV = -V_e \int_{m_1}^{m_2} \frac{dm}{m}, \tag{3.2}$$

$$(V_2 - V_1) = -V_e \ln(\frac{m_2}{m_1}). \tag{3.3}$$

Defining $\Delta V = (V_2 - V_1)$ gives

$$\Delta V = V_e \ln(\frac{m_1}{m_2}), \tag{3.4}$$

or equally

$$m_2 = m_1 e^{\frac{-\Delta V}{V_e}}. \tag{3.5}$$

Efficiency of a given thruster is measured in specific impulse, $I_{sp}$, which is defined as momentum gained per unit weight of propellant used,

$$I_{sp} = \frac{dmV_e}{gdm} = \frac{V_e}{g}, \tag{3.6}$$

where $g$ is the gravitational acceleration. Here, $I_{sp}$ is measured in seconds. Sometimes, $I_{sp}$ is measured in $m/s$. This results in the normal form of the rocket equation

$$m_2 = m_1 e^{\frac{-\Delta V}{gI_{sp}}}, \tag{3.7}$$

and when defining $\Delta m = m_1 - m_2$

$$\Delta m = m_1(1 - e^{\frac{-\Delta V}{gI_{sp}}}). \tag{3.8}$$

### 3.1.2 Getting from one orbit to another

To get from a circular orbit to an elliptical transfer orbit, which is often used in different manoeuvres, the change in velocity, $\Delta V$, required can be found by finding the velocity on a circle, equation (2.48), and subtract it from the velocity on an ellipse, equation (2.46). The resulting $\Delta V$ is

$$\Delta V = \sqrt{\frac{2\mu}{r_1} - \frac{2\mu}{r_1 + r_2}} - \sqrt{\frac{\mu}{r_1}}, \tag{3.9}$$

where $r_1$ is the radius of a circle and the pericentre of an ellipse, and $r_2$ is the radius of the apocentre of an ellipse.

To get from a circular to a parabolic orbit, the required $\Delta V$ is (McInnes [2004])

$$\Delta V = (\sqrt{2} - 1)\{\sqrt{\frac{\mu}{r_1}} + \sqrt{\frac{\mu}{r_2}}\}. \tag{3.10}$$

To get from a circular to a hyperbolic orbit, the required $\Delta V$ is

$$\Delta V = \sqrt{\frac{2\mu}{R} + V_\infty^2} - \sqrt{\frac{\mu}{R}}. \tag{3.11}$$

## 3.2 Trajectories

The trajectory of the ESMO satellite is not yet determined (June 13, 2005). Therefore, more than one option is considered in this section.

### 3.2.1 Hohmann Transfer

The Hohmann Transfer is the traditional way to construct a satellite transfer to the Moon. It only uses two-body dynamics, as described in Section 2.1.4, and is constructed by determining an elliptic transfer orbit from an Earth parking orbit to the orbit of the Moon. It is however an expensive approach, when the ratio of the two radii of the orbits is large as it requires a large $\Delta V$. This subsection will therefore only describe it briefly. The information is mainly taken from (Sidi [2000]).

The initial orbit has radius $r_1$, and the second orbit has radius $r_2$. Two impulses are applied. The first is applied at the pericentre of the transfer orbit to get it from the first circular orbit to an elliptical transfer orbit. The second is applied at the apocentre of the transfer orbit, which corresponds to the orbital radius of the Moon. This is illustrated in Figure 3.2. When in Moon orbit, the gravitational field of the Moon will capture the satellite. To find the required $\Delta V$ to construct this transfer, the $\Delta V$ to get from the circular orbit to the elliptical orbit and the $\Delta V$ to get from the elliptical orbit to the Moon orbit, are added. The first $\Delta V$, $\Delta V_1$, is given by equation (3.9). The second $\Delta V$, $\Delta V_2$, is found in the same way as $\Delta V_1$, only the

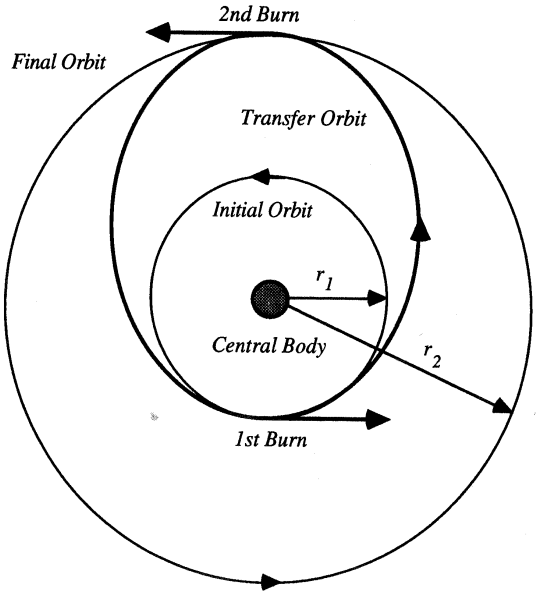



Figure 3.2: Hohmann transfer

velocity on the elliptical orbit is subtracted from the velocity on the larger circular orbit. $\Delta V_2$ becomes

$$\Delta V_2 = \sqrt{\frac{\mu}{r_2}} - \sqrt{\frac{2\mu}{r_2} - \frac{2\mu}{r_1 + r_2}}. \tag{3.12}$$

The resulting $\Delta V$ is found by $\Delta V = \Delta V_1 + \Delta V_2$. This gives

$$\Delta V = \sqrt{\frac{2\mu}{r_1} - \frac{2\mu}{r_1 + r_2}} - \sqrt{\frac{\mu}{r_1}} + \sqrt{\frac{\mu}{r_2}} - \sqrt{\frac{2\mu}{r_2} - \frac{2\mu}{r_1 + r_2}}. \tag{3.13}$$

### 3.2.2 Patched Conic Approximation Method (PCA)

The Patched Conic Approximation is a well-known method. When used on a transfer between Earth and the Moon, it is also referred to as the Lunar Patched Conic. It is a good way to make an approximation of a simulation of a lunar transfer orbit. Still it is restricted to the two-body problem, but more than one two-body problem are used, hence the name of the method.

This method is however not very likely to be chosen for the launch of the ESMO satellite as it is a rough approximation. It also has a condition that is hard to obtain; the lunar transfer orbit has to be coplanar. This is not often the case. For a coplanar trajectroy that is launched from a latitude of 28.5 degrees (Cape Canaveral), it is only possible when the inclination of the Moon's orbit is at its maximum. This happened in the first months of 1969 and will happen again in 2006, (Bate et al. [1971]).

The trajectory of the method is explained graphically in Figure 3.3. The trajectory is broken into regions that can be handeled. Here, gravity from one body at a time is considered in each region, that is, the problem is divided into three two-body problems. Earth's sphere of influence extends well beyond the orbit of the Moon, so the patched conic method is a rough approximation. The three regions of the transfer are (Sellers [1994])

1. Sun-centered transfer from Earth; Sun's gravitational pull dominates
2. Earth departure; Earth's gravitational pull dominates
3. Arrival at the Moon; Moon's gravitational pull dominates

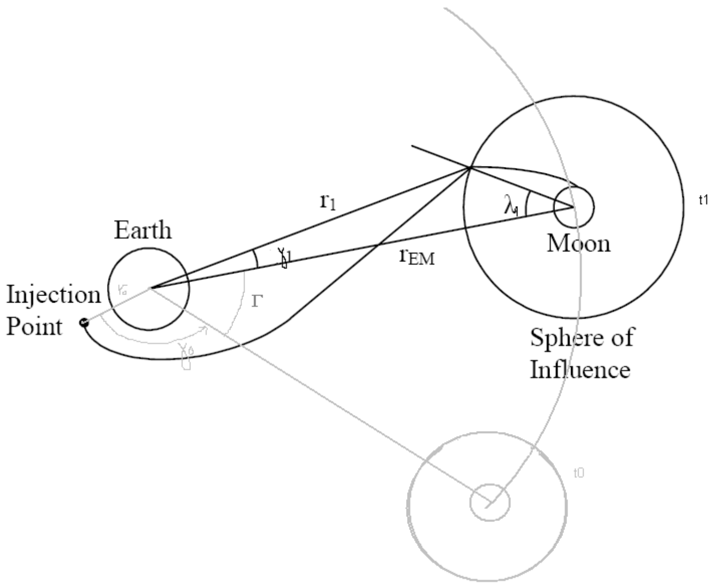



Figure 3.3: Lunar Patched Conic

A planet's sphere of influence (SOI) has been mentioned earlier. It is the area within where the planet's gravity dominates a satellite's motion. The size of the SOI depends on the planet's size and its distance to the Sun. The radius of a planet's SOI is given by (Sellers [1994])

$$r_s = a_{planet}\left[\frac{m_{planet}}{m_{Sun}}\right]^{\frac{2}{5}}, \tag{3.14}$$

where $a_{planet}$ is the semi-major axis of the planet's orbit around the Sun.

In the first region, motions are relative to the Sun. The problem is treated like the satellite is travelling from Earth's orbit around the Sun, to the planet's orbit around the Sun. In this problem, the planet is the Moon and it is not really orbiting the Sun like the planets, but as Earth is orbiting the Sun, so is the Moon. This trajectory is a Hohmann-transfer ellipse around the Sun. The Hohmann transfer was described in Section 3.2.1. In the second region, motions are relative to Earth. This is really the first part of the trajectory. Here, the satellite escapes Earth and arrives at the SOI with the required velocity to enter into the heliocentric transfer orbit of region one. The satellite needs to increase its velocity in the parking orbit by a certain amount. In the third region, motions are relative to the Moon. Here, the satellite needs to be slowed down. If not, it will only swing by the Moon on a hyperbolic trajectory and depart the SOI on the other side.

Calculations of the design of a mission can be divided into eight steps (Brown [1998]).

1. Initial conditions are set. These are the parameters of the transfer ellipse; the right injection radius $r_0$, velocity $v_0$ and the flight path angle $\gamma_0$. Another parameter that needs to be set, is the angle $\lambda$ which defines the arrival location at the Moon's sphere of influence, see Figure 3.3.

2. Define the ellipse. This can be done by using the energy equation (2.42); $E_el = \frac{1}{2}v_0^2 - \frac{\mu}{r_0}$. It is not necessary to reach the escape velocity for Earth as the departure trajectory is an ellipse rather than a hyperbola. If the injection velocity is not high enough, the initial conditions have to be set again. Another parameter to be found for the ellipse, is

its eccentricity. It can be found from

$$e = \sqrt{1 + 2E_{el}\frac{h^2}{\mu^2}}, \tag{3.15}$$

where the specific momentum $h$ is found from

$$h = r_0 v_0 \cos\gamma_0. \tag{3.16}$$

3. The radius to the sphere of influence, $r_1$, is found, see Figure 3.3. It is found by the cosine law as

$$r_1 = \sqrt{r_{EM}^2 + r_s^2 - 2r_{EM} r_s \cos\lambda}, \tag{3.17}$$

where $r_s$ is found by (3.14) and $r_{EM}$ is the distance between centres of mass for Earth and the Moon which is 384 400 km. $\lambda$ was found in step 1. The phase angle shown in Figure 3.4 can be found from the same triangle as

$$\phi_1 = \cos^{-1}\left[\frac{r_1^2 + r_{EM}^2 - r_s^2}{2r_1 r_{EM}}\right]. \tag{3.18}$$

Also the velocity on the elliptic orbit in the point where the satellite hits the SOI, can be found from equation (2.22) as

$$v_1 = \sqrt{\mu_e(\frac{2}{r_1} - \frac{1}{a_{el}})}, \tag{3.19}$$

where the semi-major axis of the ellipse is found from

$$a_{el} = -\frac{\mu_e}{2E_{el}}. \tag{3.20}$$

4. The time of flight to the sphere of influence boundary can be found by combining equation (2.35) and (2.34) to

$$t = \frac{\epsilon - e\sin\epsilon}{\sqrt{\frac{\mu}{a^3}}}, \tag{3.21}$$

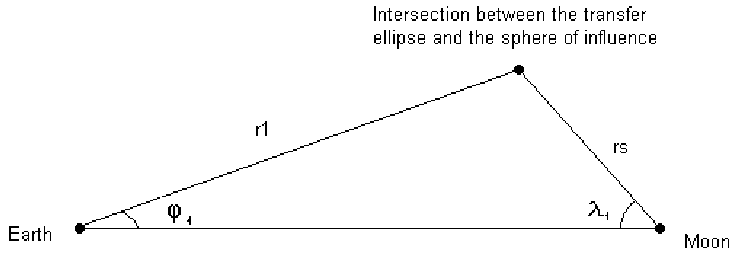



Figure 3.4: The phase angle and $r_1$

where $\epsilon$ is the eccentric anomaly. Equation (2.30) can be used to find $\epsilon$. Equation (2.30) uses the true anomaly $\nu_1$ to calculate $\epsilon$, and $\nu_1$ can be found by rearranging equation (2.36) and (2.37) to

$$\cos\nu_1 = \frac{\frac{r_0}{r_1}(1+e)-1}{e}, \tag{3.22}$$

where $r_0$ (corresponds to $r_p$ in equation (2.36)) was found in step 1, e in step 2 and $r_1$ in step 3.

5. So far, the parameters for the flight to the sphere of influence are found. Now the parameters inside the sphere of influence need to be found. The necessary parameters are the velocity $v_2$, the flight path angle $\sigma$ and the radius $r_2$, which is constant as it is the Moon's sphere of influence, which is 66 300 km. The cosine law can find $v_2$ to be

$$v_2 = \sqrt{v_m^2 + v_1^2 - 2v_m v_1 \cos\alpha}, \tag{3.23}$$

where the velocities are shown in Figure 3.5. To calculate this the angle $\alpha$ needs to be found. It is calculated as

$$\alpha = \gamma_1 - \phi_1, \tag{3.24}$$

where $\gamma_1$ is the flight path angle calculated as

$$\gamma_1 = \tan^{-1}\left[\frac{e\sin(\nu)}{1+e\cos(\nu)}\right]. \tag{3.25}$$

The angle, $\sigma$, associated with the velocity $v_2$ inside the sphere of influence, can be found from Figure 3.5 as

$$\sigma = \sin^{-1}\left[\frac{v_m}{v_2}\cos\lambda - \frac{v_1}{v_2}\cos(\lambda + \phi_1 - \gamma_1)\right]. \tag{3.26}$$

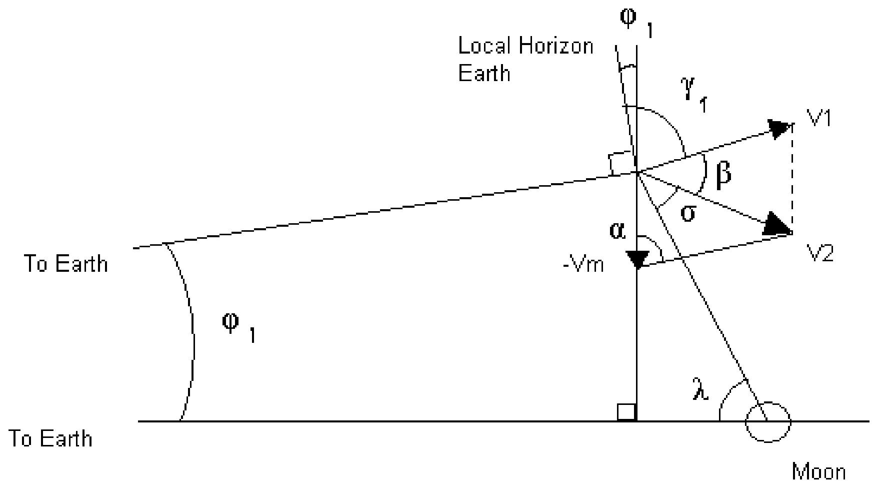



Figure 3.5: Geometry of the lunar arrival

6. The definition of the arrival orbit is made. Parameters that are to be found are the specific enery $E$ for the lunar orbit, the specific momentum $h$, the semimajor axis $a$ and the eccentricity $e$. Starting with $E$, equation (2.42) gives

$$E = \frac{v_2^2}{2} - \frac{\mu_m}{r_2}, \tag{3.27}$$

where $\mu_m = 4902.8km^3/s^2$. The specific momentum can be calculated by

$$h = r_2 v_2 \sin(\sigma). \tag{3.28}$$

The semimajor axis is found by rearranging equation (2.44) to

$$a = -\frac{\mu_m}{2E}, \tag{3.29}$$

and the eccentricity is given by (3.15) to be

$$e = \sqrt{1 + 2E\frac{h^2}{\mu_m^2}}. \tag{3.30}$$

The radius of the pericentre is found to be

$$r_p = \frac{h^2}{\mu_m(1+e)}. \tag{3.31}$$

Also, the satellite will hold a certain velocity around the Moon. This velocity depends upon the type of orbit. The velocities for the different orbits can be found in Section 2.2.1 and 2.2.2. The obtained orbit might not be the desired to orbit the Moon. Therefore, more manoeuvres might be required to descend the satellite to a suitable orbit around the Moon. A Hohmann manoeuvre is often used in an approximated method as this.

7. By now, all necessary parameters are set. The launch day may now be found using the time of flight and the average orbital velocity.

8. If the desired orbit is not achieved by the calculated arrival orbit, the initial conditions needs to be adjusted and the process starts at step one again.

As the Moon orbits Earth, naturally the position of the Moon at the time of injection and arrival of the satellite are not the same. The angle between the position of the Moon at injection and at arrival, see Figure 3.3, can be calculated as (Marthinussen [2004])

$$\Gamma = 13.177(t_{arrival} - t_{injection}). \tag{3.32}$$

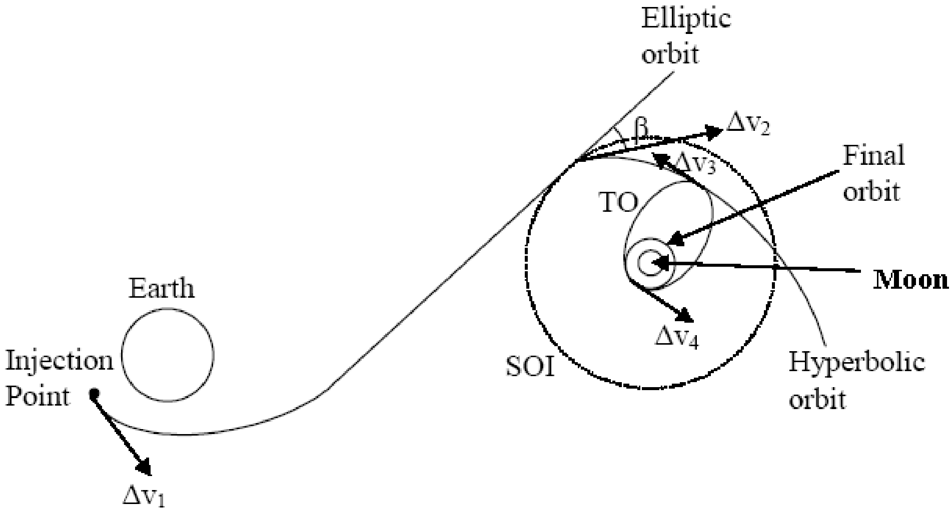



Figure 3.6: $\Delta$Vs in the Lunar Patched Conic

### 3.2.3 Ballistic Capture Transfer (BCT)

BCT is one of the most fuel-efficient methods used to put a spacecraft into orbit around the Moon, (Belbruno and Carrico [2000]). It takes advantage of the Moon's sphere of influence (SOI) to avoid having to use an impulse $\Delta V$ to get the satellite from the transfer orbit to the orbit around the Moon. This way the satellite only requires one $\Delta V$ to get the satellite from Earth to the Moon. The main drawback with this is that the satellite's Moon orbit is not stable. After a few days, the satellite will have gained enough energy from its orbit to escape from the sphere of influence again. This period can however be increased by applying a small extra $\Delta V$ when in Moon orbit. The figures in this section are taken from (Koon et al. [2001]).

The only required $\Delta V$ is applied at Earth. This launches the satellite into a hyperbolic orbit. When the satellite reaches the Moon's sphere of influence with hyperbolic velocity, the Moon will capture it and keep it in orbit around itself for a few days. The way to design this transfer, is to think of it backwards. Start with the inital conditions that are desired for the lunar orbital elements. Then propagate the trajectory back in time to reach an orbit that barely escapes the Moon, travels to the Earth-Sun sphere of influence, and

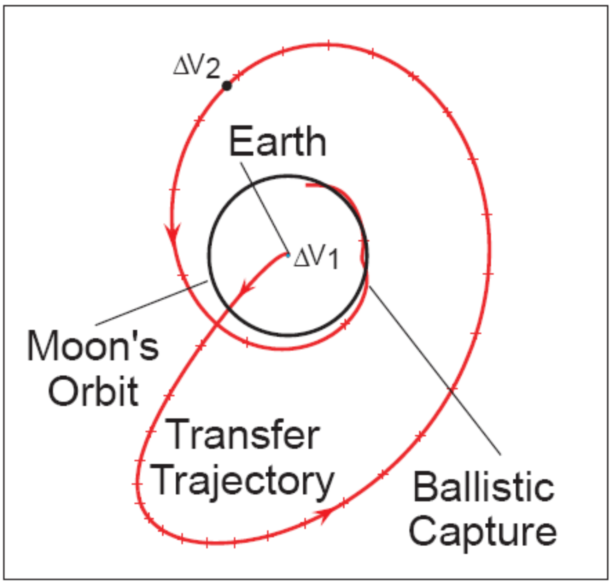



Figure 3.7: BCT

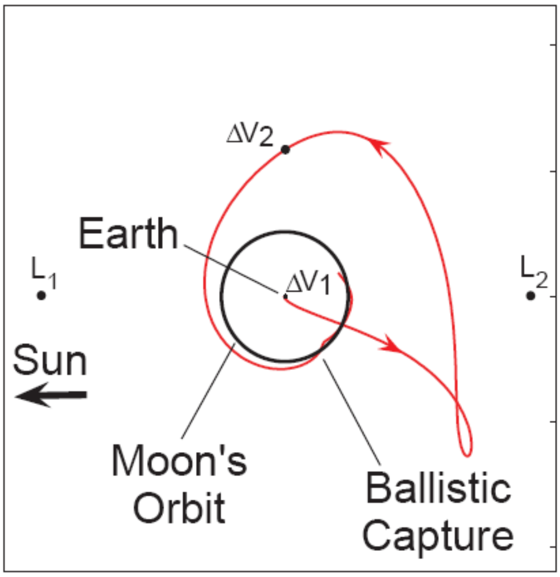



Figure 3.8: BCT in Sun-Earth rotatin frame

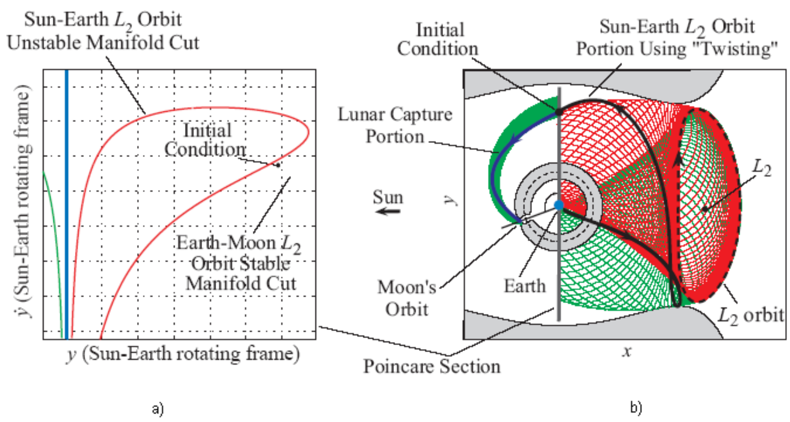



Figure 3.9: Different views of the Poincaré section

ends up with dropping back to pass Earth at a very close range, (Belbruno and Carrico [2000]).

The first step in the design of the trajectory is therefore to choose the initial state so that the Moon escapeing orbit can be found. The velocity vector is parallel to the ecliptic plane[1] and perpendicular to the position vector. The other elements are chosen so that the point of perilune is on the Earth side of a line which connects Earth and the Moon. This is done because it gives optimal approach to the Moon for the satellite. This initial state (position and velocity) should be on the Poincaré section $\Gamma$. An explanation of $\Gamma$ is found in (Koon et al. [2001]) and will not be explained further here as the BCT is not the chosen simulation approach. This section helps to glue the Sun-Earth Lagrange point portion of the trajectory with the lunar ballistic capture portion.

The second step is to choose the eccentricity. The eccentricity is adjusted until it is at its minimum for the satellite to escape the Moon. When the escape orbit is found, the eccentricity is increased with very small amounts to reach an orbit that will travel past the Earth-Sun sphere of influence and then return back to Earth. Some orbits will pass Earth at a very close range. These are the one used to design BCT trajectories by reversing time again.

A standard mission design approach is to view the solar system as a series of

[1] see Figure 2.4

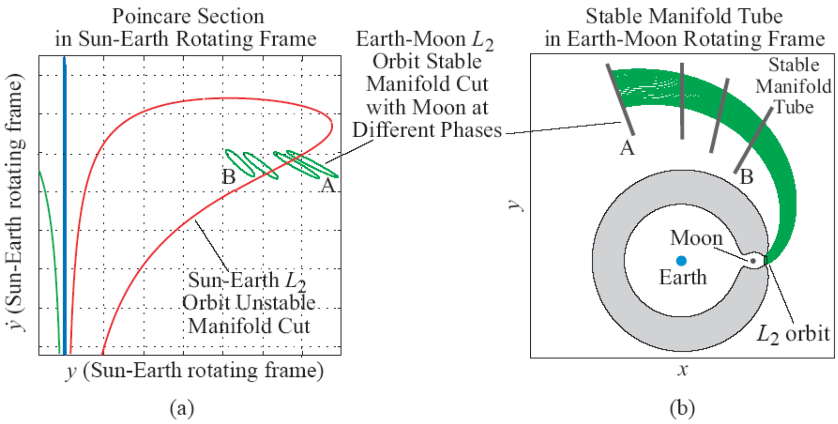



Figure 3.10: Vary the phase of the Moon until Earth-Moon $L_2$ manifold cut intersects Sun-Earth $L_2$ manifold cut.

two-body problems where Keplerian theory applies. But when the ballistic capture regime of motion is dealt with, a three-body decomposition of the Solar System is absolutely necessary, (Koon et al. [2001]). The system considered here is really a four-body system (Earth, Moon, Sun and Satellite). Since the structure of the phase space of the four-body system is poorly understood in comparison with the three-body system, it is more convenient to model it as two coupled planar circular restricted three-body systems. In doing so, the Lagrange point dynamics[2] of both the Earth-Moon-satellite and Sun-Earth-satellite systems can be utilized.

As mentioned earlier, the mission is designed by propagating back in time. With the right initial state, the satellite will be guided by the $L_2$ Earth-Moon manifold[3] and get ballistically captured by the Moon. Looking at Figure 3.10 the orbit should lie in the interior of the gray curve but in the exterior of the black curve.

The BCT would be a good alternative for bringing ESMO to the Moon as it is fuel efficient.

[2]Further explanation in (Koon et al. [2001]) as the BCT is not the chosen simulation approach.

[3]Further explanation in (Koon et al. [2001]) as the BCT is not the chosen simulation approach.

# Chapter 4

# Control

In the presented two- and three-body problems, many perturbing effects are ignored. Some are mentioned in Section 2.3. To make sure the satellite follows the desired trajectory, feedback control can be applied. A sketch of what is meant by this is found in Figure 4.1. It is assumed that the satellite has a thruster in the x-, y- and z-direction.

Ideally, to apply feedback control to the satellite orbit, the real values of the position and velocity of the satellite are compared to the desired values. Then the desired thrust direction is determined based on the chosen control law.

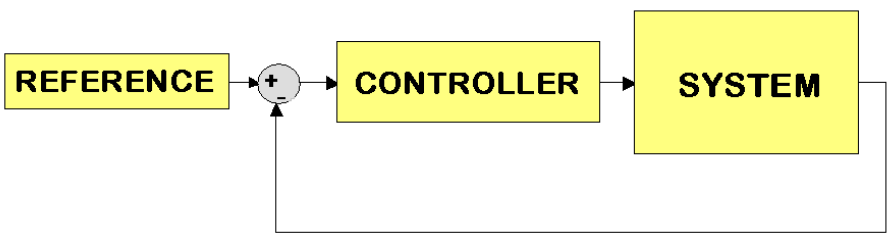



Figure 4.1: A standard feedback system

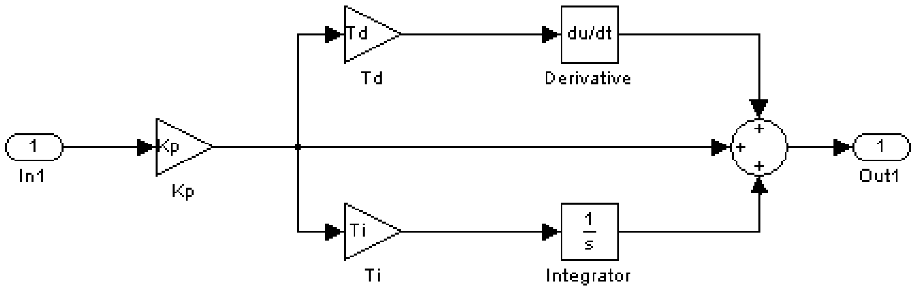



Figure 4.2: Block diagram of PID controller

## 4.1 PID Controllers

The controller needed to make the satellite follow the desired trajectory is not expected to be very advanced. Therefore, a PID (proportional, integral, derivative) controller is a natural choice. The PID controller is a popular controller, therefore many of its characteristics are well known.

A block diagram of an ideal PID controller is shown in Figure 4.2. The corresponding transfer function is given as

$$h_r(s) = K_p\big(1 + \frac{1}{T_i s} + T_d s\big) = K_p \frac{1 + T_i s + T_i T_d s^2}{T_i s}. \tag{4.1}$$

The parameters deciding the characteristics of the controller, are $K_p$, $T_i$ and $T_d$. If $T_d$ is set to zero, the controller is a PI controller and, if in addition $T_i$ is set to $\infty$, the controller is a P controller.

A way to tune the PID controller is to use Ziegler-Nichols method, (Balchen et al. [2002]). This method is used for experimental tuning of controllers. To use this method, the system has to be stable. The method can in short terms be described as follows:

- Set $T_i = \infty$ and $T_d = 0$. This gives a pure P-controller
- Increase the gain $K_p$ until the system gives a standing oscillation on the system exit. This value is called the critical gain and is denoted $K_{pk}$.

| *Controller* | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5K_{pk}$ | $\infty$ | 0 |
| PI | $0.45K_{pk}$ | $0.85T_k$ | 0 |
| PID | $0.6K_{pk}$ | $0.5T_k$ | $0.12T_k$ |

Table 4.1: Ziegler-Nichols rules

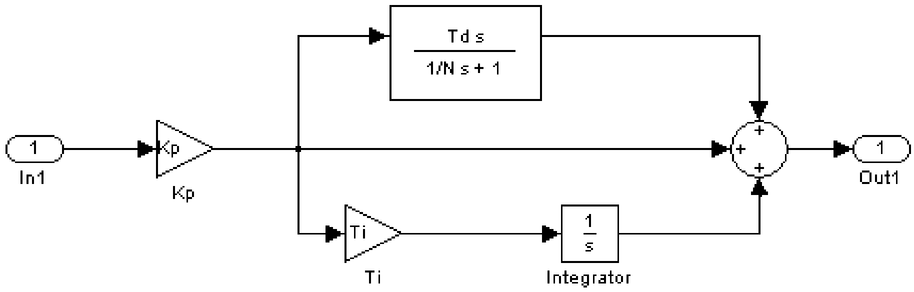



Figure 4.3: A limited PID controller

- The period of the oscillation is denoted $T_k$.
- The values $K_{pk}$ and $T_k$ are used to find the parameters of the desired controller. How this is done is found in Table 4.1.

Ziegler-Nichol's method is primarily used on scalar, linear systems. If the state is a vector instead of a scalar, one of the states have to be chosen to tune by. Alternatively, many of the states could be tuned and a compromise that satisfy most of them could be chosen. There is no common rule on how to do this except experimental try and fail. The same goes for nonlinear systems.

Another version of the PID controller that is often used, is the limited PID controller, as seen in Figure 4.3. Its transfer function is found to be

$$h_r(s) = K_p(1 + \frac{T_i}{s} + \frac{T_d s}{\frac{1}{N}s + 1}) = K_p \frac{T_i + (1 + \frac{T_i}{N})s + (T_d + \frac{1}{N})s^2}{s(\frac{1}{N}s + 1)}. \quad (4.2)$$

## 4.2 Nonlinear controller and stability

Nonlinear systems are dealt with in detail in (Khalil [2000]). In this thesis, the systems will be on the form

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

that are nonlinear, nonautonoumus systems. The uncontrolled, nonperturbed, differential equations for the satellite, derived from equation (2.5)[1], are

$$\begin{aligned} \dot{\vec{r}} &= \vec{v} & (4.4)\\ \dot{\vec{v}} &= -Gm_2\frac{\vec{r}_E}{r_E^3} - Gm_4\frac{\vec{r}_{Sun}}{r_{Sun}^3}, & (4.5)\end{aligned}$$

where $G$ is the gravitational constant, $m_2$ is the mass of Earth and $m_4$ is the mass of the Sun. When used in a feedback loop, a control input $u$ will be included in the equations, and the errors will be $\Delta r = r_{controlled} - r_{reference}$ and $\Delta v = v_{controlled} - v_{reference}$. With $u$ as a velocity control input, the error system is

$$\begin{aligned} \Delta\dot{r} &= \Delta v + u & (4.6)\\ \Delta\dot{v} &= \dot{\vec{v}}_{controlled} - \dot{\vec{v}}_{ref} \\ &= -Gm_2\frac{\vec{r}_E}{r_E^3} - Gm_4\frac{\vec{r}_{Sun}}{r_{Sun}^3} - (-Gm_2\frac{\vec{r}_{Eref}}{r_{Eref}^3} - Gm_4\frac{\vec{r}_{Sunref}}{r_{Sunref}^3}) \\ &= -Gm_2\left(\frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3}\right) - Gm_4\left(\frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3}\right), & (4.7)\end{aligned}$$

and if the control $u$ is set to be a force, or an acceleration, the error system is

[1] more on the dynamic model in Chapter 5.2

$$\begin{aligned}
\Delta \dot{r} &= \Delta v && (4.8) \\
\Delta \dot{v} &= \dot{\vec{v}}_{controlled} - \dot{\vec{v}}_{ref} + u \\
&= -Gm_2 \frac{\vec{r}_E}{r_E^3} - Gm_4 \frac{\vec{r}_{Sun}}{r_{Sun}^3} - (-Gm_2 \frac{\vec{r}_{Eref}}{r_{Eref}^3} - Gm_4 \frac{\vec{r}_{Sunref}}{r_{Sunref}^3}) + u \\
&= -Gm_2 \left( \frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3} \right) - Gm_4 \left( \frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3} \right) + u. && (4.9)
\end{aligned}$$

To be able to say something about the stability of the systems, Lyapunov stability theorems can be applied to these systems. The following theorems are taken from (Khalil [2000]).

**Theorem 4.1** [Theorem 4.8 Khalil [2000]]*Let $x = 0$ be an equilibrium point for (4.3) and $D \subset R^n$ be a domain containing $x = 0$. Let $V : [0, \infty) \times D \rightarrow R$ be a continouously differentiable function such that*

$$W_1(x) \leq V(t, x) \leq W_2(x) \tag{4.10}$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq 0 \tag{4.11}$$

*$\forall\ t \geq 0$ and $\forall\ x \in D$, where $W_1(x)$ and $W_2(x)$ are continuous positive definite functions on D. Then, $x = 0$ is uniformly stable.*

□

**Theorem 4.2** [Theorem 4.9 Khalil [2000]]*Suppose the assumptions of Theorem 4.8 are satisfied with inequality (4.11) strengthened to*

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -W_3(x) \tag{4.12}$$

*$\forall\ t \geq 0$ and $\forall\ x \in D$, where $W_3(x)$ is a continuously positive definite function on D. Then, $x = 0$ is uniformly asymptotically stable. Moreover, if r and c are chosen such that $B_r = \{\parallel x \parallel \leq r\} \subset D$ and $c \leq min_{\|x\|=r} W_1(x)$, then every trajectory starting in $\{x \in B_r \mid W_2(x) \leq c\}$ satisfies*

$$\parallel x(t) \parallel \leq \beta(\parallel x(t_0) \parallel, t - t_0),\ \forall\ t \geq t_0 \geq 0 \tag{4.13}$$

*for some class $\mathcal{KL}$ function $\beta$. Finally, if $D = R^n$ and $W_1(x)$ is radially unbounded, then $x = 0$ is globally uniformly asymptotically stable.*

□

First of all, a Lyapunov function that satisfies equation (4.10) in Theorem 4.1 has to be found for the error systems. A standard choice that can be applied is (Naasz [2002])

$$V(\Delta r, \Delta v) = \frac{1}{2}k_1 \Delta r^T \Delta r + \frac{1}{2}k_2 \Delta v^T \Delta v, \tag{4.14}$$

The time derivative of equation (4.14) is

$$\dot{V}(\Delta r, \Delta v) = \begin{bmatrix} k_1 \Delta r \\ k_2 \Delta v \end{bmatrix}^T \begin{bmatrix} \Delta \dot{r} \\ \Delta \dot{v} \end{bmatrix}, \tag{4.15}$$

With the error system with control in the velocity, see equation (4.6) and (4.7), inserted in equation (4.15), $\dot{V}$ is

$$\begin{aligned} \dot{V} \;&=\; k_1 \Delta r^T (\Delta v + u) - \\ &\quad k_2 \Delta v^T \left( Gm_2 \left( \frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3} \right) + Gm_4 \left( \frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3} \right) \right), \end{aligned} \tag{4.16}$$

and with the control in the acceleration, as given in equation (4.8) and (4.9), $\dot{V}$ is

$$\begin{aligned} \dot{V} \;&=\; k_1 \Delta r^T \Delta v - k_2 \Delta v^T \Bigg( Gm_2 \left( \frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3} \right) \\ &\quad + Gm_4 \left( \frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3} \right) - u \Bigg). \end{aligned} \tag{4.17}$$

Equation (4.17) will now be used to design control laws. Equation (4.16) will not be used further as it is more physically correct to apply the control in the acceleration than in the velocity.

### 4.2.1 Nonlinear controller

To design a nonlinear controller, Lyapunov theory can be applied. In this case there is a time-varying reference, so the theory has to be applicable to nonautonomous systems. This was derived in Section 4.2. To ensure that the resulting control of the system is stable, the control input $u$ in equation (4.17) is chosen so that it cancels the possibly positive terms in the same equation. The resulting $u$ is

$$\begin{aligned} u &= -\frac{k_1}{k_2}\Delta r - k_3\Delta v \\ &\quad + Gm_2\left(\frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3}\right) + Gm_4\left(\frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3}\right), \end{aligned} \tag{4.18}$$

giving the error system

$$\Delta\dot{r} = \Delta v \tag{4.19}$$

$$\Delta\dot{v} = -\frac{k_1}{k_2}\Delta r - k_3\Delta v. \tag{4.20}$$

The resulting $\dot{V}$ is

$$\dot{V} = -k_2 k_3 \Delta v^T \Delta v \leq 0. \tag{4.21}$$

Hence, from Theorem 4.1 it can be seen that the system with the nonlinear controller is uniformly stable.

However, to show asymptotic stability, Theorem 4.2 states that it is needed to show that $\dot{V} \leq -W_3$, see equation (4.12). This can not be found as the expression in equation (4.21) will be zero if $\Delta v = 0$.

Another approach can be taken. The system can be shown to be stable and convergent using a version of Barbalat's lemma restated in Lemma 4.1 taken from (Slotine and Li [1991]).

**Lemma 4.1** [Lemma 4.3 Slotine and Li [1991]]*If a scalar function $V(x,t)$ satisfies the following conditions*

- *$V(x,t)$ is lower bounded*
- *$\dot{V}(x,t)$ is negative semi-definite*
- *$\dot{V}(x,t)$ is uniformly continuous in time*

*then $\dot{V}(x,t) \to 0$ as $t \to \infty$.*

□

The first criteria was shown with equation (4.14) and the second with equation (4.21). To use this lemma, the third criteria also have to be checked. The derivative of $\dot{V}$ is

$$\begin{aligned}
\ddot{V} &= -k_2k_3\Delta v^T\Delta\dot{v} - k_2k_3\Delta\dot{v}^T\Delta v \\
&= -2k_2k_3\Delta v^T\Delta\dot{v} \\
&= -2k_2k_3\Delta v^T(-\frac{k_1}{k_2}\Delta r - k_3\Delta v) \\
&= 2k_1k_3\Delta v^T\Delta r + 2k_2k_3^2\Delta v^T\Delta v.
\end{aligned} \tag{4.22}$$

This shows that $\ddot{V}$ is bounded, since $\Delta r$ and $\Delta v$ were shown by equation (4.21) to be bounded. Therefore, $\dot{V}$ is uniformly continuous and hence, $\dot{V}(x,t) \to 0$ as $t \to \infty$. The system is stable and convergent.

**Remark 4.1** *The system (4.19) and (4.20) has through feedback linearization become linear. Regular linear systems theory can however not be applied as it is only applicable to linear time-invariant (LTI) systems, and this is an error system that has a time-varying reference.*

## 4.3 Propulsion use

The best possible controller might not be very propulsion efficient. As it is desirable with a controller that works well and at the same time is propulsion efficient, compromises have to be made.

The propellant needed depends on the size of the impulses needed, see Section 3.1.1. It also depends on the kind of propellant used, but even if propellant with low $I_{sp}$ is used, the size of the impulses should still not be too high.

If the satellite's thrusters give an impulse each time the satellite gets a bit off course, it will use a lot of propellant. Therefore, to reduce the use a propellant, a limit can be set as to how far away from the desired orbit the satellite should be before the thrusters are fired. This will however reduce the satellite's ability to stay directly on course.

# Chapter 5

# Simulations

In this section, a satellite trajectory calculated by the PCA is simulated. This is first done without perturbations, then with perturbations and last with perturbations and different controllers. The chosen platform for the simulations is MatLab/Simulink[1].

## 5.1 Numerical integrators

To propagate the satellite, the equations of motion are integrated. The different n-body problems used requires different numerical integrators. As a general rule, it is easily seen that the higher the number of n, the higher an accuracy of the numerical integrator is needed.

### 5.1.1 MatLab solvers

There are two main choices in numerical integrators in MatLab; fixed-step and variable-step.

Fixed-step solvers solve the model at regular time intervals from the beginning to the end of the simulation. The size of the interval is known as the step size. The step size can be specified manually, or the solver can choose

[1]The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098,USA http://www.mathworks.com

it. Generally, decreasing the step size increases the accuracy of the results and increases the time required to simulate the system.

Some of the fixed-step solvers are the ode1 and the ode4 solvers. ode1 is a simple Euler method and ode4 is a Runge-Kutta 4.

Variable-step solvers vary the step size during the simulation, reducing the step size to increase accuracy when a model's states are changing rapidly and increasing the step size to avoid taking unnecessary steps when the model's states are changing slowly. Computing the step size adds to the computational overhead at each step but can reduce the total number of steps, and hence simulation time, required to maintain a specified level of accuracy for models with rapidly changing or piecewise continuous states.

Some of the most accurate variable-step solvers, as defined in (MatLab [2004]):

ode45 (MatLab [2004]); Based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair. It is a one-step solver - in computing $y(t_n)$, it needs only the solution at the immediately preceding time point, $y(t_{n-1})$. In general, ode45 is the best function to apply as a "first try" for most problems.

ode113 (MatLab [2004]); Variable order Adams-Bashforth-Moulton PECE solver. It may be more efficient than ode45 at stringent tolerances and when the ODE function is particularly expensive to evaluate. ode113 is a multistep solver - it normally needs the solutions at several preceding time points to compute the current solution.

### 5.1.2 Choosing solvers for the different N-body problems

As the two-body problem is the least complex of the n-body problems, it does not require a highly advanced integration method. In earlier simulations (Johansson [2004]) the ode4 has been applied with success in Earth and Moon orbits.

For the three-body problem, some of the solvers described in Section 5.1.1 were tested on a periodic solution of a normalized system with a satellite, Earth and the Moon, as such a system requires high accuracy. From Figure

5.1 it can easily be seen that the last method tried, the ode45 with relative tolerance $10^{-6}$, has to be applied to the three-body problem to get a satisfactory result.

Even the ode45 might come a bit short when applied to a higher n-body problem as it is not much over adequate for the three-body problem. The ode113 might be appropriate.

## 5.2 Trajectories and control

### 5.2.1 Dynamic model

To simulate the different stages of the trajectory, a common set of differential equations can be applied. The implementation is shown in Appendix B.1 and is taken from (Jerpetjøn [2004]). These equations are derived from equation (2.5) and includes a satellite, Earth, the Moon and the Sun. The state vector $x$ represents the 24 states of a four-body system. $x(1) - x(6)$ represents respectively the satellite's position in $x$, $y$ and $z$ coordinates and its velocity in the same coordinates. The origo is in the centre of the Sun. $x(7) - x(12)$ represents the same for Earth, $x(13) - x(18)$ for the Moon and $x(19) - x(24)$ for the Sun. As the Sun is considered the non-moving origo, the initial conditions $x0(19) - x0(24)$ is set to zero.

Also, in the simulations, everything is considered to be in the xy-plane. Therefore, all the $z$ components are initially also set to zero.

### 5.2.2 PCA

The PCA is the chosen trajectory in this thesis for use in the simulations. Parameters and initial conditions for the simulations of the PCA can be found in Table 5.1, see also Figure 5.2. As Earth is orbiting the Sun, and the Moon is orbiting Earth, these velocities have to be added to the velocity the satellite has relative to Earth or the Moon as found in the calculations from the PCA, when appropriate. For example, the initial velocity of the Moon is found as $v_{Earth\ rel\ to\ Sun} + v_{Moon\ rel\ to\ Earth}$ in the $y$-direction as all the velocities have to be relative to the Sun in the simulations.

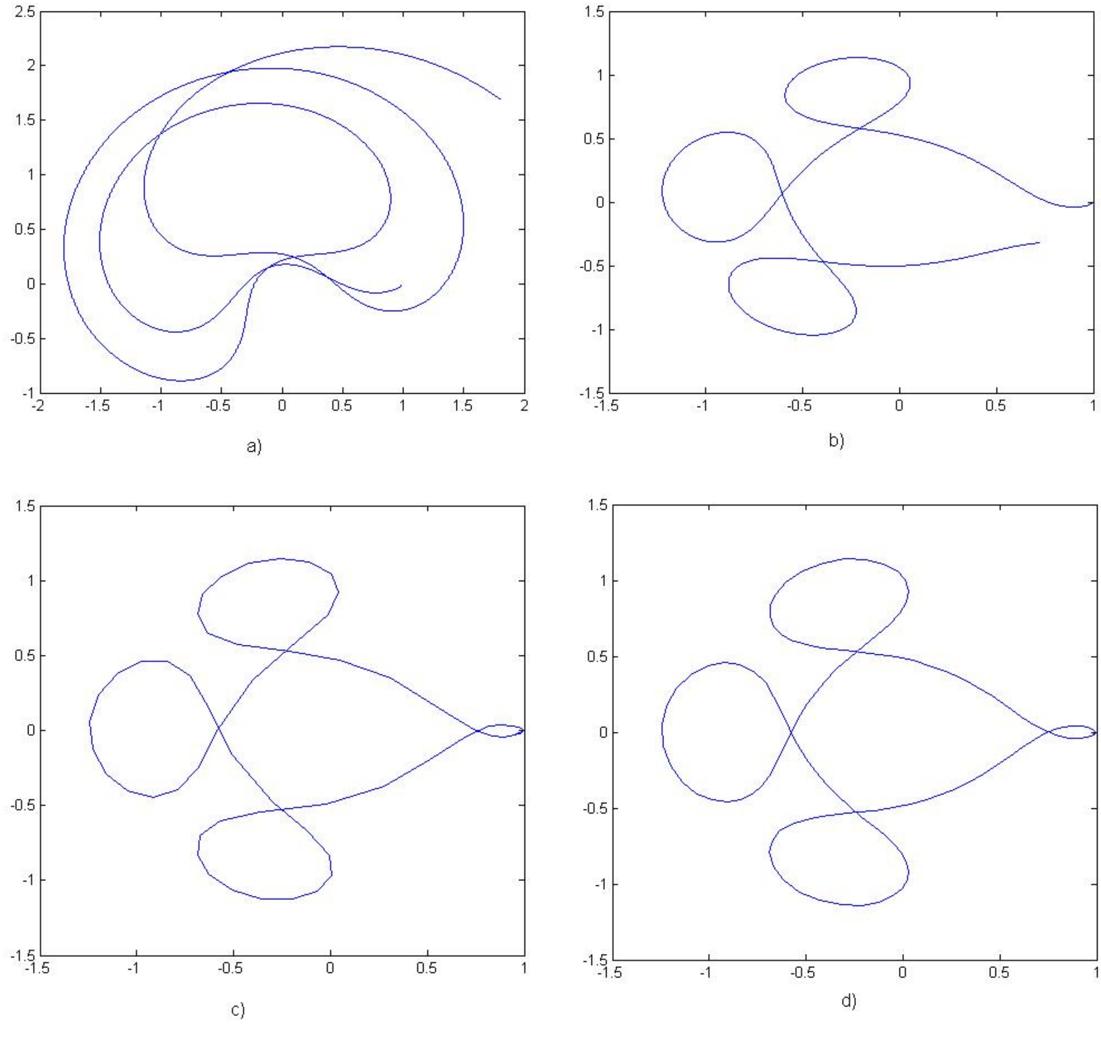



Figure 5.1: A periodic orbit simulated with the solver; a) ode1 with step size $7*10^{-4}$ b) ode4 with step size $3*10^{-3}$ c) ode45 with relative tolerance $10^{-3}$ d) ode45 with relative tolerance $10^{-6}$

| *Name* | *Value* |
|---|---|
| r Sun to Earth | 1.5e11 m |
| r Earth to Moon | 356000000 m |
| r satellite to Earth | 24 000 000 m |
| Velocity of Earth relative to Sun in y-direction | 29680 m/s |
| Velocity of Moon relative to Earth in y-direction | 1092.9 m/s |
| Velocity of sat. rel. to Earth in y-dir. before $\Delta V$, non-phased | 4084.4 m/s |
| $\Delta V$ from initial orbit to transfer ellipse | 1515 m/s |
| $\Gamma$, angle of phaseing | 0.67189 rad |
| Velocity of Moon relative to Sun in x-direction | 0 m/s |
| Velocity of Moon relative to Sun in y-direction | 30772.9 m/s |
| Velocity of satellite relative to Sun in x-direction, phased | 4578.9320 m/s |
| Velocity of satellite relative to Sun in y-direction, phased | 26457.1656 m/s |

Table 5.1: Initial values for the PCA

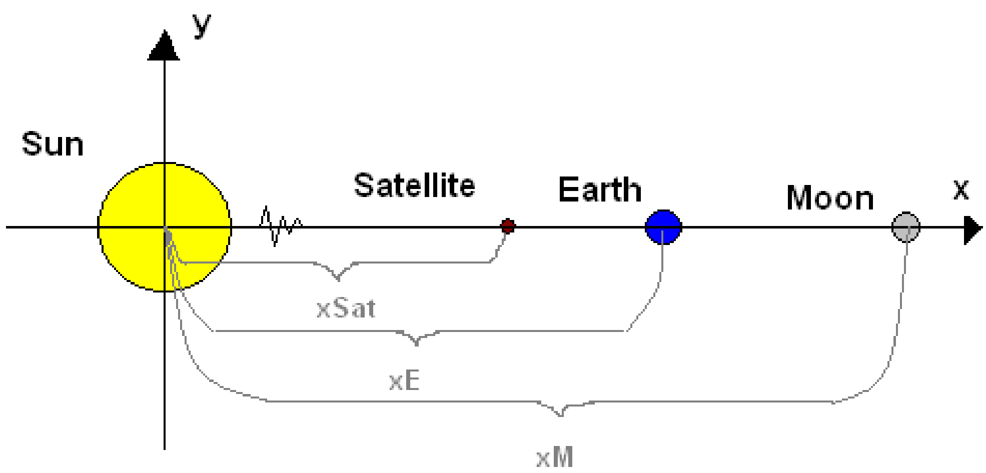



Figure 5.2: The initial distances (not to scale)

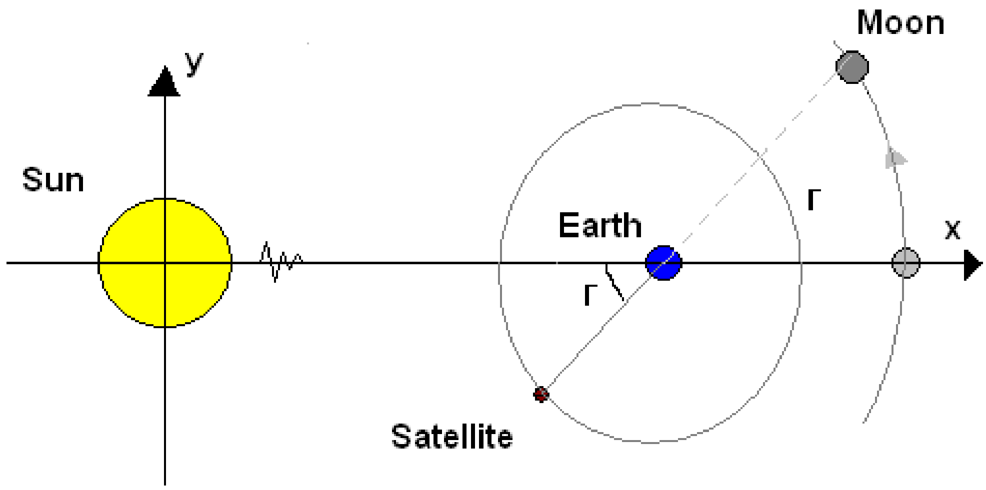



Figure 5.3: The phased initial distances (not to scale)

The transfer ellipse intersects the SOI of the Moon at an angle that can make it hard to reach the desired Moon orbit, (Johansson [2004]). Therefore, the impulse that transfers the satellite from a circular Earth orbit to the elliptic transfer orbit, is applied slightly later, more specifically at the angle $\nu = \frac{\pi}{11}$.

As the Moon moves in the simulation, the initial position of the satellite needs to be phased so that the satellite will be put into a transfer ellipse that ends up where the Moon is going to be at the time of the arrival of the satellite, see Figure 5.3. The angle $\Gamma$ is found from equation (3.32), and the angle $\nu$ is added to this. In this case $t_{injection}$ is set to zero and $t_{arrival}$ is found from the simulations to be the time when the satellite is closest to the Moon on its transfer orbit.

To simulate the PCA trajectory, the $\Delta V$ impulses is first found by running the MatLab code in Appendix A, which implements the equations described in Section 3.2.2. However, certain changes have to be made as the PCA is a strictly theoretical way of finding a trajectory to the Moon.

The second $\Delta V$ found, the one that transfers the satellite from the elliptic transfer orbit to a hyperbola around the Moon, should not be added in the simulations. This is because it only gives the difference in the velocities the

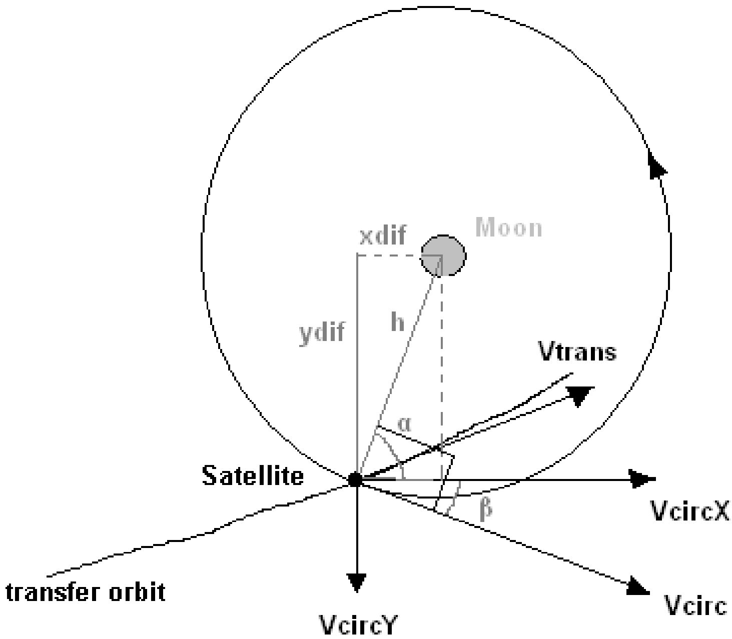



Figure 5.4: The arrival angle of the satellite

satellite has relative to Earth and relative to the Moon.

As the second $\Delta V$ is not part of the simulations, the next two $\Delta V$s calculated are not applicable either in the simulations. To put the satellite in an orbit around the Moon, another approach is taken. When the satellite reaches the point on the transfer ellipse that is closest to the Moon, the radius down to the centre of the Moon is found. Then, the desired new velocities to make the satellite orbit the Moon in a circle with this radius are found, and the appropriate $\Delta V$s are calculated from the state of the end transfer, see Appendix B.3. Here, the simple formula shown in equation (2.48) is used to find the velocity on a circular orbit with a certain radius. This is the tangential velocity in the circle. To find how much impulse to apply in each direction, the angle of arrival is found from the simulations, see Figure 5.4 and Appendix B.3.

If a smaller radius is desired, the velocity of an ellipse, as shown in equation (2.46), can be applied instead with the desired radius as the perigee. When

| *Impulse* | *Value* |
|---|---|
| $\Delta V_1$ | 1515 |
| $\Delta V_{2x}$ | -1249.2 |
| $\Delta V_{2y}$ | 308.7 |

Table 5.2: $\Delta V$s used in the simulations

in perigee, another $\Delta V$ can be applied to bring the satellite over on a circular orbit. The $\Delta V$s used in the simulations is found in Table 5.2

To add the $\Delta V$s at the correct time in the simulations, the simulation is run in two parts, see Appendix B.2. First the transfer ellipse is simulated for the desired time. Then, the found $\Delta V$ are added. This can be done in two different ways that give the same result.

First, the $\Delta V$s were added to the previous velocity of the satellite and the simulation is continued with the previous state parameters, included the added $\Delta Vs$, as initial conditions. This is not a very physically correct way of doing it. It is more natural to convert the $\Delta Vs$ to accelerations (or forces) and add them over a short time interval to the acceleration of the satellite. Therefore, this is the way $\Delta Vs$ are added in the simulations when control is applied and Simulink is used.

Then, the simulations with the satellite in orbit around the Moon is run for however long it is desired.

### 5.2.3 Control of the satellite

The different controlled satellite trajectories are simulated in Simulink. Both control schemes have the same main configuration, see Appendix C.1 and C.2. It consists of the differential equations of the system, which are called as a MatLab function, a feedback to a controller, and a control input from the controllers to the system again. The controller receives a signal that is the difference between a reference signal and the current states of the system. The reference signal is obtained by running the system without perturbations and correction in the differential equations, see Appendix D.1. The true position of the satellite is simulated by a perturbed model. In the physical satellite this position will be given by the sensors on the satellite.

| *Perturbation* | *Function in x-direction* |
|---|---|
| Moon | $(Gm_3(x_M - x_E))/\|\vec{r}_{Sat} - \vec{r}_M\|^3$ |
| Oblateness of Earth | $GA_{J2}(15\frac{(x_{Sat}-x_E)(z_{Sat}-z_E)^2}{\|\vec{r}_{Sat}-\vec{r}_E\|^7} - 3\frac{x_{Sat}-x_E}{\|\vec{r}_{Sat}-\vec{r}_E\|^5})$ |
| Atmospheric drag | $-((0.5\rho C_d A v v_x)/m_{Sat})$ |
| Solar radiation pressure | $((KPAx_{Sat})/(\|\vec{r}_{Sat} - \vec{r}_{Sun}\| m_{Sat}))$ |

Table 5.3: Functions for perturbations

The MatLab code for the perturbed system with PID control is found in Appendix D.2, and with nonlinear control in Appendix D.3. The perturbations available for simulations are listed in Table 5.3.

However, only the gravitational perturbation from the Moon was used in the simulations as other perturbing forces were negligible compared to the gravity force from the Moon. The controlled simulations are only performed on the transfer orbit. This is because the Moon is orbiting Earth. Therefore, the Moon and Earth have to be part of the reference of the simulation when the satellite is in Moon orbit, even though only the Moon should be considered according to the PCA method. Thus there are no large perturbations in Moon orbit and there is no need to simulate this each time.

# Chapter 6

# Results

The results consist of plots from simulations described in Chapter 5 and a plot of the Lyapunov function described in Chapter 4.

## 6.1 PCA

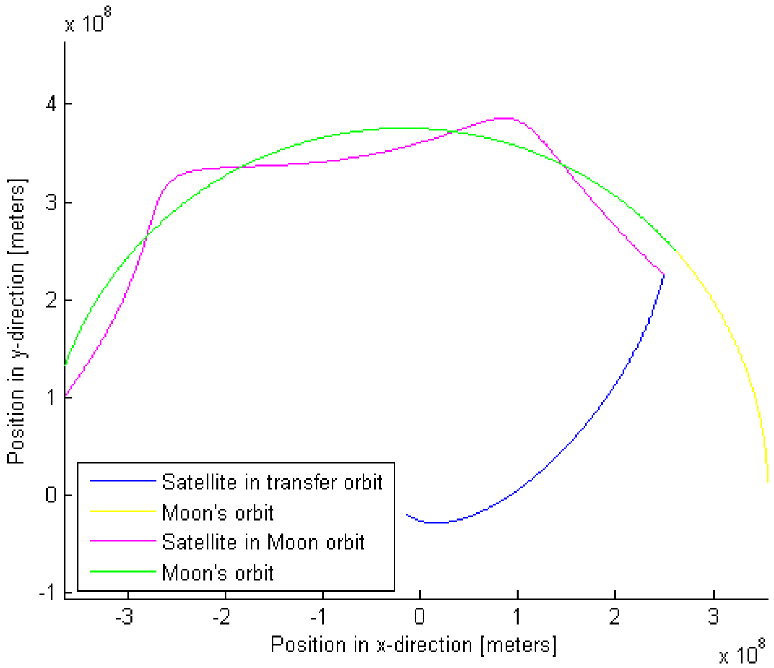
The simulations described in Section 5.2.2 are presented in Figure 6.1. The upper plot is the satellite's orbit relative to Earth and the lower plot is the satellite's orbit relative to the Moon.

The largest perturbing force comes from the Moon. It is therefore the one chosen in the simulations, see Section 5.2.3. The transfer orbit with and without the perturbing Moon is plotted in Figure 6.2.

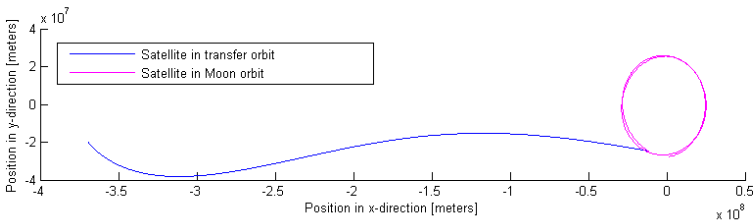
## 6.2 Control

### 6.2.1 PID controller

The PID controller's performance depends on the tuning parameters. As stated in Section 4.1, the system is nonlinear, so tuning has to be done by adjusting the parameters without any specific rules to go by. Parameters that give a satisfying result are presented in Table 6.1. Also a PD controller is suggested.




(a) The satellite in PCA trajectory seen from Earth

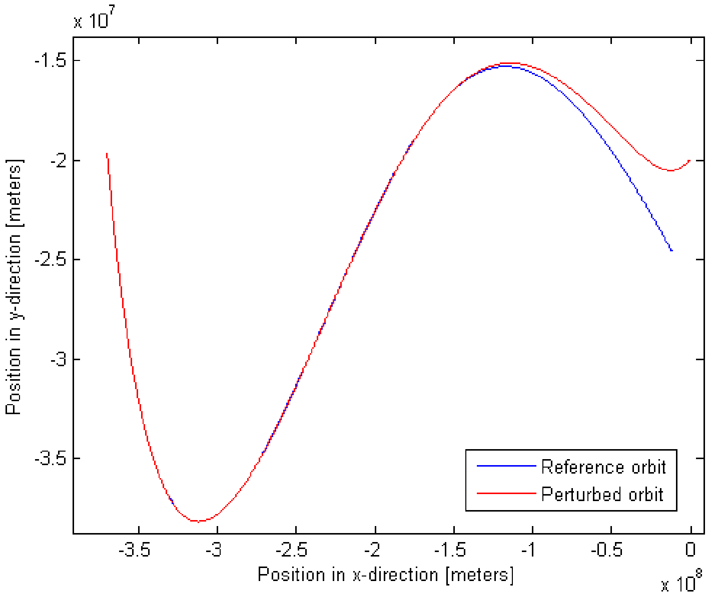



(b) The satellite in PCA trajectory seen from the Moon

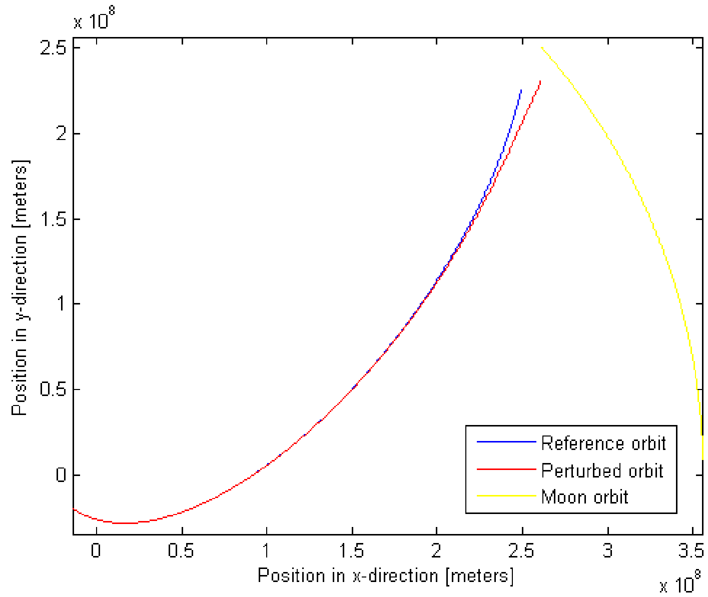
Figure 6.1: The satellite in the PCA trajectory

| *Controller* | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| PD | 1 | 0 | 10 |
| PID | 0.1 | 10 | 1 |

Table 6.1: Parameters of the simulated PID controllers




(a) The transfer orbit with and without the perturbing Moon, seen from the Moon




(b) The transfer orbit with and without the perturbing Moon, seen from Earth

Figure 6.2: The transfer orbit with and without the perturbing Moon

| *Controller* | $K_p$ | $T_i$ | $T_d$ | $N$ |
|---|---|---|---|---|
| PID | 1 | 5 | 5 | 10 |

Table 6.2: Parameters of the simulated limited PID controller

| *Controller* | $k_1$ | $k_2$ | $k_3$ |
|---|---|---|---|
| Nonlinear | 1 | 1 | 10 |

Table 6.3: Parameters of the simulated Nonlinear controller

Figure 6.3 and 6.4 show the position of the satellite and the errors, that is the difference between the reference and actual position and velocity for the PD controller respectively. It is the Euclidean norm of the x-, y- and z-components that is used for the plots. Figure 6.5 and 6.6 show the same for the PID controller. For all plots of the position of the satellite, the reference is barely visible as the plot of the actual satellite orbit covers it almost completely.

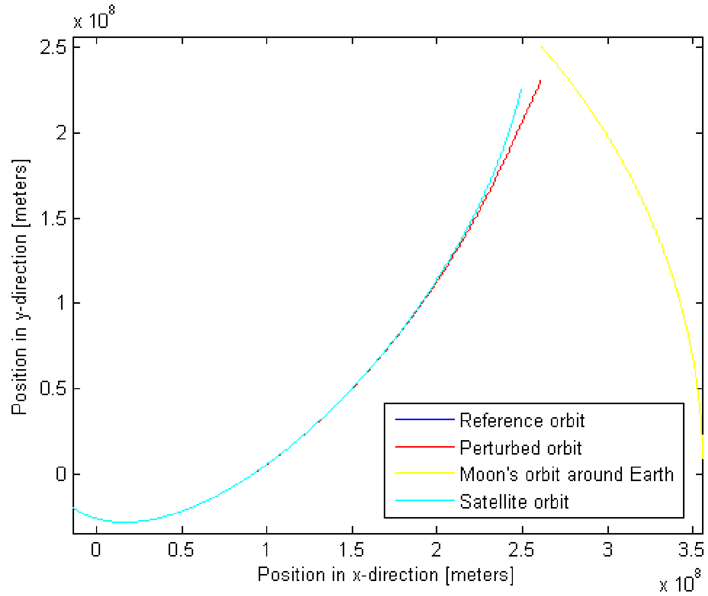
The same simulations are also done with a limited PID controller. The suggested set of values are given in Table 6.2. Here, $N$ is the denominator seen in Figure 4.3. The corresponding plots are shown in Figure 6.7 and 6.8.

### 6.2.2 Nonlinear controller

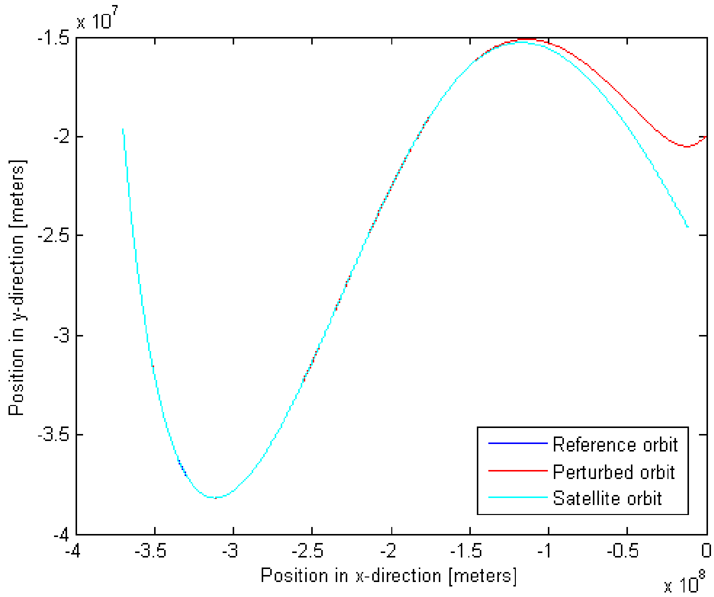
Different values of the nonlinear controller's parameters were tested. The values that give the best results are shown in Table 6.3. The corresponding plots are shown in Figure 6.9 and 6.10.

## 6.3 Stability

The chosen Lyapunov function for the nonlinear controller is plotted to see if it concurrs with the stability theory. The plots are shown in Figure 6.11.

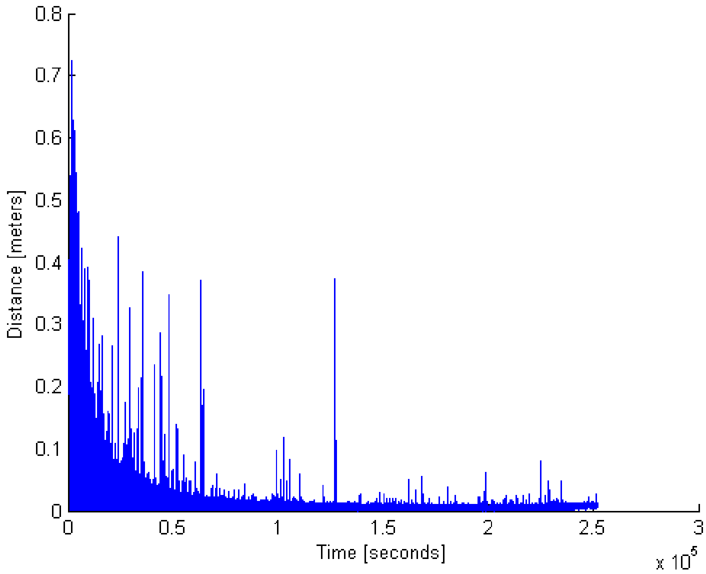



(a) The position seen from Earth

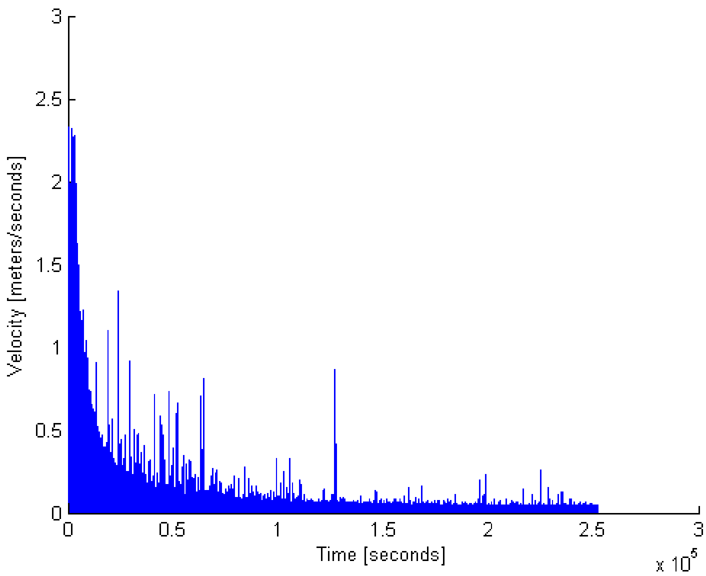



(b) The position seen from the Moon

Figure 6.3: Simulation results for the PD controller; positions

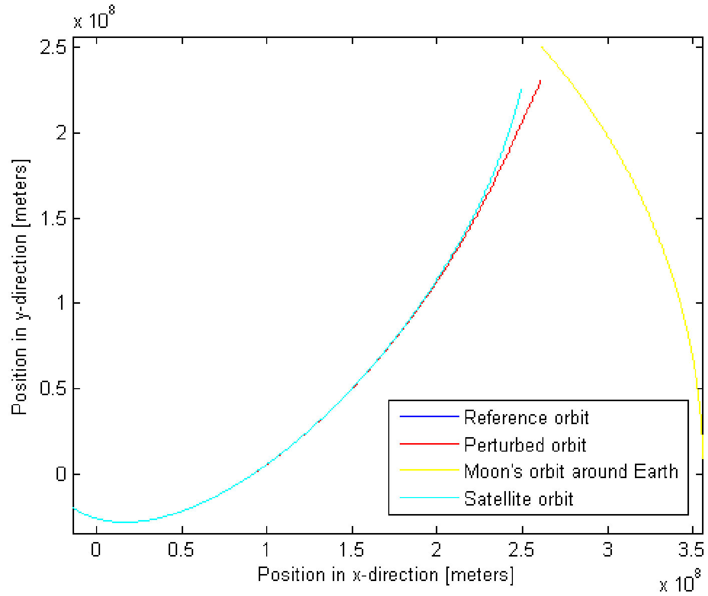



(a) The error in position

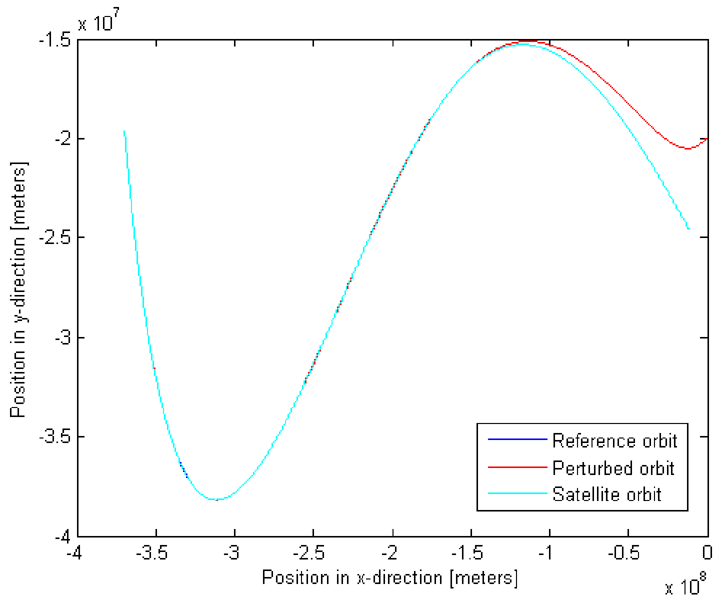



(b) The error in velocity

Figure 6.4: Simulation results for the PD controller; errors

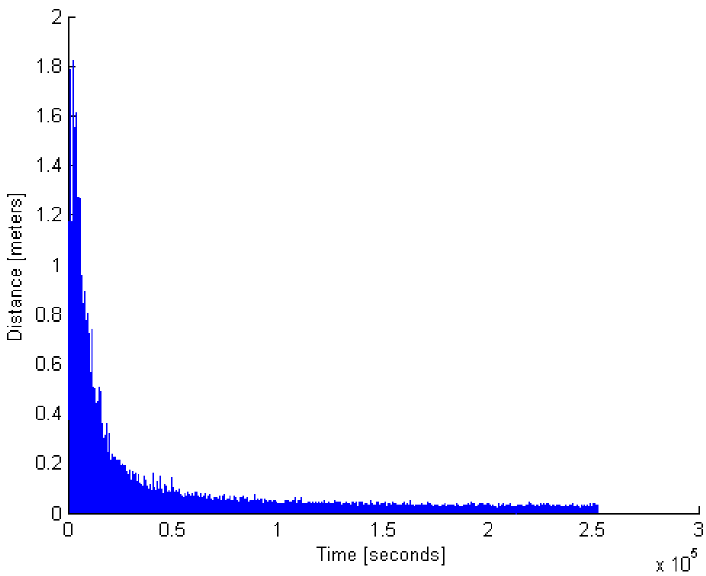



(a) The position seen from Earth

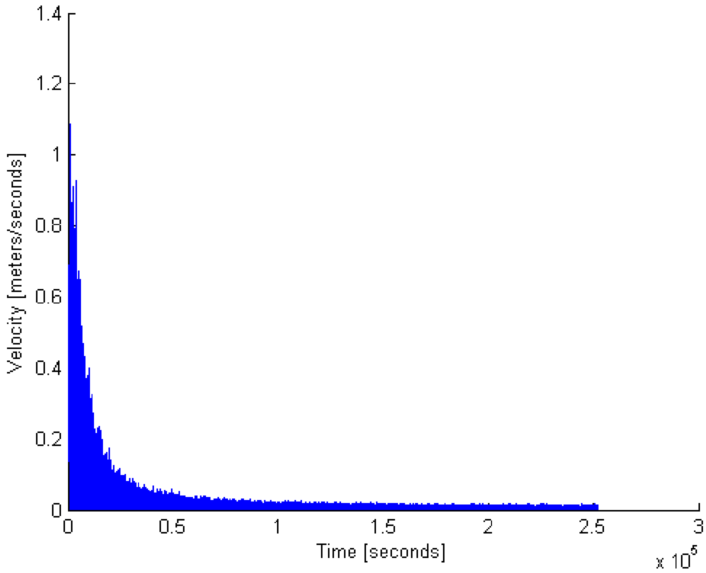



(b) The position seen from the Moon

Figure 6.5: Simulation results for the PID controller; positions

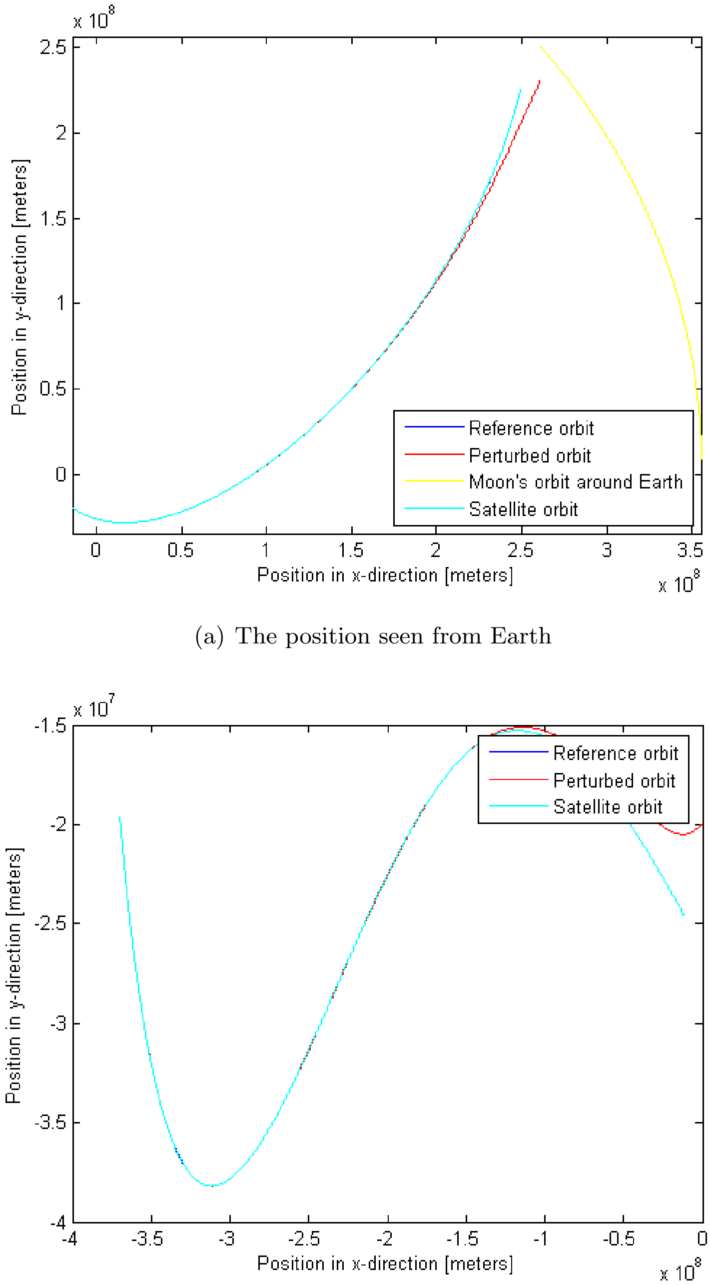



(a) The error in position




(b) The error in velocity

Figure 6.6: Simulation results for the PID controller; errors

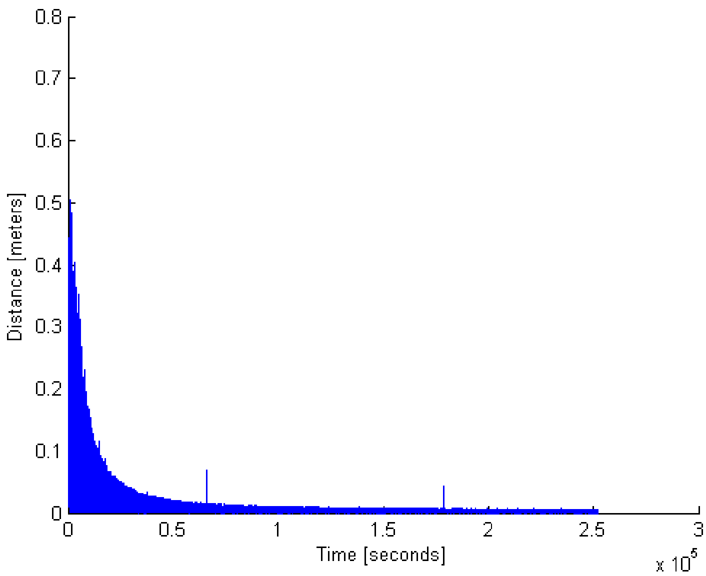



(a) The position seen from Earth

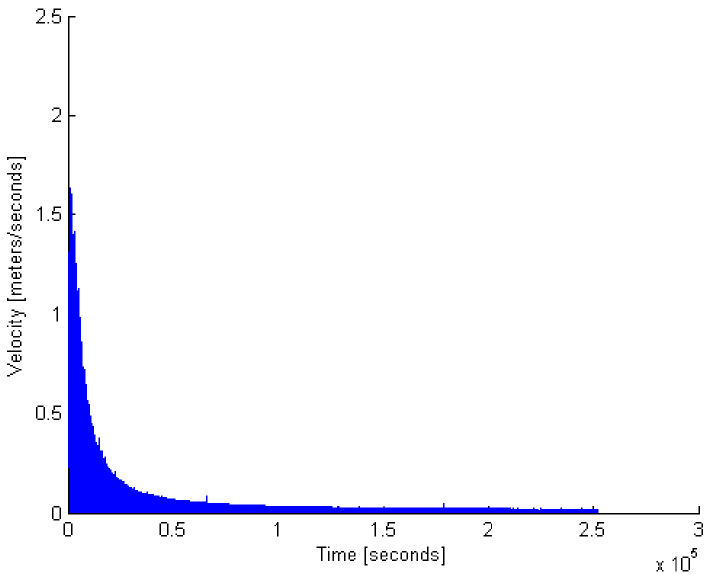


(b) The position seen from the Moon

Figure 6.7: Simulation results for the limited PID controller; positions

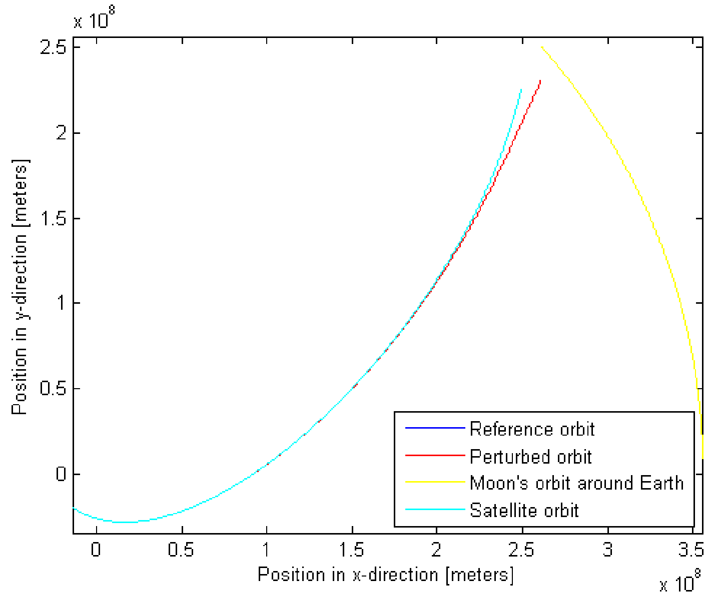



(a) The error in position

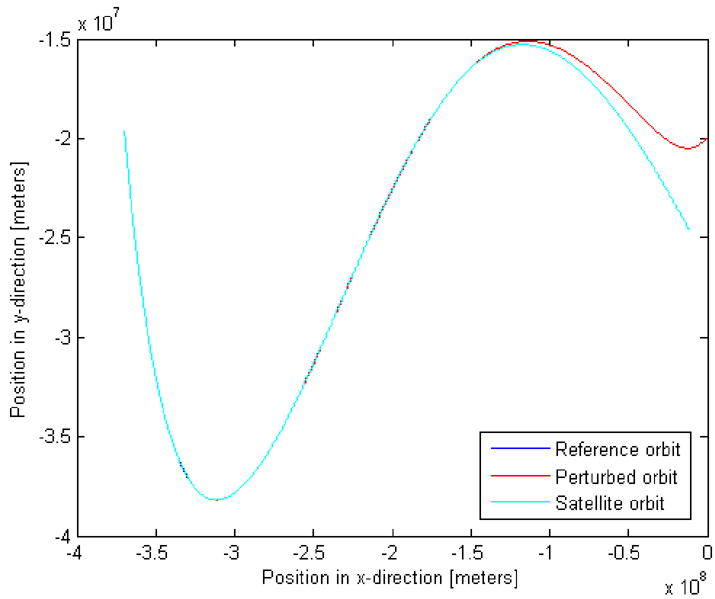



(b) The error in velocity

Figure 6.8: Simulation results for the limited PID controller; errors

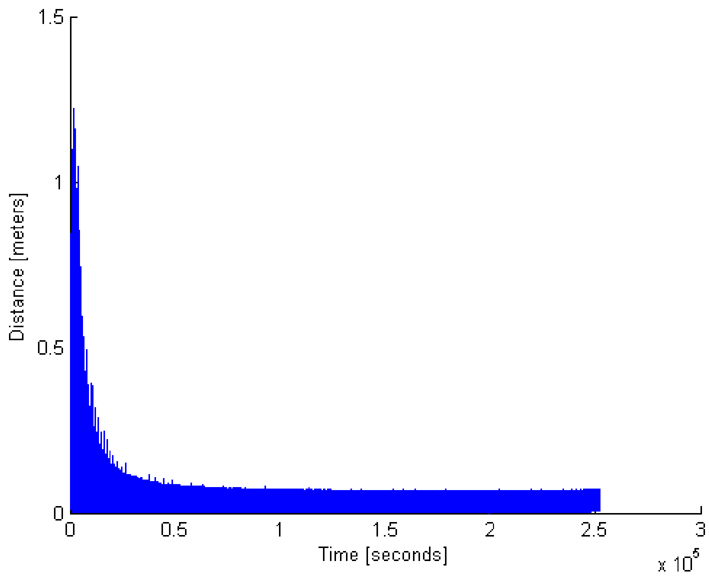



(a) The position seen from Earth

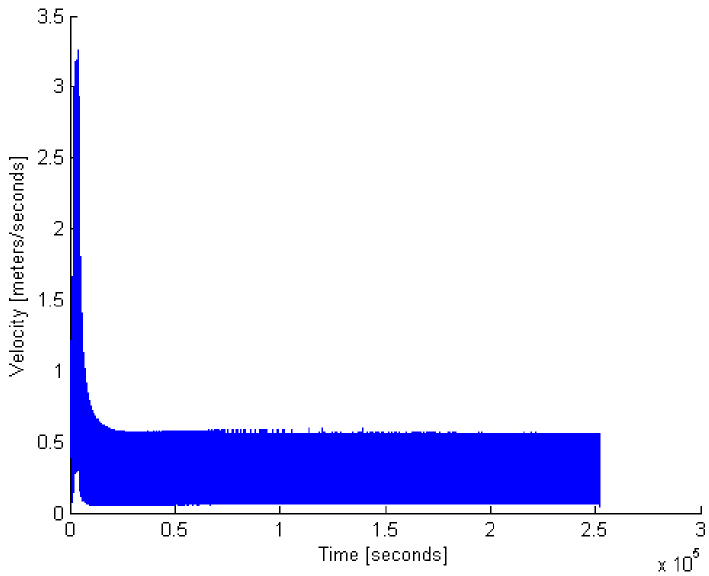



(b) The position seen from the Moon

Figure 6.9: Simulation results for the Nonlinear controller; positions

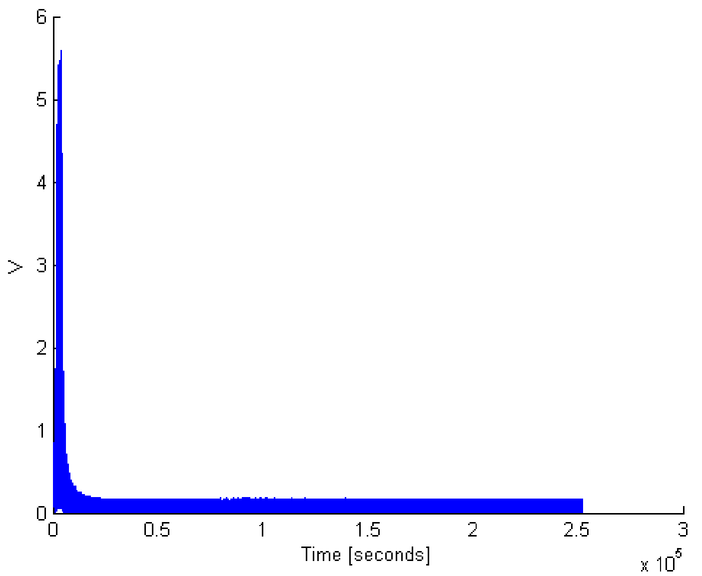



(a) The error in position

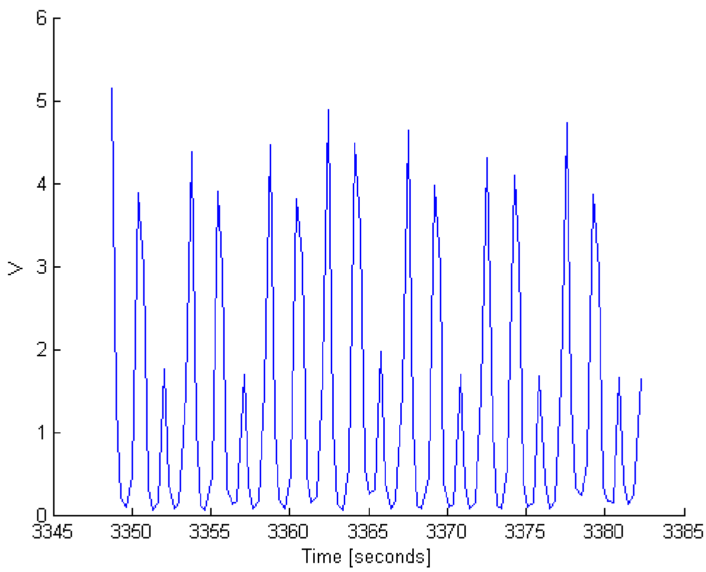



(b) The error in velocity

Figure 6.10: Simulation results for the Nonlinear controller; errors




(a) Lyapunov function for the nonlinear controller




(b) Lyapunov function for the nonlinear controller, close-up

Figure 6.11: Lyapunov function plots

## 6.4 Propulsion use

The controllers are both continuous. They therefore consume a lot of propellant. As can be seen from the plots, the satellite orbit oscilliates around the desired orbit before it gets closer to it. This is propellant consuming, and ideally the satellite orbit should be smoother.

# Chapter 7

# Discussion

## 7.1 Trajectories

This thesis describes three methods of getting the satellite to the Moon; the Hohmann transfer, the PCA and the BCT. The cheapest of these three is the BCT as it uses less fuel than the other two methods. On the other hand, it is also the most complicated of the three.

The chosen method to use for the simulations in this thesis, is the PCA as the main goal is to use a controller to make the satellite follow a desired trajectory. Therefore, the choice of trajectory does not really matter that much as the control theory will be the same for any trajectory.

Because of the choise of dynamic model, where celestial bodies move, a variation of the PCA method is used. The impulse from the transfer orbit to the orbit around the Moon had to be recalculated from the simulations. One could therefore argue that another trajecotry might have been better.

With the newly calculated impulses however, the simulations work well, and the satellite ends up orbiting the Moon in a circle as long as no perturbations are taken into consideration.

Many perturbations are described in Section 2.3, but only the perturbing Moon is used in the transfer trajectory in the simulations. This is because the others become insignificante next to it. Figure 6.2 in the results shows that the Moon very much makes the satellite go out of course. This is most easily seen, naturally enough, as the satellite gets closer to the Moon.

## 7.2 Control

In the later sections, different PID controllers and a nonlinear controller are applied to a simulated satellite to make it follow a desired trajectory despite perturbations.

The plots from the different PID controllers show that the results vary a lot depending on the tuning of the parameters and if a normal or a limited PID controller is used. The pure PD controller has many spikes in the plots of the error in position and velocity, see Figure 6.4. These spikes vanish when a PID or a limited PID is used.

The PID controller, when correctly tuned, seems from the plots to give a satisfactory result. It is however more difficult to say something about this in theory because of the many nonlinearities. In the plots it can be seen that the satellite oscillates around the desired orbit. Therefore stability can not be proven without further theoretical analysis.

The simulation of the system with the nonlinear controller gives a plot of a Lyapunov function, Figure 6.11, that shows that the system is not stable, although stability theory concludes that the system is stable and convergent. For the system to be stable, $\dot{V} \leq 0$. This plot of the Lyapunov function $V$ is promising in the sense that it is decreasing. But at the same time it is oscillitaing, so $\dot{V} \nleq 0$ and the system is not stable, although it performs quite well.

A possible cause to these contradictory results can be the effect of the non-linear perturbation in the system. When the perturbation is reconded for in equation (4.9), the equation is changed to

$$
\begin{aligned}
\Delta\dot{v} &= -Gm_2\frac{\vec{r}_E}{r_E^3} - Gm_4\frac{\vec{r}_{Sun}}{r_{Sun}^3} - (-Gm_2\frac{\vec{r}_{Eref}}{r_{Eref}^3} - Gm_4\frac{\vec{r}_{Sunref}}{r_{Sunref}^3}) + p + u \\
&= -Gm_2\left(\frac{\vec{r}_E}{r_E^3} - \frac{\vec{r}_{Eref}}{r_{Eref}^3}\right) - Gm_4\left(\frac{\vec{r}_{Sun}}{r_{Sun}^3} - \frac{\vec{r}_{Sunref}}{r_{Sunref}^3}\right) \\
&\quad -Gm_3\frac{\vec{r}_M}{r_M^3} + u. \qquad (7.1)
\end{aligned}
$$

Here, $p$ is the perturbation and $\vec{r}_M$ is the distance from the satellite to the Moon. With the same control input $u$ as in equation (4.18), the resulting $\dot{V}$ is

$$\dot{V} = -k_2 k_3 \Delta v^T \Delta v - k_2 G m_3 \Delta v^T \frac{\vec{r}_M}{r_M^3}. \tag{7.2}$$

It is easily seen that $\dot{V}$ can not be shown to be $\leq 0$ without constrains. This indicates that the system including the perturbation might not be stable.

It does however seem to be bounded, see Figure 6.11. Theory on how to prove boundedness can be found in (Khalil [2000]) chapter 9: Stability of Perturbed Systems. All the theorems require at least uniform asymptotic stability of the nominal system. The nominal system presented in this thesis, has only been proven stable and convergent. Hence, boundedness of the perturbed system can not be proven.

Another cause for the oscillations of the satellite might be numerical errors in the simulations. The numerical solver used for the simulations, ode45, is well tested and should not give large errors. But it could be that it is not as accurate as hoped for, and it might be the reason why the nonlinear controller can not be tuned to make the errors in position and velocity cancel each other when used in the Lyapunov function after a very large number of attempts.

## 7.3 Conclusion

There are many ways of getting a satellite to the Moon. A fairly simple method is the Patched Conic Approximation. It is not the cheapest trajectory, but it gives a good platform for testing trajectory controllers.

The description of the satellite and its surroundings can be described by including a number of celestial objects such as Earth, the Moon and the Sun. In the simulations in this thesis, the Sun and Earth is always included. The Moon is considered a perturbation in the satellite's transfer orbit.

The perturbation brings the satellite out of course relative to its reference trajectory which is calculated from the PCA method. A feedback loop to a controller can make the satellite follow the reference trajectory despite the perturbation. Two controllers are suggested and tested; a PID controller and a nonlinear controller.

The PID controller is a very common controller, and is therefore easier to realize. When applied to the uncontrolled system, it makes the satellite follow the desired trajectory at an accuracy of approximately one decimeter. The PID and limited PID controller perform better than the pure PD controller.

The nonlinear controller is derived from Lyapunov control theory and is likely to be more difficult to realize. It eliminates all the nonlinearities from the set of differential functions of the satellite and stabilizes it. However, from plots it can be seen that it is only the case for the nominell system.

From the results it can be seen that it is fully possible to use a controller to keep a satellite in orbit even though the satellite dynamics are poorly described. It will however make the satellite heavy as it will require large quantities of propellant. A combination could on the other hand be very useful. The dynamics of the satellite can be more accurately described in addition to a controller that takes care of the unforeseen perturbations.

## 7.4 Future work

The controllers could be tested on a satellite that runs in a Low Earth Orbit (LEO) to see if they will be able to correct smaller errors in the orbit that are caused from the other mentioned perturbations in this thesis. The parameters might have to be adjusted to the new situation, otherwise they should work without problems.

The controllers in this thesis are used on the PCA method. But they should in theory be applicable to any desired trajectory as long as an accurate reference trajectory is given. It would therefore be interesting to use the controllers on trajectories more likely to be used for the ESMO satellite than the PCA.

Another controller should also be found for the system, one that can both be proven stable in theory and be shown to be stable in simulations for the perturbed system. Asymptotic stability would be best.

More work can be done on the thrusters with respect to the amount of time they are used. At the moment, they are used continuously. A thruster with a constant magnitude thrust is more likely to be onboard a satellite, and can therefore naturally not be used continuously, but be turned on when the error in the position of the satellite crosses a set limit.

# References


Jens G Balchen, Trond Andresen, and Bjarne A. Foss. *Reguleringsteknikk*. Institutt for teknisk kybernetikk, NTNU, 2002.

R.R. Bate, D.D. Mueller, and J.E. White. *Fundamentals of Astrodynamics*. Dover Publications Inc., 1971.

Edward A. Belbruno and John P. Carrico. *Calculation of Weak Stability Boundary Ballistic Lunar Transfer Trajectories*. American Institute of Aeronautics and Astronautics Inc., 2000.

C.D. Brown. *Spacecraft Mission Design, Second Edition*. American Institute of Aeronautics and Astronautics Inc., 1998.

Chi-Tsong Chen. *Linear System Theory and Design, Third Edition*. Oxford University Press, 1999.

Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.

Thor I. Fossen. *Marine Control Systems*. Marine Cybernetics, 2002.

Øyvind Hegrenæs. *Attitude Control by means of explicit model predictive control, via multi-parametric qudratic programming*. NTNU, 2004.

Kristian Jerpetjøn. *Flight path simulations*. HiN, 2004.

Kristin Johansson. *Orbital Mechanics for the Micro-Satellite SSETI/ESMO*. NTNU, 2004.

Hassan K. Khalil. *Nonlinear Systems*. Prentice-Hall Inc., 2000.

W.S. Koon, Marsden, and Ross. *Low energy transfer to the Moon*. Kluwer Academic Publishers, 2001.

Aleksander L. Marthinussen. *Master thesis: Banemekanikk for SSETI prosjektet*. Høgskolen i Narvik, 2004.

MatLab. *Help*. The MathWorks Inc., 2004.

Colin R. McInnes. *Course in University of Glasgow: Spaceflight Dynamics 3*. University of Glasgow, 2004.

Oliver Montenbruck. *Satellite Orbits*. Springer-Verlag Berlin Heidelberg, 2000.

Bo J. Naasz. *Classical Element Feedback Control for Spacecraft Orbital Maneuvers*. Virginia Polytechnic Institute and State University, 2002.

S.V. Rao, B.P. Dakshayani, and T. Ravindra Babu. *Orbit Information Computation for Data Products Generation for Indian Remote Sensing Spacecraft*. Flight Dynamics Division, ISRO Satellite Centre,, 2002.

Jerry Jon Sellers. *Understanding Space*. McGraw-Hill, 1994.

L.F. Shampine, R.C. Allen Jr., and S. Preuss. *Fundamentals of Numerical Computing*. John Wiley and Sons Inc., 1997.

Marcel J. Sidi. *Spacecraft Dynamics and Control*. Cambridge University Press, 2000.

Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice-Hall Inc., 1991.

Kristian Svartveit. *Attitude determination of the NCUBE satellite*. NTNU, 2003.

# Appendix A

# Finding parameters for the PCA trajectory

```
function out = calc(r0,v0e,gamma0,lambda)



% Constants
mye = 3.986*10^14;  % m^3/s^2
mym = 4.9028*10^12; % m^3/s^2
rs = 66300000;      % m
rem = 384400000;    % m
vm = 1.023e3;       % m/s

% Initial values in circular orbit
v0c = sqrt(mye/r0)
E = (1/2)*v0c^2 - (mye/r0)


% Transfer ellipse values

E = (v0e^2/2) - (mye/r0)
h = r0*v0e*cos(gamma0)
a = - (mye/(2*E));
e = sqrt(1 + 2*E*h^2/(mye^2))


% In the point where the satellite hits the SOI of the Moon

r1 = sqrt(rem^2 + rs^2 - 2*rem*rs*cos(lambda))
v1 = sqrt(mye*((2/r1) - (1/a)))
phi1 = asin(rs*sin(lambda)/r1)
nu = acos(((r0/r1)*(1 + e) - 1)/e);
epsilon = acos((e + cos(nu))/(1 + e*cos(nu)));
t = (epsilon - e*sin(epsilon))/(sqrt(mye/a^3))
```

```
% In SOI

gamma1 = atan(e*sin(nu)/(1 + e*cos(nu)))
alpha = gamma1 - phi1;
v2 = sqrt(vm^2 + v1^2 - 2*vm*v1*cos(alpha))
sigma = asin((vm/v2)*cos(lambda) - (v1/v2)*cos(lambda + phi1 - gamma1))
r2 = rs
E = (v2^2/2) - (mym/r2)
h = r2*v2*sin(sigma);
a = - (mym/(2*E))
e = sqrt(1 + (2*E*h^2/(mym^2)))
rph = h^2/(mym*(1 + e))
vhs1 = 2*mym/r2;
vhs2 = (mym/abs(a));
vhs = sqrt((2*mym/r2) + (mym/abs(a)))
betta = pi - sigma - (gamma1 - phi1) - ( pi - (pi/2) - lambda)


% From hyperbola to ellipse in SOI

r3a = rph
r3p = 2000000
ael = 0.5*(r3a + r3p)
vhe = sqrt((2*mym/r3a) + (mym/abs(a)))
vel3 = sqrt(mym*((2/r3a) - (1/ael)))


% From ellipse to circle

rc = r3p
vc = sqrt(mym/rc)
vel4 = sqrt(mym*((2/rc) - (1/ael)))


% Required deltaVs

dv1 = v0e - v0c;
dv2 = v2 - v1;
dv3 = vel3 - vhe;
dv4 = vc - vel4;

out = [dv1 dv2 dv3 dv4 t];
```

# Appendix B

# MatLab simulations

## B.1 Differential functions

```
function xdot = fourfunction (t, x)

  xdot = zeros (24,1);
  m1=0;                   % K: mass of satellite
  m2=6e24;% m2=0; %       % K: mass of earth
  m3=7.35e22;%m3=0;%      % K: mass of moon
  m4=1.99e30;%m4=0;%      % K: mass of sun

  G=6.6433e-11;

  l12=norm(x(1:3)-x(7:9));
  l13=norm(x(1:3)-x(13:15));
  l14=norm(x(1:3)-x(19:21));
  l23=norm(x(7:9)-x(13:15));
  l24=norm(x(7:9)-x(19:21));
  l34=norm(x(13:15)-x(19:21));


  xdot(1) = x(4);
  xdot(2) = x(5);
  xdot(3) = x(6);
  xdot(4) = (G*m2*(x(7)-x(1)))/l12^3 +(G*m3*(x(13)-x(1)))/l23^3+(G*m4*(x(19)-x(1)))/l14^3;
  xdot(5) = (G*m2*(x(8)-x(2)))/l12^3 +(G*m3*(x(14)-x(2)))/l23^3+(G*m4*(x(20)-x(2)))/l14^3;
  xdot(6) = (G*m2*(x(9)-x(3)))/l12^3 +(G*m3*(x(15)-x(3)))/l23^3+(G*m4*(x(21)-x(3)))/l14^3;
  xdot(7) = x(10);
  xdot(8) = x(11);
  xdot(9) = x(12);
  xdot(10) = (G*m1*(x(1)-x(7)))/l12^3+(G*m3*(x(13)-x(7)))/l23^3+(G*m4*(x(19)-x(7)))/l24^3;
  xdot(11) = (G*m1*(x(2)-x(8)))/l12^3+(G*m3*(x(14)-x(8)))/l23^3+(G*m4*(x(20)-x(8)))/l24^3;
  xdot(12) = (G*m1*(x(3)-x(9)))/l12^3+(G*m3*(x(15)-x(9)))/l23^3+(G*m4*(x(21)-x(9)))/l24^3;
  xdot(13) = x(16);
  xdot(14) = x(17);
```

```
  xdot(15) = x(18);
  xdot(16) = (G*m1*(x(1)-x(13)))/l13^3+(G*m2*(x(7)-x(13)))/l23^3+(G*m4*(x(19)-x(13)))/l34^3;
  xdot(17) = (G*m1*(x(2)-x(14)))/l13^3+(G*m2*(x(8)-x(14)))/l23^3+(G*m4*(x(20)-x(14)))/l34^3;
  xdot(18) = (G*m1*(x(3)-x(15)))/l13^3+(G*m2*(x(9)-x(15)))/l23^3+(G*m4*(x(21)-x(15)))/l34^3;
  xdot(19) = x(22);
  xdot(20) = x(23);
  xdot(21) = x(24);
  xdot(22) = (G*m1*(x(1)-x(19)))/l14^3+(G*m2*(x(7)-x(19)))/l24^3+(G*m3*(x(13)-x(19)))/l34^3;
  xdot(23) = (G*m1*(x(2)-x(20)))/l14^3+(G*m2*(x(8)-x(20)))/l24^3+(G*m3*(x(14)-x(20)))/l34^3;
  xdot(24) = (G*m1*(x(3)-x(21)))/l14^3+(G*m2*(x(9)-x(21)))/l24^3+(G*m3*(x(15)-x(21)))/l34^3;
```

## B.2 Running PCA

```
% runPCA.m

% For transfer to the point where the deltaV should be added to make a circle
timer = 70;

% For transfer ellipse phased to reach the moving moon:
x0 = [ 1.5e11-1.38136e7 , -1.96261e7 , 0, 4578.9320 , 26457.1656 ,0 ,1.5e11 ,
 0 , 0, 0 , 29680 , 0,1.5e11+356000000,0,0,0,1092.9+29680,0,0,0,0,0,0,0];

t=[0:1:3600*24*(timer/24)];

tol = 1e-13;
NUM=[tol:tol:tol*24];%66]
options = odeset('RelTol',tol,'AbsTol',NUM);
[t,x] = ode45 ('fourfunction', t, x0, options);

figure;
plot(x(:,1)-x(:,13),x(:,2)-x(:,14),'b')
figure;
plot(x(:,1)-x(:,7),x(:,2)-x(:,8),'b')
hold on
plot(x(:,13)-x(:,7),x(:,14)-x(:,8),'g')
axis equal;

% After transfer ellipse:
b = size(t);
c = b(1);
a = x(c,:);
save initAtMoon a


% For circle
timer = 200;

load initAtMoon
findingV;
x0 = [a(1),a(2),a(3),a(4)+dvx,a(5)+dvy,a(6),a(7),a(8),a(9),a(10),a(11),
```

```
a(12),a(13),a(14),a(15),a(16),a(17),a(18),a(19),a(20),a(21),a(22),a(23),a(24)];

t=[0:1:3600*24*(timer/24)];

tol = 1e-13;
NUM=[tol:tol:tol*24];%66]
options = odeset('RelTol',tol,'AbsTol',NUM);
[t,x] = ode45 ('fourfunction', t, x0, options);

figure(1);
hold on;
plot(x(:,1)-x(:,13),x(:,2)-x(:,14),'r')
figure(2);
hold on;
plot(x(:,1)-x(:,7),x(:,2)-x(:,8),'r')
plot(x(:,13)-x(:,7),x(:,14)-x(:,8),'g')
axis equal;
```

## B.3 Finding the velocities

### B.3.1 Finding the new velocities of the Moon orbit

```
% findingV.m

% Finding circle velocity in x and y from transfer ellipse

xSat = a(1);
ySat = a(2);
xMoon = a(13);
yMoon = a(14);
xdif = (xMoon - xSat);
ydif = (yMoon - ySat);

r = sqrt(xdif^2 + ydif^2);

G = 6.6433e-11;
mMoon = 7.35e22;

vCirc = sqrt(G*mMoon/r);

alpha = acos(xdif/r);
betta = (pi/2) - alpha;

vxSat = vCirc*cos(betta);
vySat = vCirc*sin(betta);
```

### B.3.2 Finding $\Delta V_{2x}$ and $\Delta V_{2y}$

```
b = size(t);
c = b(1);
a = x(c,:);

findingV;               % Gives vxSat and vySat.

vxWanted = a(16)-vxSat; % Velocity of the Moon in x-direction and the wanted velocity of
                          the satellite relative to the Moon in x-direction.
vyWanted = a(17)+vySat; % Velocity of the Moon in y-direction and the wanted velocity of
                          the satellite relative to the Moon in y-direction.

dvx = vxWanted - a(4);  % The velocity wanted minus the former velocity of the satellite
                          giving the difference that should be applied, x-direction.
dvy = vyWanted - a(5);  % The velocity wanted minus the former velocity of the satellite
                          giving the difference that should be applied, y-direction.
```

# Appendix C

# Simulink diagrams

## C.1 Transfer with PID control

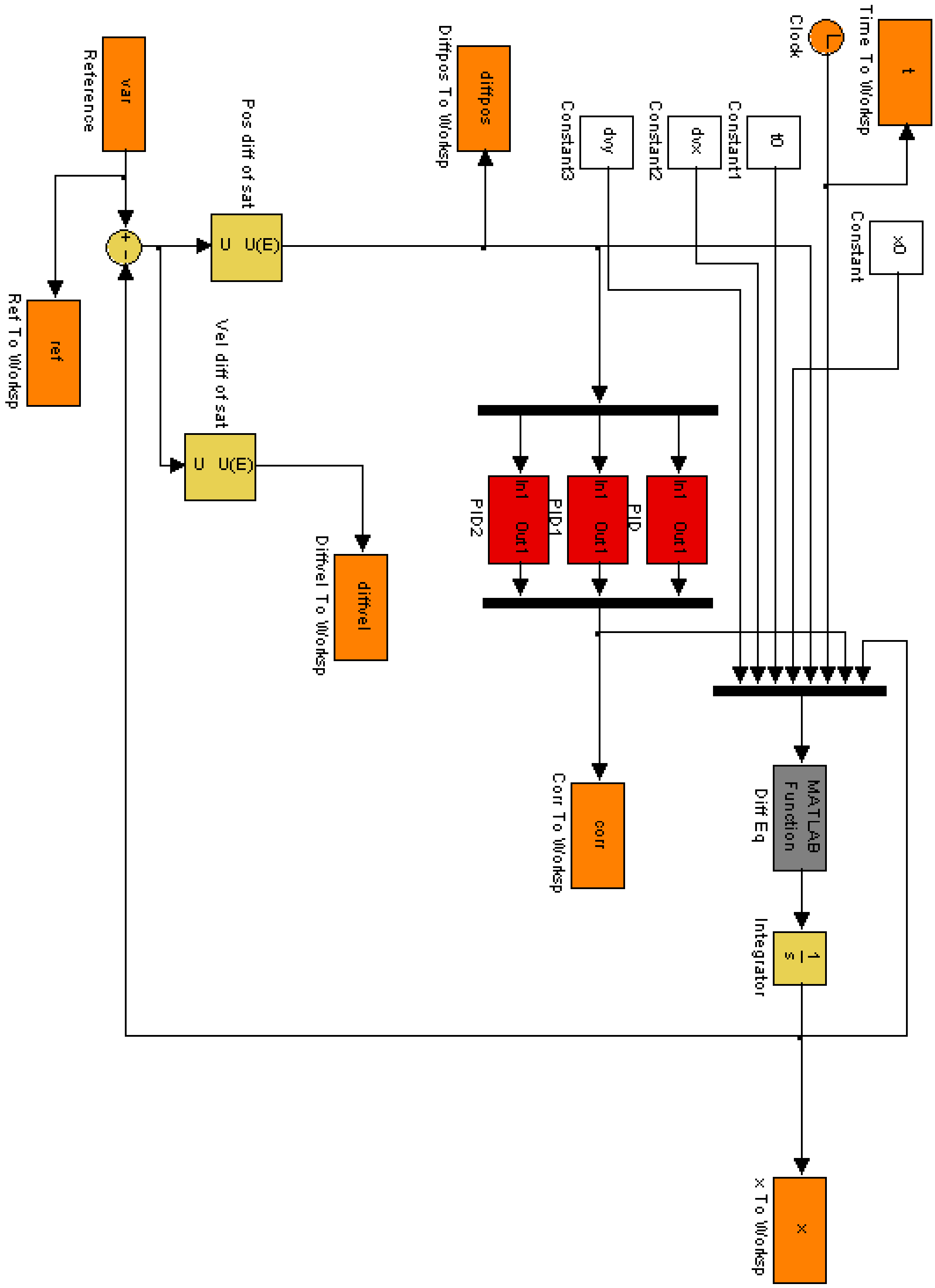



Figure C.1: Simulink diagram of PID controlled transfer

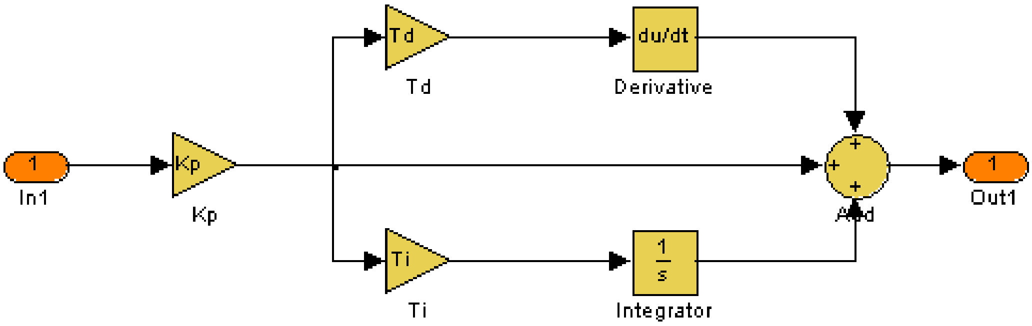



Figure C.2: Simulink diagram of the PID controller

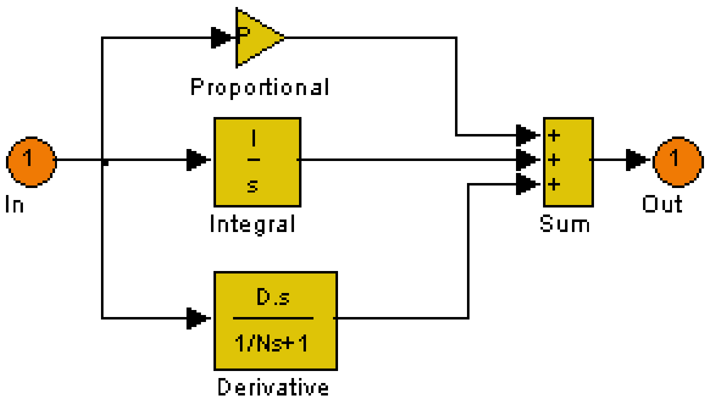



Figure C.3: Simulink diagram of the limited PID controller

## C.2 Transfer with Nonlinear control

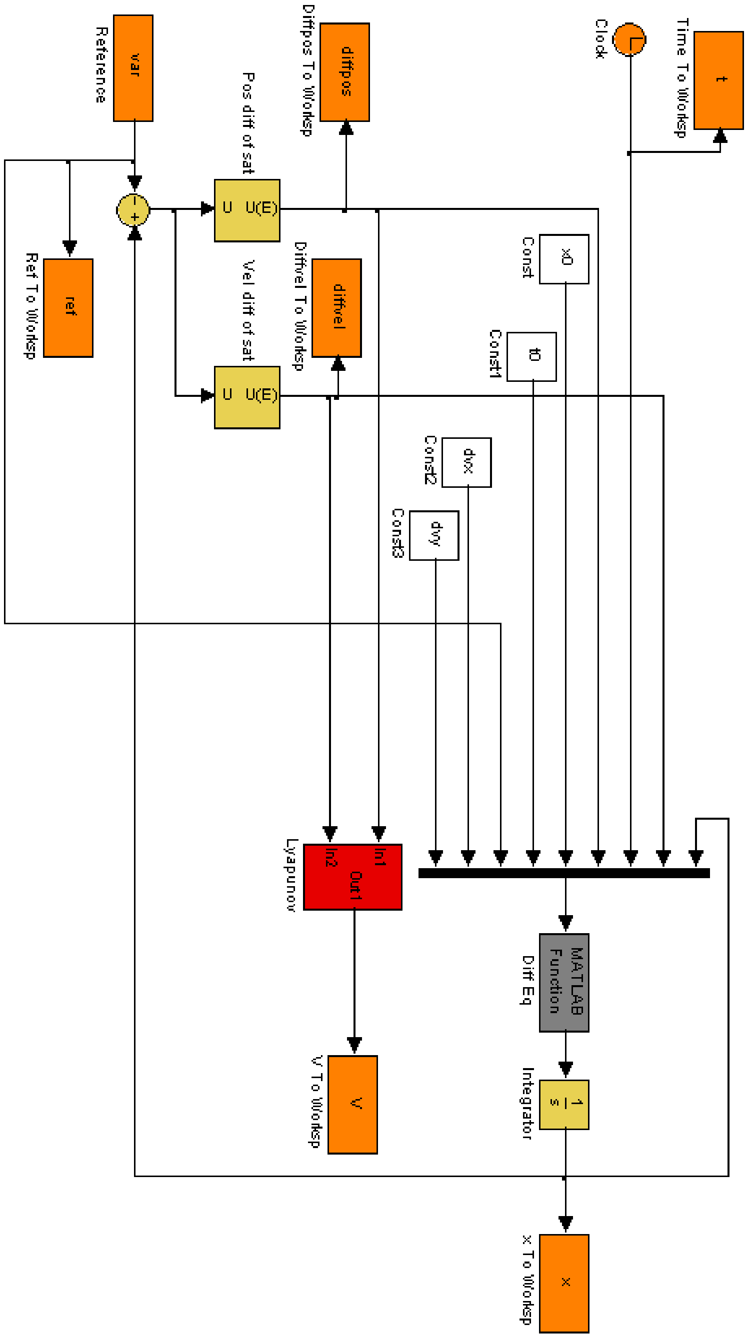



Figure C.4: Simulink diagram of transfer with nonlinear controller

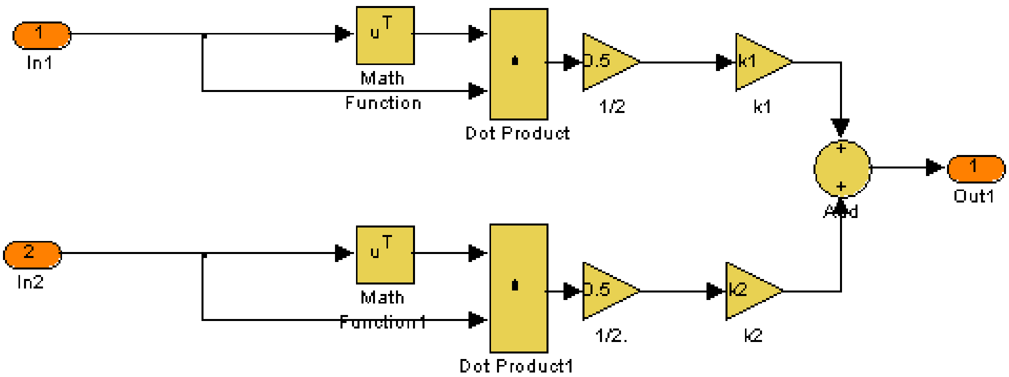



Figure C.5: Simulink diagram of the Lyapunov function

# Appendix D

# MatLab code for Simulink boxes

## D.1 Non-perturbed differential functions

```
function xdot = fourfunction(input)

  x0 = input(26:49);
  t0 = input(50);
  dvx = input(51);
  dvy = input(52);

  t = input(25); % from clock
  xdot = zeros (24,1);

  if (t <= (t0 + 1))
      x = x0; % initial states
  else
      x = input(1:24); % state; positions and velocities
  end

  If going into Moon orbit: (comment out if not)
  if (t <= (t0 + 10))
      ux = dvx/10;
      uy = dvy/10;
  else
      ux = 0;
      uy = 0;
  end



  m1=0;                % mass of satellite
  m2=6e24;% m2=0;      % mass of earth
```

```
  m3=7.35e22;%m3=0;    % mass of moon
  m4=1.99e30;%m4=0;    % mass of sun

  G=6.6433e-11;

  l12=norm(x(1:3)-x(7:9));
  l13=norm(x(1:3)-x(13:15));
  l14=norm(x(1:3)-x(19:21));
  l23=norm(x(7:9)-x(13:15));
  l24=norm(x(7:9)-x(19:21));
  l34=norm(x(13:15)-x(19:21));



  xdot(1) = x(4);
  xdot(2) = x(5);
  xdot(3) = x(6);
% If for Earth and transfer orbit:
%   xdot(4) = (G*m2*(x(7)-x(1)))/l12^3+(G*m4*(x(19)-x(1)))/l14^3;
%   xdot(5) = (G*m2*(x(8)-x(2)))/l12^3+(G*m4*(x(20)-x(2)))/l14^3;
%   xdot(6) = (G*m2*(x(9)-x(3)))/l12^3+(G*m4*(x(21)-x(3)))/l14^3;
% If for Moon orbit:
xdot(4) = (G*m2*(x(7)-x(1)))/l12^3+(G*m4*(x(19)-x(1)))/l14^3+(G*m3*(x(13)-x(1)))/l13^3 + ux;
xdot(5) = (G*m2*(x(8)-x(2)))/l12^3+(G*m4*(x(20)-x(2)))/l14^3+(G*m3*(x(14)-x(2)))/l13^3 + uy;
xdot(6) = (G*m2*(x(9)-x(3)))/l12^3+(G*m4*(x(21)-x(3)))/l14^3+(G*m3*(x(15)-x(3)))/l13^3;

  xdot(7) = x(10);
  xdot(8) = x(11);
  xdot(9) = x(12);
  xdot(10) = (G*m1*(x(1)-x(7)))/l12^3+(G*m3*(x(13)-x(7)))/l23^3+(G*m4*(x(19)-x(7)))/l24^3;
  xdot(11) = (G*m1*(x(2)-x(8)))/l12^3+(G*m3*(x(14)-x(8)))/l23^3+(G*m4*(x(20)-x(8)))/l24^3;
  xdot(12) = (G*m1*(x(3)-x(9)))/l12^3+(G*m3*(x(15)-x(9)))/l23^3+(G*m4*(x(21)-x(9)))/l24^3;
  xdot(13) = x(16);
  xdot(14) = x(17);
  xdot(15) = x(18);
  xdot(16) = (G*m1*(x(1)-x(13)))/l13^3+(G*m2*(x(7)-x(13)))/l23^3+(G*m4*(x(19)-x(13)))/l34^3;
  xdot(17) = (G*m1*(x(2)-x(14)))/l13^3+(G*m2*(x(8)-x(14)))/l23^3+(G*m4*(x(20)-x(14)))/l34^3;
  xdot(18) = (G*m1*(x(3)-x(15)))/l13^3+(G*m2*(x(9)-x(15)))/l23^3+(G*m4*(x(21)-x(15)))/l34^3;
  xdot(19) = x(22);
  xdot(20) = x(23);
  xdot(21) = x(24);
  xdot(22) = (G*m1*(x(1)-x(19)))/l14^3+(G*m2*(x(7)-x(19)))/l24^3+(G*m3*(x(13)-x(19)))/l34^3;
  xdot(23) = (G*m1*(x(2)-x(20)))/l14^3+(G*m2*(x(8)-x(20)))/l24^3+(G*m3*(x(14)-x(20)))/l34^3;
  xdot(24) = (G*m1*(x(3)-x(21)))/l14^3+(G*m2*(x(9)-x(21)))/l24^3+(G*m3*(x(15)-x(21)))/l34^3;
```

## D.2 Perturbed and controlled differential functions for PID controller

```
function xdot = fourfunctionControlPID(input)
```

```
x0 = input(32:55);
t0 = input(56);
dvx = input(57);
dvy = input(58);

t = input(28); % from clock
xdot = zeros (24,1);

if (t <= (t0 + 1))
    x = x0; % initial states
    incorrx = 0;
    incorry = 0;
    incorrz = 0;
    diffx = 0;
    diffy = 0;
    diffz = 0;
else
    x = input(1:24); % state; positions and velocities
    incorrx = input(25);
    incorry = input(26);
    incorrz = input(27);
    diffx = input(29);
    diffy = input(30);
    diffz = input(31);
end

corrx = incorrx;
corry = incorry;
corrz = incorrz;

% If going into Moon orbit: (comment out if not)
%if (t <= (t0 + 10))
%    ux = dvx/10;
%    uy = dvy/10;
%else
%    ux = 0;
%    uy = 0;
%end

% If Earth parking or transfer orbit: (comment out if not)
ux = 0;
uy = 0;


m1=0;               % mass of satellite
m2=6e24;            % mass of earth
m3=7.35e22;         % mass of moon
m4=1.99e30;         % mass of sun

G=6.6433e-11;

l12=norm(x(1:3)-x(7:9));
l13=norm(x(1:3)-x(13:15));
```

```
  l14=norm(x(1:3)-x(19:21));
  l23=norm(x(7:9)-x(13:15));
  l24=norm(x(7:9)-x(19:21));
  l34=norm(x(13:15)-x(19:21));



  xdot(1) = x(4);
  xdot(2) = x(5);
  xdot(3) = x(6);
  xdot(4) = (G*m2*(x(7)-x(1)))/l12^3+(G*m4*(x(19)-x(1)))/l14^3
           +(G*m3*(x(13)-x(1)))/l13^3 + ux + corrx;
  xdot(5) = (G*m2*(x(8)-x(2)))/l12^3+(G*m4*(x(20)-x(2)))/l14^3
           +(G*m3*(x(14)-x(2)))/l13^3 + uy + corry;
  xdot(6) = (G*m2*(x(9)-x(3)))/l12^3+(G*m4*(x(21)-x(3)))/l14^3
           +(G*m3*(x(15)-x(3)))/l13^3 + corrz;
  xdot(7) = x(10);
  xdot(8) = x(11);
  xdot(9) = x(12);
  xdot(10) = (G*m1*(x(1)-x(7)))/l12^3+(G*m3*(x(13)-x(7)))/l23^3+(G*m4*(x(19)-x(7)))/l24^3;
  xdot(11) = (G*m1*(x(2)-x(8)))/l12^3+(G*m3*(x(14)-x(8)))/l23^3+(G*m4*(x(20)-x(8)))/l24^3;
  xdot(12) = (G*m1*(x(3)-x(9)))/l12^3+(G*m3*(x(15)-x(9)))/l23^3+(G*m4*(x(21)-x(9)))/l24^3;
  xdot(13) = x(16);
  xdot(14) = x(17);
  xdot(15) = x(18);
  xdot(16) = (G*m1*(x(1)-x(13)))/l13^3+(G*m2*(x(7)-x(13)))/l23^3+(G*m4*(x(19)-x(13)))/l34^3;
  xdot(17) = (G*m1*(x(2)-x(14)))/l13^3+(G*m2*(x(8)-x(14)))/l23^3+(G*m4*(x(20)-x(14)))/l34^3;
  xdot(18) = (G*m1*(x(3)-x(15)))/l13^3+(G*m2*(x(9)-x(15)))/l23^3+(G*m4*(x(21)-x(15)))/l34^3;
  xdot(19) = x(22);
  xdot(20) = x(23);
  xdot(21) = x(24);
  xdot(22) = (G*m1*(x(1)-x(19)))/l14^3+(G*m2*(x(7)-x(19)))/l24^3+(G*m3*(x(13)-x(19)))/l34^3;
  xdot(23) = (G*m1*(x(2)-x(20)))/l14^3+(G*m2*(x(8)-x(20)))/l24^3+(G*m3*(x(14)-x(20)))/l34^3;
  xdot(24) = (G*m1*(x(3)-x(21)))/l14^3+(G*m2*(x(9)-x(21)))/l24^3+(G*m3*(x(15)-x(21)))/l34^3;
```

## D.3 Perturbed and controlled differential functions for Nonlinear controller

```
function xdot = fourfunctionLyapunovNew(input)

  x0 = input(32:55);
  t0 = input(56);
  dvx = input(57);
  dvy = input(58);

  t = input(28); % from clock
  xdot = zeros (24,1);
```

```
if (t <= (t0 + 1))
    x = x0; % initial states
    diffvelx = 0;
    diffvely = 0;
    diffvelz = 0;
    diffposx = 0;
    diffposy = 0;
    diffposz = 0;
else
    x = input(1:24); % state; positions and velocities
    diffvelx = input(25);
    diffvely = input(26);
    diffvelz = input(27);
    diffposx = input(29);
    diffposy = input(30);
    diffposz = input(31);
end

% If going into Moon orbit: (comment out if not)
%if (t <= (t0 + 10))
%    ux = dvx/10;
%    uy = dvy/10;
%else
%    ux = 0;
%    uy = 0;
%end

% If Earth parking or transfer orbit: (comment out if not)
ux = 0;
uy = 0;


m1=0;                % mass of satellite
m2=6e24;% m2=0;      % mass of earth
m3=7.35e22;%m3=0;    % mass of moon
m4=1.99e30;%m4=0;    % mass of sun

G=6.6433e-11;

l12=norm(x(1:3)-x(7:9));
l13=norm(x(1:3)-x(13:15));
l14=norm(x(1:3)-x(19:21));
l23=norm(x(7:9)-x(13:15));
l24=norm(x(7:9)-x(19:21));
l34=norm(x(13:15)-x(19:21));



%--Lyapunov control part------------
corrx = 0;
corry = 0;
corrz = 0;

if (t >= (t0 + 1))
```

```
% A quite good set of k's:
k1 = 1;
k2 = 1;
k3 = 10;

rSatx = input(57);
rSaty = input(58);
rSatz = input(59);
rEx = input(63);
rEy = input(64);
rEz = input(65);
rSx = input(75);
rSy = input(76);
rSz = input(77);

rSat = input(57:59);
rE = input(63:65);
rS = input(75:77);

r12 = norm(rSat-rE);
r14 = norm(rSat-rS);

ulinx = G*m2*(((x(1)-x(7))/l12^3)-((rSatx-rEx)/r12^3))
        + G*m4*(((x(1)-x(19))/l14^3)-((rSatx-rSx)/r14^3));
uliny = G*m2*(((x(2)-x(8))/l12^3)-((rSaty-rEy)/r12^3))
        + G*m4*(((x(2)-x(20))/l14^3)-((rSaty-rSy)/r14^3));
ulinz = G*m2*(((x(3)-x(9))/l12^3)-((rSatz-rEz)/r12^3))
        + G*m4*(((x(3)-x(21))/l14^3)-((rSatz-rSz)/r14^3));

corrx = - k3*diffvelx - (k1/k2)*diffposx + ulinx;
corry = - k3*diffvely - (k1/k2)*diffposy + uliny;
corrz = - k3*diffvelz - (k1/k2)*diffposz + ulinz;

end
%--end corr part-----------------------


xdot(1) = x(4);
xdot(2) = x(5);
xdot(3) = x(6);
xdot(4) = - (G*m2*(x(1)-x(7)))/l12^3 -(G*m4*(x(1)-x(19)))/l14^3
 -(G*m3*(x(1)-x(13)))/l13^3 + ux + corrx;
xdot(5) = - (G*m2*(x(2)-x(8)))/l12^3 -(G*m4*(x(2)-x(20)))/l14^3
 -(G*m3*(x(2)-x(14)))/l13^3 + uy + corry;
xdot(6) = - (G*m2*(x(3)-x(9)))/l12^3-(G*m4*(x(3)-x(21)))/l14^3
 -(G*m3*(x(3)-x(15)))/l13^3 + corrz;
xdot(7) = x(10);
xdot(8) = x(11);
xdot(9) = x(12);
xdot(10) = -(G*m1*(x(7)-x(1)))/l12^3-(G*m3*(x(7)-x(13)))/l23^3-(G*m4*(x(7)-x(19)))/l24^3;
xdot(11) = -(G*m1*(x(8)-x(2)))/l12^3-(G*m3*(x(8)-x(14)))/l23^3-(G*m4*(x(8)-x(20)))/l24^3;
xdot(12) = -(G*m1*(x(9)-x(3)))/l12^3-(G*m3*(x(9)-x(15)))/l23^3-(G*m4*(x(9)-x(21)))/l24^3;
xdot(13) = x(16);
```

```
xdot(14) = x(17);
xdot(15) = x(18);
xdot(16) = (G*m1*(x(1)-x(13)))/l13^3+(G*m2*(x(7)-x(13)))/l23^3+(G*m4*(x(19)-x(13)))/l34^3;
xdot(17) = (G*m1*(x(2)-x(14)))/l13^3+(G*m2*(x(8)-x(14)))/l23^3+(G*m4*(x(20)-x(14)))/l34^3;
xdot(18) = (G*m1*(x(3)-x(15)))/l13^3+(G*m2*(x(9)-x(15)))/l23^3+(G*m4*(x(21)-x(15)))/l34^3;
xdot(19) = x(22);
xdot(20) = x(23);
xdot(21) = x(24);
xdot(22) = (G*m1*(x(1)-x(19)))/l14^3+(G*m2*(x(7)-x(19)))/l24^3+(G*m3*(x(13)-x(19)))/l34^3;
xdot(23) = (G*m1*(x(2)-x(20)))/l14^3+(G*m2*(x(8)-x(20)))/l24^3+(G*m3*(x(14)-x(20)))/l34^3;
xdot(24) = (G*m1*(x(3)-x(21)))/l14^3+(G*m2*(x(9)-x(21)))/l24^3+(G*m3*(x(15)-x(21)))/l34^3;
```