UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

H_∞ Orbital Control of a Space Vehicle on a Low Earth Orbit (Versão Corrigida Após Defesa)

Pedro Miguel Mariano Dente

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(ciclo de estudos integrado)

Orientador: Prof. Doutor Kouamana Bousson

Covilhã, janeiro de 2020

Acknowledgements

I would like to thank my supervisor, Prof. Doutor Kouamana Bousson, for accepting this challenge, for his patience and for his devotion in helping me.

I would also like to give a special appreciation to my parents, Álvaro and Ana, for all the support and motivation in pursuing an academic career and to my sister, Inês, for all the joy she brings in my life.

Finally, a big shout out to my friends for their kindness.

Resumo

Uma Órbita Terrestre Baixa (LEO) é a trajetória de um veículo espacial tal como um satélite, míssil ou vaivém espacial relativa à Terra com uma altitude de 1500 km ou menos. As órbitas Keplerianas são apenas caracterizadas pelas forças gravitacionais entre corpos, no entanto, em aplicações reais há muitas perturbações que alteram a trajetória inicial ou a "órbita matemática ideal" do veículo espacial.

As perturbações analisadas nesta dissertação foram a força de resistência aerodinâmica e a não esfericidade da Terra (J_2). É necessário controlar esta trajetória de maneira a ajustar os elementos orbitais para que o veículo espacial possa operar na vizinhança da órbita inicial e executar a sua missão adequadamente.

Há vários tipos de controladores para estes veículos espaciais como o Regulador Quadrático Linear (LQR), o Controlo por Modo Deslizante (SMC), Desigualdades Matriciais Lineares (LMI) e o Controlador H-infinito (H_∞). É essencial encontrar o melhor tipo de controlador para cada missão específica. Tendo em conta as limitações intrínsecas dos métodos acima mencionados, o Controlador H_∞ foi julgado o mais adequado neste caso, dentro de certas condições.

O principal objetivo desta dissertação é o desenvolvimento de um controlador H_∞ robusto que consegue alterar a trajetória de um veículo espacial com perturbações, de maneira a fazê-lo seguir uma trajetória Kepleriana com as mesmas condições iniciais.

Palavras-chave

Veículo Espacial; Órbita Terrestre Baixa; Força de Resistência Aerodinâmica; J_2; Controlador H_∞

Abstract

A Low Earth Orbit (LEO) is the trajectory of a space vehicle such as a satellite, missile or space shuttle relative to the Earth with an altitude of 1,500 km or less. Keplerian orbits only take in account the gravitational forces between bodies, however, in real life applications there are a lot of perturbations that alter the initial trajectory or the "mathematical ideal orbit" of the space vehicle.

The perturbations analyzed in this dissertation were the atmospheric drag and the oblateness of the Earth (J_2). It is necessary to control this trajectory in order to adjust the orbital elements so that the space vehicle can operate in the surroundings of the initial orbit and perform its mission adequately.

There are many types of controllers for these space vehicles like the Linear-Quadratic Regulator (LQR), the Sliding Mode Control (SMC), the Linear Matrix Inequality (LMI) and the H-infinity (H_∞) Controller. It is essential to find the best type of controller for each specific mission. Considering the inherent limitations of the methods mentioned above, the H_∞ Controller was judged to be the most suitable for this application under certain conditions.

The focus of this dissertation is the development of an H_∞ robust controller that can successfully alter the trajectory of a space vehicle with perturbations, so that it follows a referenced Keplerian trajectory with the same initial conditions.

Keywords

Space Vehicle; Low Earth Orbit; Atmospheric Drag; J_2; H_∞ Controller

Contents

List of Figures

List of Tables

List of Acronyms

CSA	Canadian Space Agency
CMP	Control and Monitoring Processor
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
ESA	European Space Agency
GEO	Geosynchronous Equatorial Orbit
GMAT	General Mission Analysis Tool
GPS	Global Positioning System
ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency
LEO	Low Earth Orbit
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
NASA	National Aeronautics and Space Administration
ODE	Ordinary Differential Equation
ODERACS	Orbital Debris Radar Calibration Spheres
SMC	Sliding Mode Control
UTC	Coordinated Universal Time

Nomenclature

Symbol	Description	System of Units
λ	Mann Fixed-Point Iteration Coefficient	[-]
μ	Gravitational Parameter	$[m^3 \cdot s^{-2}]$
ρ	Density	$[kg \cdot m^{-3}]$
ω	Argument of the Perigee	[rad]
Ω	Right Ascension of the Ascending Node	[rad]
A	Effective Projection Area	$[m^2]$
A	State Matrix	[-]
A_d	Discretized State Matrix	[-]
B	Control Matrix	[-]
B_d	Discretized Control Matrix	[-]
C	Output Matrix	[-]
C_D	Drag Coefficient	[-]
C_d	Discretized Output Matrix	[-]
C_o	Controllability Matrix	[-]
D	Feedthrough Matrix	[-]
E	Disturbance Matrix	[-]
E_d	Discretized Disturbance Matrix	[-]
E	Eccentric Anomaly	[rad]
G	Concatenated Matrix	[-]
G_d	Discretized Concatenated Matrix	[-]
J	Quadratic Cost Function	[-]
J	Jacobian Matrix	[-]
J_2	Second Zonal Harmonic	[-]
K	Feedback Gain Matrix	[-]
M	Mean Anomaly	[rad]
O	Observability Matrix	[-]
P	Algebraic Riccati Equation Solution	[-]
T_{yw}	Transfer Function Matrix from w to y	[-]
V	Velocity	$[m \cdot s^{-1}]$
a	Semimajor Axis	[m]
a	Apogee	[m]
a_D	Atmospheric Drag	[N]
a_e	Earth's Equatorial Radius	[m]
dt	Time Step	[s]
e	Eccentricity	[-]
h	Error signals to be minimized error signals to be minimized	[-]
h_a	Apogee Height	[m]
h_p	Perigee Height	[m]
i	Inclination	[rad]
m	Mass	[kg]
n	Mean Motion	$[rad \cdot s^{-1}]$
p	Perigee	[m]
t	Time	[s]

$u(t)$	Control Input Vector	[-]
v	True Anomaly	[rad]
w	Disturbance Vector	[-]
$x(t)$	State Vector	[-]
$\dot{x}(t)$	State Differential Equation	[-]
x_k	Discretized State Vector	[-]
x_k^*	Desired Discretized State Vector	[-]
$y(t)$	Output Variable	[-]
$y^*(t)$	Desired Output Variable	[-]
$y(k)$	Discretized Output Variable	[-]
x, y, z	Relative Position Components in Cartesian Coordinates	[m]
$\dot{x}, \dot{y}, \dot{z}$	Relative Velocity Components in Cartesian Coordinates	[m · s^{-1}]
$\ddot{x}, \ddot{y}, \ddot{z}$	Relative Acceleration Components in Cartesian Coordinates	[m · s^{-2}]
r, θ, ϕ	Spherical Coordinates	[m; rad; rad]
$\dot{r}, \dot{\theta}, \dot{\phi}$	First Derivative of the Spherical Coordinates	[m · s^{-1}; rad · s^{-1}; rad · s^{-1}]
$\ddot{r}, \ddot{\theta}, \ddot{\phi}$	Second Derivative of the Spherical Coordinates	[m · s^{-2}; rad · s^{-2}; rad · s^{-2}]

Chapter 1

Introduction

In the past century, space exploration has evolved from uncertain experiments to great accomplishments. More importantly, it has made life easier, more productive, clean and sustainable on Earth. From Global Positioning System (GPS) to camera phones, to scratch-resistant lenses and countless other inventions that were made possible because of mankind's curiosity and will to achieve. In order to make the world a better place, acquire more knowledge and upgrade technologies, everything related to the space sector needs to be perfected, including space vehicles, thrusters, communication systems, controllers, power systems and structures.

After a space vehicle reaches its desired orbit, minor adjustments must be done to compensate for perturbations so that the orbital elements remain within chosen limits and the spacecraft performs its mission correctly [1]. Orbit perturbations change the initially calculated trajectory of a Keplerian orbit into a real orbit. The real orbit of a spacecraft, varying in time and space, is a sequence of osculating Keplerian orbits arranged in chronological order [2]. These orbit perturbations include the atmospheric drag, Earth's oblateness, solar winds, electromagnetic perturbations, the gravitational force of other celestial bodies (Sun, Moon, other planets and asteroids) and the solar radiation pressure, among others. Keplerian orbits are good approximations for most cases, with some exceptions like the atmospheric drag in Low Earth Orbit (LEO), the rotation of the ascending node and perigee due to Earth's oblateness, and lunar-solar effects at geosynchronous altitudes. This dissertation will focus on the specific case of a LEO, therefore the perturbations considered are the atmospheric drag and the Earth's oblateness [1].

It is necessary to know which orbit parameters need to be controlled and how to do it as efficiently as possible in order to do so properly and to save fuel. An uncontrolled trajectory can be suitable for a short mission spacecraft (1 to 3 years), for a more extended mission the vehicle requires some form of propulsion system which can be part of the vehicle itself or may be provided by another spacecraft, the International Space Station (ISS) periodically has its orbit raised a few kilometers through resupply vehicles like space shuttles or the Jules Verne automated transfer vehicle, however, in the case of emergency, the ISS can use the thrusters on its own Zvezda service module for orbital maneuvering to avoid space debris. The ISS is illustrated in Fig.1.1.

In satellite formation flying, multiple satellites work together in order to achieve a common goal. This is another case where changes in the initial orbit of each satellite must not be

tolerated because the relative positions between them need to be as stable as possible in order to make the whole system work.

The propulsion system controls the space vehicle by modifying the magnitude and direction of its velocity vector. This requires a thruster burn of a specific duration, and often requires a prior attitude change maneuver to properly orient the thruster(s) utilized for trajectory control [1].

Even though the orbit parameters are well calculated, and the main perturbations are considered, random errors in the control system caused by inaccurate orbit determination (due to other less significant perturbations) and the thrust implementation device may lead to failure of the entire spacecraft mission. This means a huge amount of financial resources and valuable science data wasted, thus, it is essential to evaluate the robustness of the controller and to develop a way to keep a spacecraft in a previously assigned orbit through orbital maneuvers executed by thrusters (station-keeping strategy) that tolerate a somewhat large magnitude of errors [2].

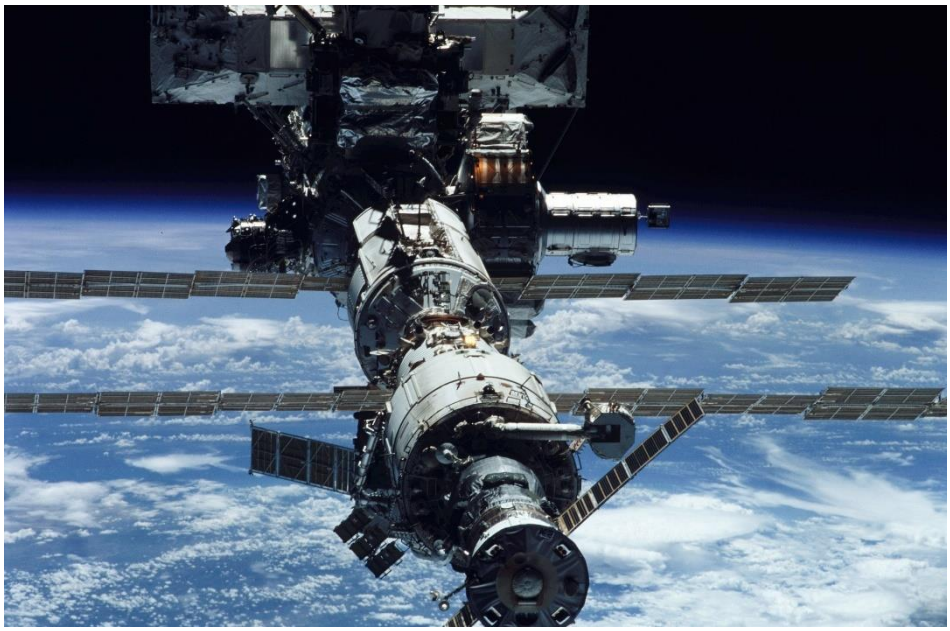


Figure 1.1: International Space Station [3]

1.1 Atmospheric Drag

When the orbit altitude is less than 1,500 km, air molecules in the Earth atmosphere interfere with the space vehicle in the direction of its motion. The force that the molecules exert on the spacecraft surface in the opposite direction of its motion is called atmospheric drag. The equation that determines the drag acceleration is:

$$\mathrm{a_D} = -(1/2)\rho V^2 (C_D A/m)\mathbf{i}_v \tag{1.1}$$

where ρ is the atmospheric density; V is the velocity of the spacecraft relative to the atmosphere; C_D is the drag coefficient, a dimensionless quantity; A is the spacecraft effective projection area; m is the mass of the spacecraft; and $\mathbf{i}_v$ is the unit vector of the spacecraft velocity relative to the atmosphere. It is concluded that the magnitude of the atmospheric drag increases with the increase of: - the atmospheric density (which increases with the decrease of altitude); - the velocity V; - the ballistic coefficient (C_DA/m).

The atmospheric density is computed from density models such as Jacchia 71 and MSIS90. For a dynamic density model like MSIS90, the density is a function of satellite altitude, solar flux, Earth magnetic index, time of day, and geocentric longitude/latitude of the spacecraft. According to an evaluation using decay data from NASA's ODERACS (Orbital Debris Radar Calibration Spheres) project, in which metal spheres were released from the space shuttle, the MSIS90 model is slightly more accurate than the Jacchia 71 model [4].

Atmospheric drag is directed along the satellite's negative velocity relative to the atmosphere. It gradually decreases the mean orbit altitude and speeds up the satellite, therefore periodic maneuvers are needed to recover the semimajor axis of the satellite's decaying orbit. For simple drag-makeup maneuvers, a single-burn or a Hohmann transfer is performed to recover the orbit altitude when the semimajor axis reaches its lower limit [4]. The Hohmann transfer orbit, represented in Fig.1.2 [5], is an elliptical orbit used to transfer between two circular orbits. Two burns are realized, the first burn, Δv, is used to transfer the initial circular orbit (1) into an elliptical transfer orbit (2) where the perigee is the same as the radius of the initial orbit. The second burn, Δv', is used to transfer the elliptical transfer orbit into a circular orbit where its initial apogee becomes the radius of the new higher circular orbit (3).

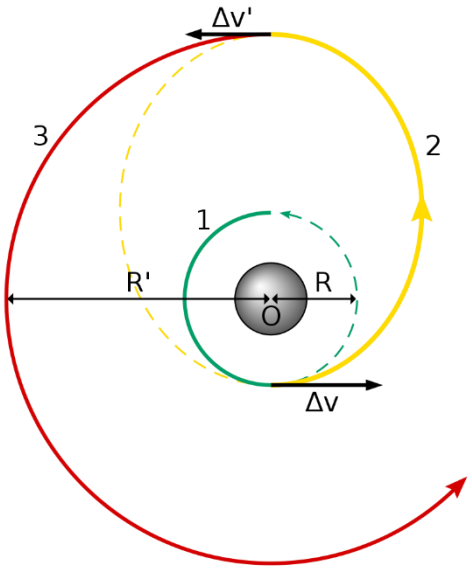


Figure 1.2: Hohmann transfer orbit [5]

1.2 Earth's Oblateness

The shape of the blue planet is very similar to a sphere. But the centrifugal force due to its rotation turns it into an oblate spheroid with flattening at the poles and an equatorial bulge (the equatorial radius is greater than the polar radius). At the equator, Earth, resembles a pear. Also, there are a lot of minor mass anomalies such as continents and mountains [1] (Fig.1.3 [6]). This lack of symmetry implies that the center of mass of the Earth is not in its geometric center, therefore, the force of gravity on an orbiting body is not perfectly directed towards it. While the gravitational field of a perfectly shaped spherical planet depends only on the distance from its center, oblateness also causes a variation with the angular distance from the equator (latitude). This is called a zonal variation. The dimensionless parameter which quantifies the major effects of oblateness on orbit is J_2, the second zonal harmonic. J_2 is not a universal constant meaning that each planet has its own value depending on its oblateness, rotation rate and mass. For Earth, $J_2 = 1.082 \cdot 10^{-3}$, approximately [7].

The potential generated by the non-sphericity of the Earth causes periodic variations in all the orbital elements, especially in the right ascension of the ascending node (Ω) and the argument of perigee (ω). The variation of the right ascension of the ascending node and the argument of the perigee due to J_2 are:

$$\dot{\Omega}_{J_2} = -1.5 n J_2 (a_e / a)^2 (\cos i)(1-e^2)^{-2} \tag{1.2}$$

$$\dot{\omega}_{J_2} = 0.75 n J_2 (a_e / a)^2 (4 - 5\sin^2 i)(1-e^2)^{-2} \tag{1.3}$$

Where n is the mean motion in deg/day, a_e is the Earth's equatorial radius, a is the semimajor axis in km, e is the eccentricity, i is the inclination, $\dot{\Omega}$ and $\dot{\omega}$ are in deg/day. In Fig.1.4 [7], graphs are presented for better understanding of the equations above. It is shown that these orbital elements have heavier deviations for lower orbits and for orbits with lower inclinations.

The oblateness term, J_2, is one of the most significant perturbation acting on Geosynchronous Equatorial Orbit (GEO) and below, which includes altitudes lower than 36,000 km. This perturbation implies major impacts in the initial orbit mainly because of the rotation of several degrees of the right ascension of the ascending node and the argument of perigee per day [8].

Besides the rotation of the orbit, the principal effect of the non-spherical mass distribution is to cause small changes in the shape of the orbit. Instead of a perfect ellipse, the space vehicle moves in an irregular one, with maximum deviations from a Keplerian orbit on the order of 3 km. In a repeating ground-track orbit, the satellite flies over the same location on the ground, day after day, and it is possible for resonances to build up when the satellite continuously passes [1]. It is fundamental to control the spacecraft's trajectory by avoiding these kinds of heavy deviations especially if a repeating ground-track orbit is desired so that the space vehicle can realize its mission properly.

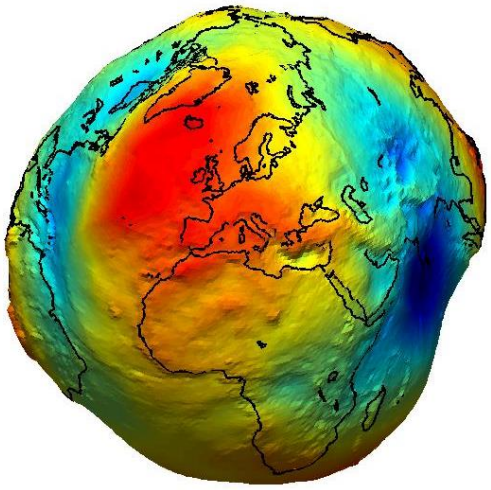


Figure 1.3: A computer-generated model of the Earth's actual shape shows it is shaped like a pear (an oblate spheroid) [6]

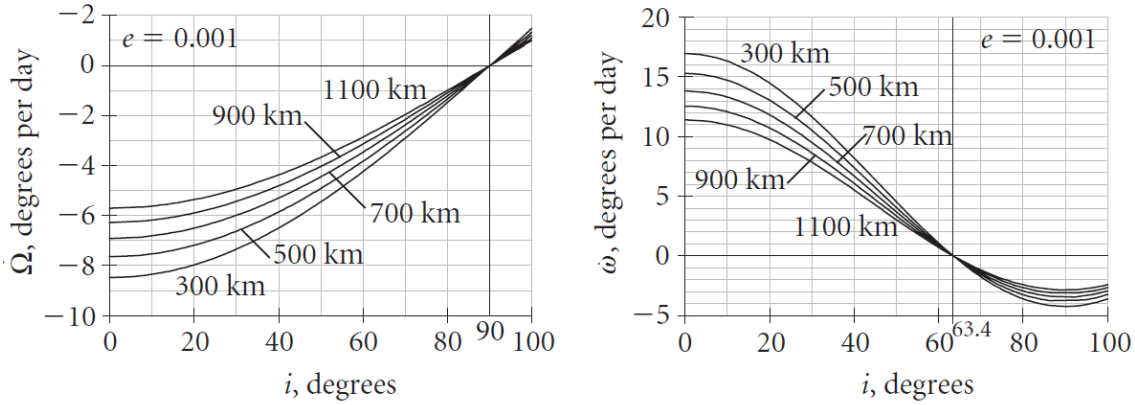


Figure 1.4: Regression of the node and advance of perigee for nearly circular orbits of altitudes 300 km to 1,100 km [7]

1.3 Orbit Control

There are six orbital elements that define the orbit of a space vehicle. These are: a, semimajor axis, e, eccentricity, i, inclination, Ω, right ascension of the ascending node, ω, argument of the perigee and v, true anomaly.

Only some elements require control for most space missions. True anomaly never needs control because it only defines the current location of the spacecraft of a given orbit. Of the other five elements, the semimajor axis and eccentricity are the ones that require maintenance first and the remaining need control depending on the mission. For LEO, as explained, the main perturbations are the atmospheric drag and the J_2 perturbation. These do not interfere directly with the inclination, however: - The semimajor axis and eccentricity decrease continuously because of the atmospheric drag; - The right ascension of the ascending node and the argument of the perigee may vary several degrees per day depending on the inclination and altitude because of the J_2 perturbation. Therefore, only a, e, Ω and ω require control for this case.

The orbit elements of the reference orbit are denoted by a^*, e^*, i^*, Ω^*, ω^*, and v^*, and the osculating elements are $a(t)$, $e(t)$, $i(t)$, $\Omega(t)$, $\omega(t)$, and $v(t)$. Then, the deviations of the orbit elements (or the associated error) can be written as $\delta a = a(t) - a^*$, $\delta e = e(t) - e^*$, $\delta\Omega = \Omega(t) - \Omega^*$ and $\delta\omega = \omega(t) - \omega^*$. The orbit elements maximum acceptable variation intervals are defined as I_a, I_e, I_Ω and I_ω and its values are attributed before the controller starts working.

I_e is omitted because its variation is very small, and a circular orbit is considered ideal (the atmospheric drag helps achieving $e = 0$). Also, every time a Hohmann transfer is executed because of the correction of the semimajor axis, the orbit becomes circular even if it is not for a long time. Therefore, it is pointless to define the maximum acceptable variation interval for the eccentricity.

When $\delta a\,(t) \in Ia$ is violated, a Hohmann transfer is executed to correct the semimajor axis and eccentricity simultaneously. When $\delta\Omega(t) \in I\Omega$ or $\delta\omega(t) \in I\omega$ is violated, an impulse will be performed at the perigee along the radial direction in the orbital plane to correct either the right ascension of the ascending node or the argument of the perigee, depending on which element is required to be corrected [2].

To simplify this concept, we can imagine a trajectory tube, where the acceptable deviations from the reference trajectory (ideal Keplerian orbit) are contained (see Fig.1.5 [9]). If the spacecraft trespasses this tube, the designed controller will activate one or more thrusters in order to place the space vehicle back on this imaginary tube.

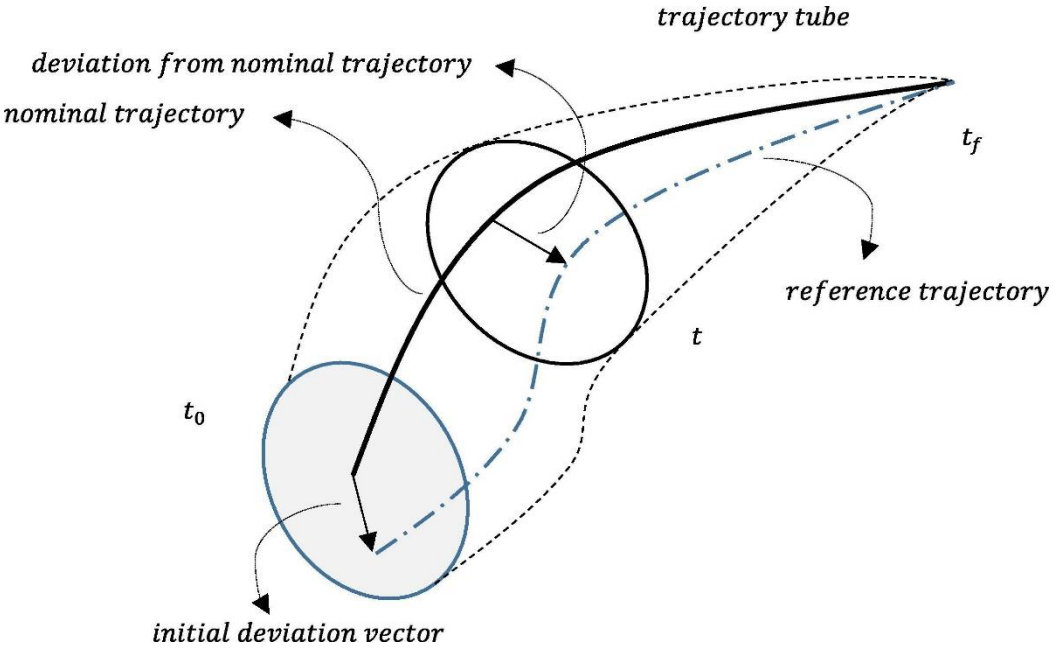


Figure 1.5: Conceptual presentation of the trajectory tube [9]

1.4 Objective

The main objective of the dissertation is to apply knowledge from Astrodynamics and Control theory in the development and simulation of an H_∞ robust controller on *Python*™ that can successfully alter the trajectory of a space vehicle on LEO, with perturbations considered, so that it follows a referenced Keplerian trajectory with the same initial conditions.

1.5 Layout of the Dissertation

This dissertation is organized as follows.

The current chapter exposed the problem, by providing an overview of the study in astrodynamics. Chapter 2 explains the dynamic model used to describe the movement of a space vehicle orbiting Earth and concludes the astrodynamics analysis. Chapter 3 studies the properties of linear systems, clarifies the reason for the chosen controller and explains its development. Chapter 4 demonstrates the simulations performed and respective analysis of the results. Chapter 5 concludes the dissertation and provides suggestions for future work.

Chapter 2

Dynamic Model of Space Vehicles for Earth Orbits

The equations of motion are deduced through the equations of acceleration in an inertial frame for a space vehicle orbiting Earth. Its scalar relations, can be given in a spherical coordinate system, represented in Fig.2.1, as [10]:

$$\begin{cases} \ddot{r} - r\dot{\theta}^2 = -\dfrac{\mu}{r^2} \\ 2\dot{r}\dot{\theta} + r\ddot{\theta} = 0 \end{cases} \tag{2.1}$$

The unit vector from the central body to the spacecraft's position, $\vec{r}$, its orbital angular velocity, ω and its orbital angular acceleration, $\dot{\omega}$, in the spherical coordinate system are, respectively [10]:

$$\vec{r} = \begin{bmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{bmatrix} \tag{2.2}$$

$$\omega = \begin{bmatrix} \dot{\phi}\sin\theta \\ -\dot{\phi}\cos\theta \\ \dot{\theta} \end{bmatrix} \tag{2.3}$$

$$\dot{\omega} = \begin{bmatrix} \ddot{\phi}\sin\theta + \dot{\theta}\dot{\phi}\cos\theta \\ -\ddot{\phi}\cos\theta + \dot{\theta}\dot{\phi}\sin\theta \\ \ddot{\theta} \end{bmatrix} \tag{2.4}$$

Substituting equations (2.2), (2.3) and (2.4) into (2.1) and solving for $\ddot{r}$, $\ddot{\theta}$ and $\ddot{\phi}$, conducts to the equations of motion for a space vehicle revolving around the Earth in a Keplerian orbit:

$$\begin{cases} \ddot{r} = r\dot{\theta}^2\sin^2\phi + r\dot{\phi}^2 - \dfrac{\mu}{r^2} + u_r \\ \ddot{\theta} = \dfrac{-2\dot{r}\dot{\theta}}{r} - 2\dot{\theta}\dot{\phi}\cot\phi + \dfrac{u_\theta}{r\sin\phi} \\ \ddot{\phi} = \dfrac{-2\dot{r}\dot{\phi}}{r} + \dot{\theta}^2\sin\phi\cos\phi + \dfrac{u_\phi}{r} \end{cases} \tag{2.5}$$

Where μ is the Earth's gravitational constant, r is the radial distance of the space vehicle from the center of the Earth, θ is the azimuth angle measured from the X-axis in the XY-

plane, and ϕ is the inclination measured from Z-axis to the vector r. u_r, u_θ and u_ϕ are added to the deduced equations and relate to the trust acceleration components in the $\vec{i}_r$, $\vec{i}_\theta$ and $\vec{i}_\phi$ directions, respectively. The space vehicle is modeled as a point mass.

Since the perturbations being analyzed in this dissertation are the atmospheric drag and the Earth's oblateness and a simple, yet effective solution was given to the first one, it is necessary to solve the problem for an orbit under the effects of the second zonal harmonic, J_2. The perturbing acceleration due to the earth's gravitational potentials are enough to determine the second zonal harmonic's major accelerations [11]:

$$V = \frac{\mu}{r}\left\{1 - \mathrm{J}_2\left(\frac{a_e}{r}\right)^2 P_2(\cos\phi)\right\} \tag{2.6}$$

$$a = \nabla V = \frac{\partial V}{\partial r}\vec{i}_r + \frac{1}{r\sin\phi}\frac{\partial V}{\partial \theta}\vec{i}_\theta + \frac{1}{r}\frac{\partial V}{\partial \phi}\vec{i}_\phi \tag{2.7}$$

Where μ is $3.986 \cdot 10^{14}\ m^3 \cdot s^{-2}$, J_2 is $1.082 \cdot 10^{-3}$, a_e is the equatorial radius of the Earth, $6{,}378 \cdot 10^3$ m approximately and $P_n(\cos\phi)$ are the Legendre polynomial functions.

The equations of motion in (2.5) can be rewritten, in order to consider these perturbing effects as presented below, using equations (2.6) and (2.7) [11]:

$$\begin{cases} \ddot{r} = r\dot{\theta}^2\sin^2\phi + r\dot{\phi}^2 - \dfrac{\mu}{r^2} + \dfrac{3}{2}\mu \mathrm{J}_2 a_e^{\,2}\dfrac{3\cos^2\phi - 1}{r^4} + u_r \\ \ddot{\theta} = \dfrac{-2\dot{r}\dot{\theta}}{r} - 2\dot{\theta}\dot{\phi}\cot\phi + \dfrac{u_\theta}{r\sin\phi} \\ \ddot{\phi} = \dfrac{-2\dot{r}\dot{\phi}}{r} + \dot{\theta}^2\sin\phi\cos\phi + 3\mu \mathrm{J}_2\dfrac{a_e^{\,2}}{r^5}\cos\phi\sin\phi + \dfrac{u_\phi}{r} \end{cases} \tag{2.8}$$

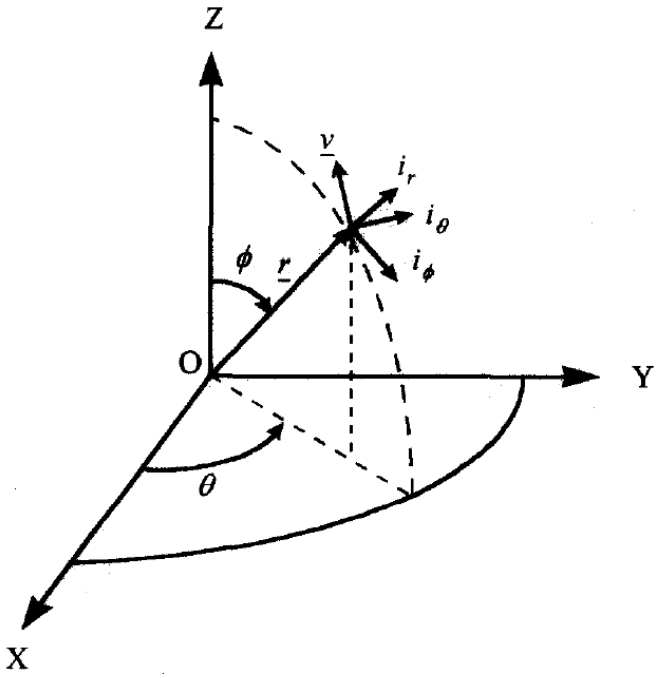


Figure 2.1: Motion of a space vehicle orbiting Earth in the spherical coordinate system [10]

2.1 Initial Conditions

Equations (2.5) and (2.8) can be rewritten using state space notation, which doubles the number of equations, but transforms them from second order differential equations to first order differential equations. State space notations attributes a new variable, x_n, for each variable from the spherical coordinate system:

$$\begin{cases} x_1 = r \\ x_2 = \dot{r} \\ x_3 = \theta \\ x_4 = \dot{\theta} \\ x_5 = \phi \\ x_6 = \dot{\phi} \end{cases} \tag{2.9}$$

Equations (2.10) are the state space form of equations (2.5) and equations (2.11) are the state space form of equations (2.8), as seen below:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 x_4{}^2 \sin x_5{}^2 + x_1 x_6{}^2 - \dfrac{\mu}{x_1{}^2} + u_{x_1} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -\dfrac{2x_2 x_4}{x_1} - 2x_4 x_6 \cot x_5 + \dfrac{u_{x_3}}{x_1 \sin x_5} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -\dfrac{2x_2 x_6}{x_1} + x_4{}^2 \sin x_5 \cos x_5 + \dfrac{u_{x_5}}{x_1} \end{cases} \tag{2.10}$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 x_4{}^2 \sin x_5{}^2 + x_1 x_6{}^2 - \dfrac{\mu}{x_1{}^2} + \dfrac{3}{2}\mu \mathrm{J}_2 a_e{}^2 \dfrac{3\cos x_5{}^2 - 1}{x_1{}^4} + u_{x_1} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -\dfrac{2x_2 x_4}{x_1} - 2x_4 x_6 \cot x_5 + \dfrac{u_{x_3}}{x_1 \sin x_5} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -\dfrac{2x_2 x_6}{x_1} + x_4{}^2 \sin x_5 \cos x_5 + 3\mu \mathrm{J}_2 \dfrac{a_e{}^2}{x1^5} \cos x_6 \sin x_6 + \dfrac{u_{x_5}}{x_1} \end{cases} \tag{2.11}$$

Before entering the control chapter of this dissertation, it is fundamental to find the initial conditions of the trajectory of a space vehicle, in order to simulate and discuss the Keplerian orbit, the Real orbit (associated with perturbations) and finally, the Controlled orbit, which is the Real orbit with the developed controller in order to change it into an ideal one.

For this problem, the initial conditions will be the ones mentioned in (2.3) and the chosen space vehicle is the International Space Station (ISS), because it fits with the requirements for this dissertation, it is on a Low Earth Orbit, it is the only place where astronauts live currently, of its international cooperation, NASA, National Aeronautics and Space Administration, (United States of America), Roscosmos (Russia), JAXA, Japan Aerospace Exploration Agency, ESA, European Space Agency, and CSA, Canadian Space Agency, and for its importance to humankind in general.

According to the DLR (German Aerospace Center), on the 20th of February of 2019 at 06:19:32 am (UTC), the orbit data of the ISS was [12]:

Table 2.1: ISS Orbit Data.

Eccentricity e	0.0005800
Inclination i	0.9013 rad
Perigee height h_p	403,000 m
Apogee height h_a	411,000 m
Right ascension of the ascending node Ω	3.886 rad
Argument of the perigee ω	0.7564 rad
Mean anomaly M	5.530 rad

In order to convert the values from the table into spherical coordinates, NASA's General Mission Analysis Tool (GMAT) was used.

The input parameters for the Keplerian state type, in GMAT, are the semi major axis, the eccentricity, the inclination, the right ascension of the ascending node, the argument of the perigee and the true anomaly. Therefore, it is only necessary to calculate the semi major axis, a, the eccentric anomaly, E, and the true anomaly, v. Below, usual equations relative to astrodynamics are shown that can determine the missing orbital elements.

$$a = \frac{2a_e + h_p + h_a}{2} \tag{2.12}$$

$$M = E - e\sin E \tag{2.13}$$

$$\tan\frac{v}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2} \tag{2.14}$$

The Earth's equatorial radius is approximately $6{,}378 \cdot 10^3$ m. Below, the remaining variables are calculated.

$$a = 6{,}785 \cdot 10^3 \text{ m} \tag{2.15}$$

$$E = 5.529 \text{ rad} \tag{2.16}$$

$$v = 5.528 \text{ rad} \tag{2.17}$$

The orbit of the ISS was simulated in GMAT with many other factors taken in account, like the date and time used above, the most accurate atmospheric model, MSIS90, referred in chapter

1 of this dissertation, the effects of J_2, and an approximate value for the ISS mass (419,000 kg), and for the drag area (8,140 m^2).

After simulation and through the use of "Command and Summary", under the section "Mission Sequence", and an "Ephemeris File", the following initial conditions were found for the orbit of the ISS for the spherical state:

Table 2.2: Initial Conditions for the Orbit of the ISS in Spherical Coordinates.

r	6782,000 m
$\dot{r}$	-3.049 m·s^{-1}
θ	-2.396 rad
$\dot{\theta}$	7.016·10^{-4} rad·s^{-1}
ϕ	1.569 rad
$\dot{\phi}$	-8.865·10^{-4} rad·s^{-1}

Chapter 3

Controller Design

Control systems engineering is a field of science whose main objective is to control the behavior of dynamic systems, using inputs, a system, outputs, actuators, sensors and, of course a controller.

For this case, a motion control problem, the most important scenario to understand is the Trajectory Tracking. It is a control method where the system output $y(t)$ is controlled over time in order to oblige a vehicle to follow a pre-determined and desired trajectory, $y^*(t)$. For this dissertation, the ISS is controlled over time in order to follow a desired trajectory (the Keplerian orbit), which is used as a reference signal.

The following sub-chapter will give an insight in basic control theory of linear systems.

3.1 Linear System

Ordinary Differential Equations (ODEs), can model equations of motion and control systems through the state-space approach, like in equations (2.10) and (2.11). These can also be described linearly as a state differential equation:

$$\dot{x}(t) = f[x(t), u(t), t] \tag{3.1}$$

Where $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector, $u(t) \in \mathbb{R}^p$ is the input or control variable, also given by a varying vector, f is a real function and t is the time variable [13].

The output variable $y(t) \in \mathbb{R}^q$ of the system can be observed and is given by the following equation:

$$y(t) = g[x(t), u(t), t] \tag{3.2}$$

Where g is a real function.

The system described with (3.1) and (3.2) gives the usual linear continuous-time system in the state-space form, also known as a finite-dimensional linear differential system and is represented by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.3}$$

Where $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times p}$ is the input matrix, $C \in \mathbb{R}^{q\times n}$ is the output matrix and $D \in \mathbb{R}^{q\times p}$ is the feedforward matrix.

3.2 Stability, Controllability and Observability

It is fundamental to determine whether a dynamic system is stable, controllable and observable in the evaluation of a dynamic linear system.

A system is stable if it always returns to an equilibrium point after small disturbances. A system is marginally stable if it has non attractive characteristics but, despite the circumstances it is still considered stable [14].

Without appropriate actuators the system can't be properly controllable and without proper sensors, the right states can't be measured and therefore the system can't be properly observed.

3.2.1 Stability

A usual method to verify the stability of a system is to analyze the eigenvalues of its A matrix.

The system is linearly stable if all the real parts of its eigenvalues are less than zero, $Re(\lambda) < 0$.

The system is marginally stable if its A matrix is critical. The eigenvalues of a critical matrix are all composed of negative real parts except for at least one part equal to zero [14].

3.2.2 Controllability

The dynamic system is controllable if and only if an input can be found that takes every state variable from a desired initial state to a desired final state in a finite amount of time [15].

For the system to be controllable, the rank of the controllability matrix C_o needs to be equal to n.

$$C_o = [B\ AB\ A^2B\ \ldots\ A^{n-1}B] \tag{3.4}$$

3.2.3 Observability

The dynamic system is observable if and only if the state of a system $x(t)$ can be determined given the input $u(t)$ and output $y(t)$ without knowing the initial state $x(t_0)$ [15].

For the system to be observable, the rank of the observability matrix O needs to be equal to n.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3.5}$$

3.3 Control Methods

The objective of a control system is to work in real life applications. The real environment can suffer variations and operating conditions may change, thus, the control system must be able to cope with these anomalies. Even if the environment does not change, other factors like uncertainties as well as noises must be considered. Mathematical representations of systems are simplified assumptions most of the time. Consequently, control systems designed based on simplified models are not expected to work correctly in real life applications. The property that a control system must possess for it to operate properly in realistic situations is defined as robustness. If a controller can be designed such that the whole system to be controlled remains stable when its parameters vary within certain expected limits, the system is said to possess robust stability. Besides, if it can satisfy performance specifications such as steady state tracking, disturbance rejection and speed of response requirements, it is said to possess robust performance [16].

As understood in previous chapters, a LEO control system for a spacecraft should not tolerate significant trajectory deviations nor a significant range of random errors. There is no such thing as a perfect system, however, for the presented problem, it is wise to choose a robust controller that is optimized to reduce all kinds of uncertainties because a space vehicle is astronomically costly due to development, fabrication, launch, maintenance and therefore the risks taken should be minimum.

In the following sub-sections, the main principles and fundaments of four different controllers will be analyzed and compared with space applications in mind.

3.3.1 Linear-Quadratic Regulator

Linear-Quadratic Regulator (LQR) is a controller developed using a mathematical algorithm that minimizes a cost function according to other factors, carefully thought and supplied by an engineer. The cost function is the sum of deviations between simulated measurements and their desired values. The algorithm calculates the controller settings that minimize these deviations [17].

The solution to the optimal control problem, in this case, is a feedback control where the measurements used for the feedback are the state variables. In this feedback control, each of the state variables is multiplied by a gain, K matrix, and the results are summed to get a single actuation value [17]. A schematic representation of the LQR controller is represented in Fig.3.1 [18].

In the case of a satellite on a LEO, the autonomous LQR controller achieves overall good performance when comparing to a ground-based optimized orbital control but more expensive in terms of Δv budgets when executing maneuvers [19].

When talking about robustness, there are some doubts about this controller. It has been shown that it has attractive stability robustness, however, some authors have concluded that it can suffer from poor robustness when special perturbations are introduced in the state-space formulation and this is the case for the motion of a LEO spacecraft [20].

For LQR, the optimal control problem determines the feedback gain, K matrix, of the optimal control vector:

$$u(t) = -Kx(t) \tag{3.6}$$

The goal of the LQR control system is to minimize the quadratic cost function:

$$J = \int_0^\infty (x^T Qx + u^T Ru)dt \tag{3.7}$$

Where Q and R are positive-definite real symmetric matrices, $Q \geq 0$ and $R > 0$. These act like weights. The Q matrix is related with the system state, x, and accounts for the error's importance. R is related to the control, u, and accounts for the intensity of the energy in the control signal [21].

The solution to the minimization problem in (3.7) is the positive-definite real symmetric matrix P, which corresponds to the Algebraic Riccati Equation (ARE). In the time-continuous form it is given by [21]:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{3.8}$$

And:

$$K = R^{-1}B^T P \tag{3.9}$$

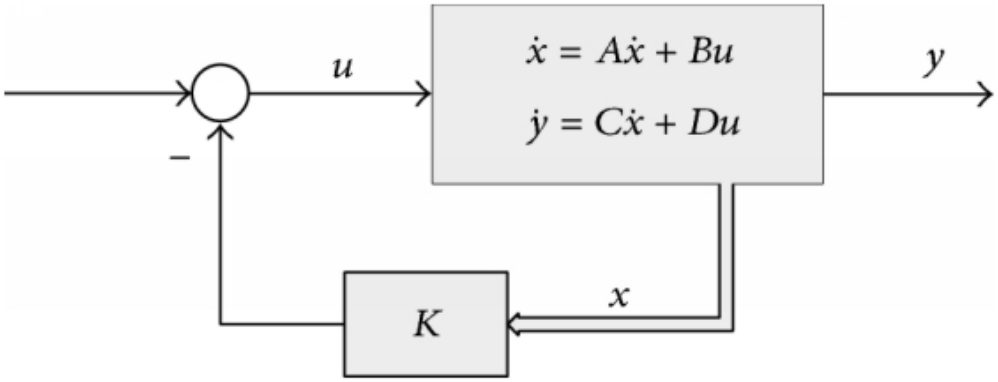


Figure 3.1: Structure of the LQR control system [18]

3.3.2 Sliding Mode Control

Sliding Mode Control (SMC) is a nonlinear control method that, when a discontinuous control signal is implemented, the dynamics of the nonlinear system are modified. Consequently, the controller "slides" along the cross-section of the system's normal behavior, stabilizing it.

A switching logic is designed with the controller, so that the closed loop dynamics of the system are dominated with the goal of acquiring a desirable trajectory.

The advantages of this controller are that it is robust even with a large class of perturbations, it does not require a great amount of information when compared to classical control techniques and it can stabilize systems via continuous state feedback laws.

The main disadvantage of this controller is the chattering. Theoretically, this is not a disadvantage, however, in real life applications the mechanical systems controlled with SMC would become worn and could even break. This is due to the high frequency of the actuators [22].

For a low-thrust LEO spacecraft, a SMC was simulated in [23]. It was shown that this controller, for this specific case, requires a smaller amount of fuel when executing maneuvers and offers more precise tracking control when compared to a linear model. It provides robust tracking control, even when the Earth's oblateness is considered and in the presence of atmospheric drag, solar radiation pressure and other unknown bounded disturbances.

3.3.3 Linear Matrix Inequality

Linear Matrix Inequality (LMI) method is a valuable tool to solve control problems. The principle of this method is to analyze a specific control problem as an optimization problem with linear objective and positive semidefinite constraints involving symmetric matrices that are related to the decision variables. A positive outcome arises from interpreting a problem in this form for an efficient, numerical solution is encountered. This method is suitable for problems with uncertainty, because a systematic procedure enables the formulation of a semidefinite program optimization problem that produces a robust solution [24].

The Riccati equation is used to design many controllers such as the LQR and the H_∞. Although this is a very powerful and important tool in control theory, it has some disadvantages in some specific cases, especially when additional requirements are added like the structural constraints on the control gain form, the limitation on the gain magnitude and the minimization of the upper bond on the value of the cost function. It was discovered, however, that it is advantageous to solve problems like the ones mentioned above using an LMI method. This method belongs in the group of convex problems and, therefore, guarantees a cost control gain [25].

For a sun-synchronous LEO satellite formation, it was discovered in [26] that an H_∞ synthesis with a different approach (use of LMIs instead of the classic Riccati equation) can produce the same performance as a classic H_∞ controller but with the benefit of allowing flexibility and a distributed architecture. It can, for example, have better robustness when model uncertainties are present and minimize the propellant consumption of the spacecraft while maintaining acceptable deviations from the previously defined orbit.

3.3.4 H_∞ Controller

The H-infinity (H_∞) control method is an approach to robust active control that appeared in the second half of the 20th century. It was a promising controller, however, due to technological limitations like weak processing power, the saturation of actuators and sensors

was a risk. Nowadays, this risk has decreased significantly due to technological advancement, leading to the possibility of the application of the H_∞ control method in newer systems.

The H_∞ controller contains a system, Σ, and its state-space representation. The system has two inputs: w, which refers to the disturbances and u, which refers to the controlled input. u is dependent of the controller, Control and Monitoring Processor (CMP), Σ_{cmp}, and the new measurement output, y. The two outputs of the system are h, that consists of the error signals to be minimized and y. A schematic representation of the H_∞ controller is represented in Fig.3.2 [16].

The standard H_∞ optimization problem is to solve an internally stabilizing proper measurement feedback control law, Σ_{cmp}, such that the H_∞ norm of the overall closed-loop transfer matrix function from w to h is minimized [16]. This means that we are defining the norm as a measure of the worst-case gain in terms of energy that the system can have [26]. If a controller is designed to work in the worst-case scenario it is expected that it will have an excellent overall performance in regular conditions.

When analyzing the robustness of a system, the H_∞ controller is reliable even considering a high noise environment and it could potentially lead to less fuel consumption when executing space maneuvers. The total force applied to the satellite within 100 minutes of operation with the H_∞ controller is much less compared to a satellite controlled by the LQR [27].

H_∞ enhances the control system performance by minimizing the control error and control effort. It also provides precise control performance and high robustness to model uncertainties and external disturbances. Signals from GPS satellites are used to directly or indirectly determine the satellite orbital state, and is used in real time feed-back loop to continuously estimate the error between a satellite and an ideal pre-defined orbit. For the specific cases of geostationary and LEO satellites, H_∞ provides the greatest robustness to the feedback control loop, increases the operating range of the system and enhances the convergence and stability of the control loop [28].

Satellites can be controlled through a ground station or can have a built-in autonomous system. For example, geostationary satellites take advantage of the stationary nature relative to the ground stations, which provide a continuous ranging, tracking and commanding, relieving the computational processing power on-board the satellite. LEO satellites, on the other hand, have intermittent contact with the ground station which implies a limited capacity of controlling the satellites in real time. Therefore, these are usually equipped with more on-board processing capability compared to geostationary satellites to provide improved autonomy in navigation and greater control performance [28].

After all the given evidence, analysis and comparisons, an H_∞ controller was selected for the purpose of this dissertation.

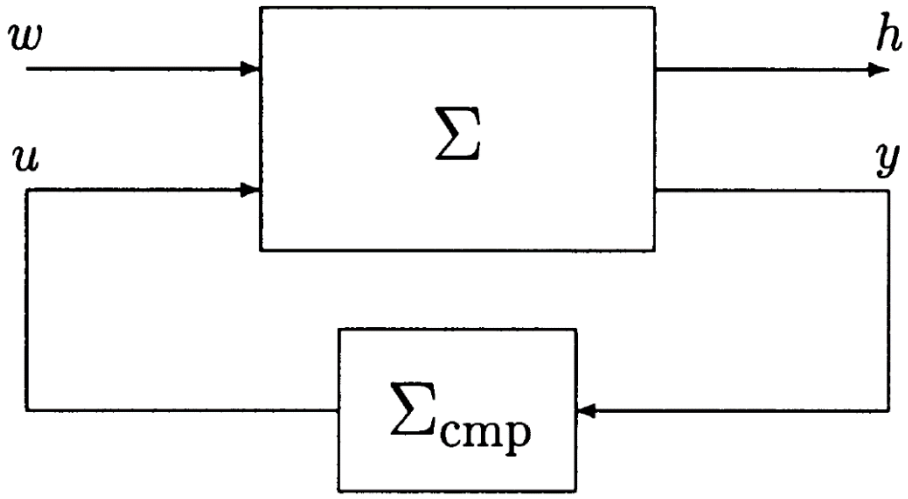


Figure 3.2: Structure of the H_∞ control system [16]

The linear continuous-time system in the state-space form of an H_∞ controller is given below:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ew(t) \\ y(t) &= Cx(t) \\ x(t_0) &= x_0 \end{aligned} \tag{3.10}$$

Where $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times p}$ is the input matrix, $C \in \mathbb{R}^{q\times n}$ is the output matrix and $E \in \mathbb{R}^{n\times p}$ is the disturbance matrix.

And where $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector, $y(t) \in \mathbb{R}^q$ is the output variable, $u(t) \in \mathbb{R}^p$ is the input or control variable, $w \in \mathbb{R}^p$ is the disturbance vector and x_0 is the state vector referent to the initial conditions.

The D matrix in equation (3.3), for this dissertation, is a zero matrix because the system model does not need direct feedthrough.

The system should be stable or marginally stable, controllable and observable.

The first step to develop an H_∞ controller is to linearize the state-space system. This dissertation uses the one represented in equation (2.11).

In the state-space notation of a linear system, the A matrix is given by the Jacobian Matrix, J, of a non-linear system, excluding control or disturbance variables.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} & \frac{\partial f_1}{\partial x_5} & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_2}{\partial x_5} & \frac{\partial f_2}{\partial x_6} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_3}{\partial x_5} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_3} & \frac{\partial f_5}{\partial x_4} & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_4} & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix} \tag{3.11}$$

Finally, the A matrix will be the Jacobian matrix with x_1, x_2, x_3, x_4, x_5 and x_6 equal to their respective equilibrium points.

The B, C and E matrices are composed of the information found on the non-linear system's variables for control, outputs and disturbances respectively.

A discrete-time approach should be considered so that the controller might be implemented on a real spacecraft and perform its tasks according to its digital devices. Therefore, the A, B, C, D and E matrices need to be discretized into A_d, B_d, C_d, D_d, E_d.

The A_d matrix should be nonsingular and for this to occur, the following condition must be secured [29]:

$$\det(A_d) \neq 0 \tag{3.12}$$

The objective of the H_∞ controller is to minimize the norm of the transfer function matrix T_{yw} from w to y, given by [29]:

$$T_{yw} = C(zI - A + BK)^{-1}E \tag{3.13}$$

Where z is the z-transform, which can be considered the discrete-time equivalent of the Laplace transform.

However, since it is difficult to achieve this goal, the minimum possible H_∞-norm of T_{yw} by γ_o is denoted and an approximated γ is estimated such that $\gamma - \gamma_o > 0$ is less than a prespecified tolerance. Also, u_k is determined so that the following condition is true [29]:

$$\left\| T_{yw} \right\|_\infty < \gamma \tag{3.14}$$

For this dissertation the standard value, $\gamma = 1$, is assumed.

The optimal control problem determines the feedback gain, K matrix, of the optimal control vector:

$$u_k = -Kx_k \tag{3.15}$$

A stabilizing solution to (3.14) is found if there exists $P > 0$ that satisfies the following algebraic Riccati equation [29]:

$$P = A_d^T P A_d - A_d^T P B_d (I + B_d^T P B_d)^{-1} B_d^T P A_d + P E_d (I + E_d^T P E_d)^{-1} E_d^T P + C_d^T C_d \tag{3.16}$$

P is estimated iteratively. Its initial state, P_0, is an identity matrix of appropriate size and with each iteration, a better solution is found for the problem. The frobenius norm of the difference after every iteration, $\left\| P_{k+1} - P_k \right\|_{\mathrm{F}}$, should become lower and lower, however if this

is not the case and a convergence problem is encountered, equation (3.16) should be rewritten in to:

$$P = P\lambda^k + (1-\lambda^k)(A_d^T PA_d - A_d^T PB_d (I + B_d^T PB_d)^{-1} B_d^T PA_d + PE_d (I + E_d^T PE_d)^{-1} E_d^T P + C_d^T C_d) \quad (3.17)$$

Where λ is an attributed coefficient and should be]0.5, 1[and $k \in \mathbb{N}^0$ is the number of the iteration. This is called a Mann fixed-point iteration.

If there exists a positive definite solution P to (3.16) or (3.17), then (3.14) is guaranteed by the following stabilizing gain matrix [29]:

$$K = (I + B_d^T PB_d)^{-1} B_d^T PA_d \quad (3.18)$$

For a trajectory tracking problem, a reference signal must be given to the system so that a desired trajectory is followed. For that to happen, the desired state of the system, x_k^*, should be known for every instant of time that the controller is performing its duty. Equation (3.15) can be rewritten in order to take this factor into account as seen below:

$$u_k = -K(x_k - x_k^*) \quad (3.19)$$

The control vector is calculated in a way so that the error of the system, $e = x_k - x_k^*$, is minimized over time.

Figure 3.3 represents the structure of the designed H_∞ control system, based on figure 3.2.

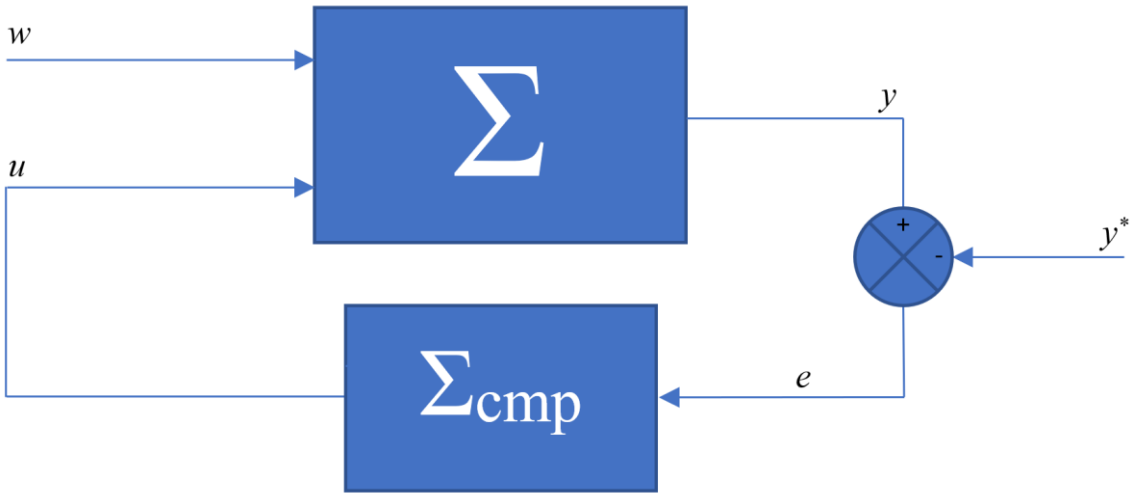


Figure 3.3: Structure of the designed H_∞ control system

The structure of the designed H_∞ control system contains the nonlinear system, Σ, the controller, Σ_{cmp}, two inputs, w and u, one output, y, and the output of the desired state, y^*. The disturbances are given by the vector w, the controlled input is given by the vector u, which is dependent of the controller, Σ_{cmp}, and the new measurement output, y. This output, collectively with the desired state output and respective state vectors, x and x^*, will give the current error, which, along with the gain matrix K, will be used to calculate the control vector, u, for every step, dt.

Chapter 4

Numerical Results and Discussion

This chapter focuses on presenting the numerical results from the development of the controller, which is carefully described in the previous chapters, and the simulation of three trajectories: the reference orbit, also known as the Keplerian Orbit for this dissertation, the Real Orbit, with perturbations considered but without control of any kind and the Controlled Orbit, with perturbations considered but with an integrated H_∞ controller that obliges the real orbit to follow the referenced one.

All the results, simulations and graphs were achieved using the $Python^{TM}$ programming language.

Section 4.1 gives a brief description of the dynamic model and shows the matrices that helped develop the controller according to previous chapters.

Section 4.2 analyses some characteristics of the dynamic system and explains how the simulations were executed.

Section 4.3 presents the most relevant results along with a detailed analysis and discussion, including comparisons between the real and the controlled orbits.

4.1 Dynamic Model

The movement of a spacecraft orbiting Earth with its oblateness considered is given by a dynamic model described in the second chapter of this dissertation through equations (2.10) and (2.11). The state vector, representing the position, its respective derivative in spherical coordinates and the control vector, relative to the thrust acceleration components in the directions of every variable relative to the position are given below:

$$x(t) = \left[r, \dot{r}, \theta, \dot{\theta}, \phi, \dot{\phi} \right]^T \tag{4.1}$$

$$u(t) = \left[u_r, u_\theta, u_\phi \right]^T \tag{4.2}$$

The equilibrium points used to determine the A matrix, through the Jacobian one were:

$$x_{eq} = \begin{bmatrix} 6.760 \cdot 10^{6} \\ 6.480 \cdot 10^{-1} \\ 0 \\ 1.270 \cdot 10^{-3} \\ 1.571 \\ 5.159 \cdot 10^{-6} \end{bmatrix} \tag{4.3}$$

These were estimated according to the values obtained from simulating the real orbit, which is later explained in this chapter.

The matrices A, B, C, E and respective discretized forms, described in chapter 3, are represented as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 4.194 \cdot 10^{-6} & 0 & 0 & 1.718 \cdot 10^{4} & 2.070 \cdot 10^{-5} & 6.975 \cdot 10^{1} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3.603 \cdot 10^{-17} & -3.758 \cdot 10^{-10} & 0 & -1.917 \cdot 10^{-7} & 1.311 \cdot 10^{-8} & -2.410 \cdot 10^{-9} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1.463 \cdot 10^{-19} & -1.526 \cdot 10^{-12} & 0 & 2.410 \cdot 10^{-9} & -1.614 \cdot 10^{-6} & -1.917 \cdot 10^{-7} \end{bmatrix} \tag{4.4}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1.479 \cdot 10^{-7} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1.479 \cdot 10^{-7} \end{bmatrix} \tag{4.5}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.6}$$

$$E = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.7}$$

To discretize, the *Python*^{TM} command *cont2discrete(A,B,C,D)* was used. It can be found in the *signal* library. Since it is an H_∞ control problem and the existence of one more matrix is present, E, a G matrix is used instead of a B matrix. G is a concatenated matrix of B and E.

$$G = \begin{bmatrix} B & E \end{bmatrix} \tag{4.8}$$

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.479 \cdot 10^{-7} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.479 \cdot 10^{-7} & 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

Later, the discretized forms of B and E will be given by the expression below:

$$G_d = \begin{bmatrix} B_d & E_d \end{bmatrix} \tag{4.10}$$

After computing *cont2discrete(A,G,C,D)* the following results were obtained:

$$A_d = \begin{bmatrix} 1 & 10 \cdot 10^{-2} & 0 & 8.588 \cdot 10^{1} & 1.316 \cdot 10^{-7} & 3.488 \cdot 10^{-1} \\ 4.194 \cdot 10^{-7} & 10 \cdot 10^{-1} & 0 & 1.718 \cdot 10^{3} & 2.633 \cdot 10^{-6} & 6.975 \\ -2.139 \cdot 10^{-19} & -1.879 \cdot 10^{-12} & 1 & 10 \cdot 10^{-2} & 6.554 \cdot 10^{-11} & -1.533 \cdot 10^{-11} \\ -4.279 \cdot 10^{-18} & -3.758 \cdot 10^{-11} & 0 & 10 \cdot 10^{-1} & 1.311 \cdot 10^{-9} & -3.066 \cdot 10^{-10} \\ -8.689 \cdot 10^{-22} & -7.632 \cdot 10^{-15} & 0 & 5.497 \cdot 10^{-12} & 10 \cdot 10^{-1} & 10 \cdot 10^{-2} \\ -1.738 \cdot 10^{-20} & -1.526 \cdot 10^{-13} & 0 & 1.099 \cdot 10^{-10} & -1.614 \cdot 10^{-7} & 10 \cdot 10^{-1} \end{bmatrix} \tag{4.11}$$

$$B_d = \begin{bmatrix} 5 \cdot 10^{-3} & 6.352 \cdot 10^{-7} & 2.580 \cdot 10^{-9} \\ 10 \cdot 10^{-2} & 1.270 \cdot 10^{-5} & 5.159 \cdot 10^{-8} \\ -9.369 \cdot 10^{-14} & 7.396 \cdot 10^{-10} & -1.134 \cdot 10^{-19} \\ -1.879 \cdot 10^{-12} & 1.479 \cdot 10^{-8} & -2.267 \cdot 10^{-18} \\ -3.816 \cdot 10^{-16} & 4.065 \cdot 10^{-20} & 7.396 \cdot 10^{-10} \\ -7.632 \cdot 10^{-15} & 8.131 \cdot 10^{-19} & 1.479 \cdot 10^{-8} \end{bmatrix} \tag{4.12}$$

$$C_d = \begin{bmatrix} 1 & 5 \cdot 10^{-2} & 0 & 4.294 \cdot 10^{1} & 6.582 \cdot 10^{-8} & 1.744 \cdot 10^{-1} \\ -1.070 \cdot 10^{-19} & -9.396 \cdot 10^{-13} & 1 & 5 \cdot 10^{-2} & 3.277 \cdot 10^{-11} & -7.664 \cdot 10^{-12} \\ -4.350 \cdot 10^{-22} & -3.818 \cdot 10^{-15} & 0 & 2.746 \cdot 10^{-12} & 10 \cdot 10^{-1} & 5 \cdot 10^{-2} \end{bmatrix} \tag{4.13}$$

$$E_d = \begin{bmatrix} 5\cdot10^{-3} & 0 & 1.744\cdot10^{-2} \\ 10\cdot10^{-2} & 0 & 3.488\cdot10^{-1} \\ -9.396\cdot10^{-14} & 0 & -7.664\cdot10^{-13} \\ -1.879\cdot10^{-12} & 0 & -1.533\cdot10^{-11} \\ -3.816\cdot10^{-16} & 0 & 5\cdot10^{-3} \\ -7.632\cdot10^{-15} & 0 & 10\cdot10^{-2} \end{bmatrix} \quad (4.14)$$

After obtaining all these matrices, the Riccati equation can be properly iterated in order to find the optimal solution, P through equation (3.16). However, the frobenius norm of the difference after every iteration, $\|P_{k+1} - P_k\|_F$, was becoming greater and greater. So, a convergence problem was found and, therefore, the optimal solution, P was found through equation (3.17) instead.

$$P = \begin{bmatrix} 2.258\cdot10^{6} & 2.457\cdot10^{5} & 2.688\cdot10^{-1} & -3.517\cdot10^{11} & -9.016\cdot10^{5} & -1.433\cdot10^{9} \\ 2.457\cdot10^{5} & 2.751\cdot10^{4} & 5.324\cdot10^{-3} & -1.868\cdot10^{10} & -5.064\cdot10^{4} & -7.607\cdot10^{7} \\ 2.688\cdot10^{-1} & 5.324\cdot10^{-3} & 1.920\cdot10^{2} & -6.899\cdot10^{5} & -1.682 & -2.820\cdot10^{3} \\ -3.517\cdot10^{11} & -1.868\cdot10^{10} & -6.899\cdot10^{5} & 5.853\cdot10^{17} & 1.426\cdot10^{12} & 2.386\cdot10^{15} \\ -9.016\cdot10^{5} & -5.064\cdot10^{4} & -1.682 & 1.426\cdot10^{12} & 3.476\cdot10^{6} & 5.814\cdot10^{9} \\ -1.433\cdot10^{9} & -7.607\cdot10^{7} & -2.820\cdot10^{3} & 2.386\cdot10^{15} & 5.814\cdot10^{9} & 9.728\cdot10^{12} \end{bmatrix} \quad (4.15)$$

Finally, the gain matrix K can be calculated:

$$K = \begin{bmatrix} 6.061\cdot10^{1} & 1.301\cdot10^{1} & -4.242\cdot10^{-6} & -4.190\cdot10^{4} & -1.468 & -1.533\cdot10^{2} \\ -1.511\cdot10^{1} & -7.641\cdot10^{-1} & -8.083\cdot10^{-5} & 6.706\cdot10^{7} & 1.629\cdot10^{2} & 2.734\cdot10^{5} \\ -6.157\cdot10^{-2} & -3.111\cdot10^{-3} & -4.417\cdot10^{-7} & 2.734\cdot10^{5} & 6.641\cdot10^{-1} & 1.115\cdot10^{3} \end{bmatrix} \quad (4.16)$$

4.2 First Analysis

After obtaining all these matrices, we can also analyze some important properties of the system referred in chapter 3, like stability, controllability and observability.

Through the analysis of the eigenvalues of the A matrix, given below, we can conclude that the system is marginally stable because all the real parts of its eigenvalues are negative except two of them, which are zero.

$$\begin{bmatrix} 0.000\cdot10^{0}+0.000000\,j \\ -4.107\cdot10^{-7}+0.001504\,j \\ -4.107\cdot10^{-7}-0.001504\,j \\ 0.000\cdot10^{0}+0.000000\,j \\ -9.562\cdot10^{-8}+0.001270\,j \\ -9.562\cdot10^{-8}-0.001270\,j \end{bmatrix}^{T} \tag{4.17}$$

The ranks of the controllability and observability matrices, (3.4) and (3.5) respectively, were also determined. Both have a rank equal to n, which proves the controllability and observability of the system.

It was also concluded that the A_d matrix was nonsingular, as requested by (3.12), for its determinant is different than zero.

For this dissertation, the simulated systems were always the nonlinearized ones. This is because some variables were too sensitive to be simulated on a simplified system and it was concluded that the system was stable but not attractive. The advantage in using the nonlinear system is that a more robust and realist system is being used to describe the motion of the vehicle, however it takes more time to compute the wanted results. The linearized system was, consequently, only utilized to design the controller.

To perform the simulations for this dissertation, a numerical method that gives the states of the system, x_k, for every time step, dt, with just its initial conditions, x_0, and the differential equations, $\dot{x}$, needs to be chosen.

The Keplerian and Real orbits were simulated using a simple solver called *odeint* that integrates a system of Ordinary Differential Equations (ODEs). It is found on a *Python™* library called *integrate*. It proved to be a great solver because of its accuracy and fast execution time.

The systems used for the Keplerian and Real orbits were the ones in (2.10) and (2.11), respectively without the control variable.

For the Controlled Orbit, a more robust method was needed to integrate the system because there was a new variable to consider, the control vector, that changed for every dt. The Butcher's Method [30], also known as the fifth order Runge-Kutta algorithm, was the numerical method chosen to simulate the system represented in (2.11) with the control variable varying properly for every dt. It was chosen due to its excellent accuracy, even though it has a relatively slow execution time.

It is a solver of ODEs in the following form:

$$\dot{x} = f(x,u) \tag{4.18}$$

Where, $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector and $u(t) \in \mathbb{R}^p$ is the input or control variable.

Finally, The Butcher's Method is described below:

$$x_{k+1} = x_k + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6) \tag{4.19}$$

With:

$$\begin{aligned}
k_1 &= dt\ f(x_k, u_k) \\
k_2 &= dt\ f(x_k + \frac{1}{4}k_1, u_k) \\
k_3 &= dt\ f(x_k + \frac{1}{8}k_1 + \frac{1}{8}k_2, u_k) \\
k_4 &= dt\ f(x_k - \frac{1}{2}k_2 + k_3, u_k) \\
k_5 &= dt\ f(x_k + \frac{3}{16}k_1 + \frac{9}{16}k_4, u_k) \\
k_6 &= dt\ f(x_k - \frac{3}{7}k_1 + \frac{2}{7}k_2 + \frac{12}{7}k_3 - \frac{12}{7}k_4 + \frac{8}{7}k_5, u_k)
\end{aligned} \tag{4.20}$$

Due to the control actuation frequency, time step used was $dt = 0.1\ \mathrm{s}$, which proved to be a small enough step to compute all the results with excellent accuracy, although with the inconvenience, together with the fact of using a complex system (the nonlinearized one) and a robust solver for ODEs, of having a relatively slow computational time. However, it is essential to keep in mind that for the purpose of this dissertation, the primary objective is to develop a robust controller to work on critical real-life applications with a cutting-edge on-board processor designed for a specific space vehicle.

4.3 Detailed Analysis

The previous sections of this dissertation described the problem and proposed a solution. The purpose of this sub chapter is to reveal the most important simulations of this work and analyze every single one of them.

For better visualization and understanding of the problem, a code of colors was implemented. The green color was attributed to the Keplerian Orbit, the red color was attributed to the Real Orbit and the blue color was attributed to the Controlled Orbit.

The default simulation time used for this dissertation was t = 200,000 s, which is approximately 2.5 days. This is enough to understand how the Real Orbit changes with time, because, in 2.5 days, the ISS revolves forty times around Earth, approximately. Also, it provides more than enough time to prove the robustness of the designed controller.

Below, the variation of the spherical coordinates for the Keplerian Orbit in function of time is graphically represented:

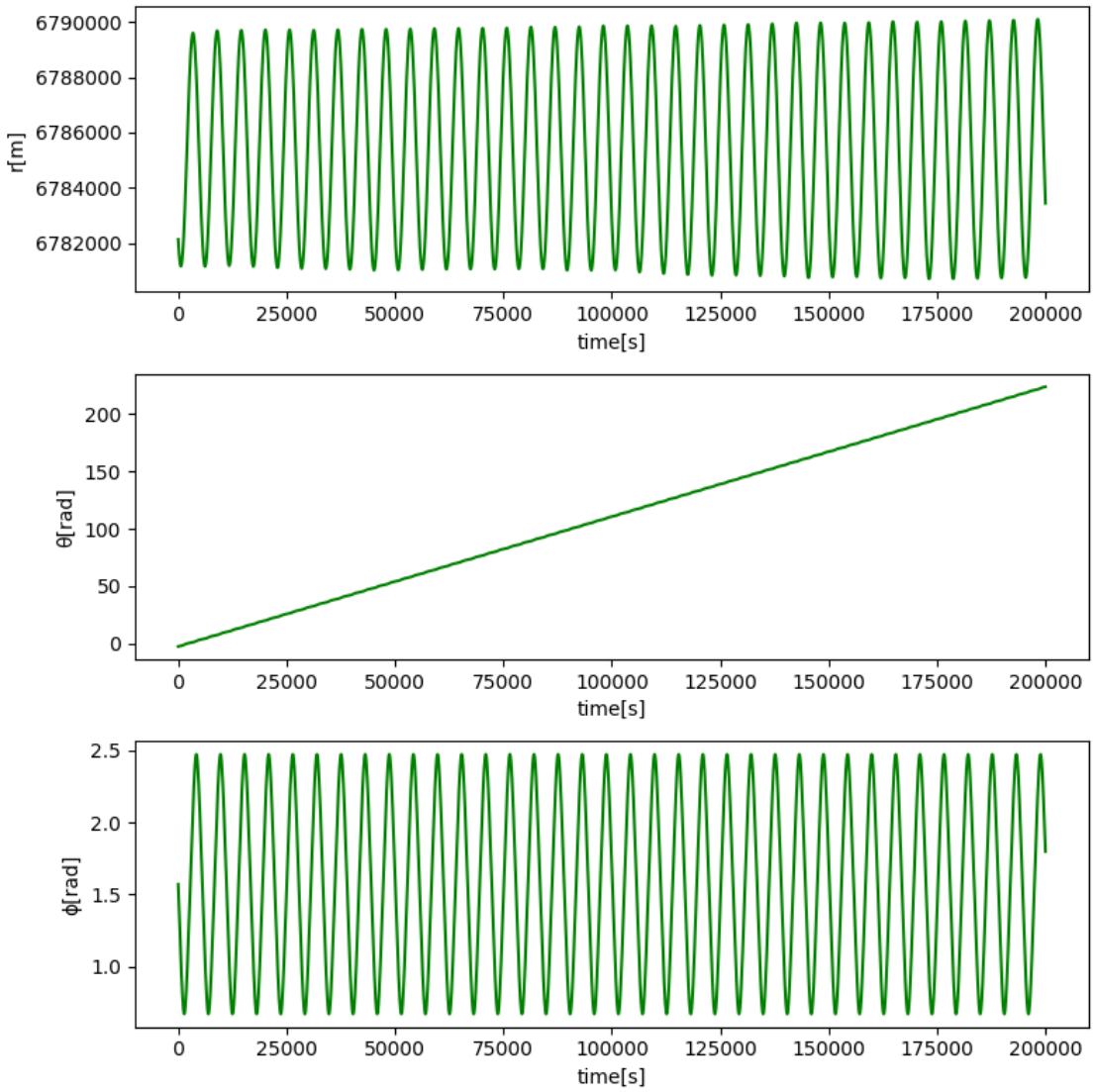


Figure 4.1: Variation of the spherical elements for the Keplerian Orbit in function of time

It is expected that r, the radial distance of the space vehicle from the center of the Earth varies a bit because the eccentricity of the orbit of the ISS is not null. When r = 6,790,000 m, the space vehicle is on the apogee of the orbit, approximately, and when r = 6,782,000 m, the space vehicle is on the perigee of the orbit, approximately. The other spherical elements, θ and ϕ vary so that the space vehicle can revolve around Earth. The first, the azimuth angle measured from the X-axis in the XY-plane, increases indefinitely and ϕ, the inclination measured from Z-axis to the vector r decreases and increases constantly.

Following the previous analysis, the same is represented below, but for the Real Orbit:

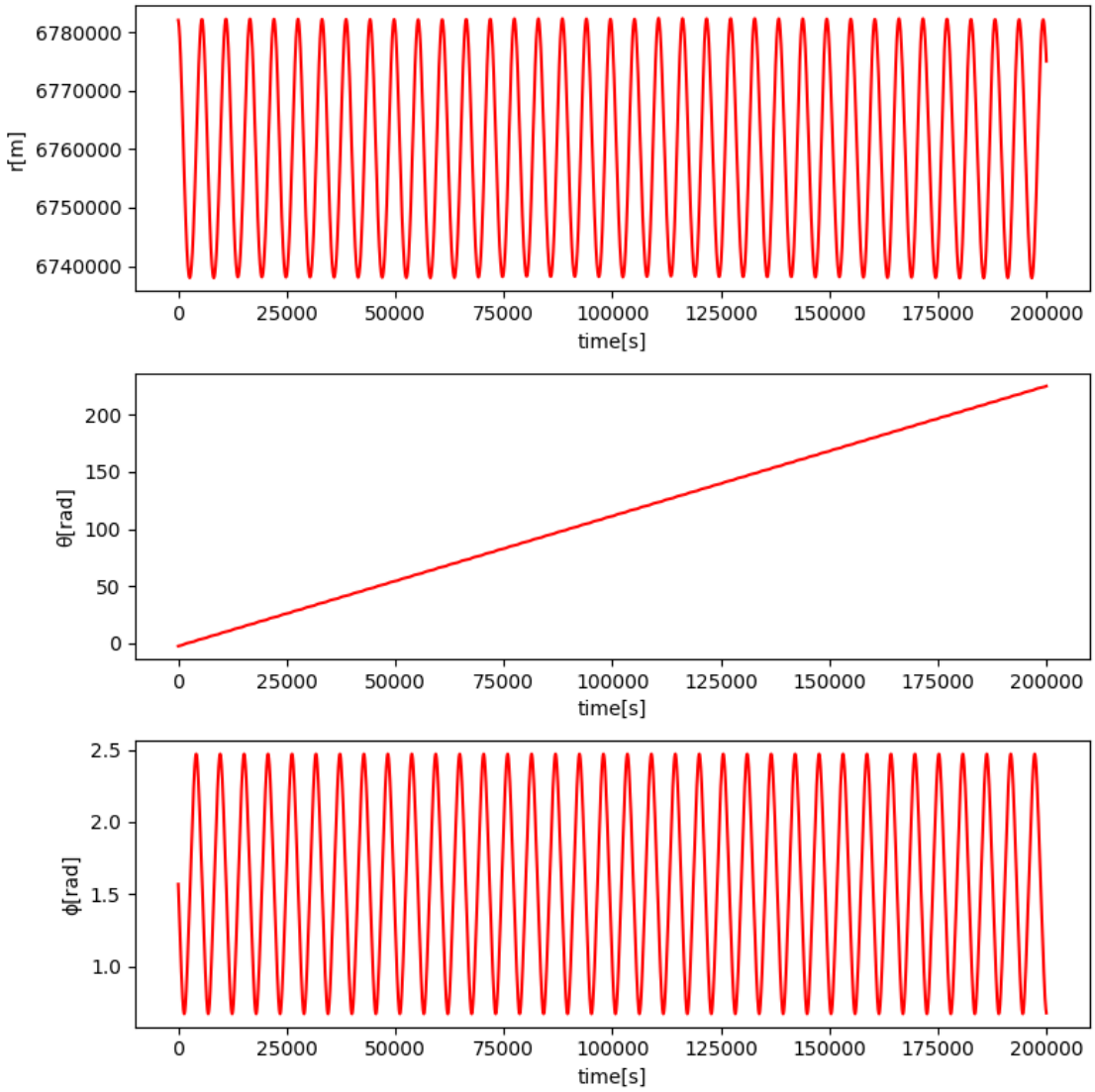


Figure 4.2: Variation of the spherical elements for the Real Orbit in function of time

The main difference encountered from the Keplerian Orbit is that r has a much bigger variance. The apogee of the orbit is now found at r = 6,780,000 m, approximately and the perigee of the orbit is now found at r = 6,740,000 m, approximately. Although the angles that compose the spherical elements seem to not variate much, later it is proven that this is not the case.

The current analysis is also done for the Controlled Orbit:

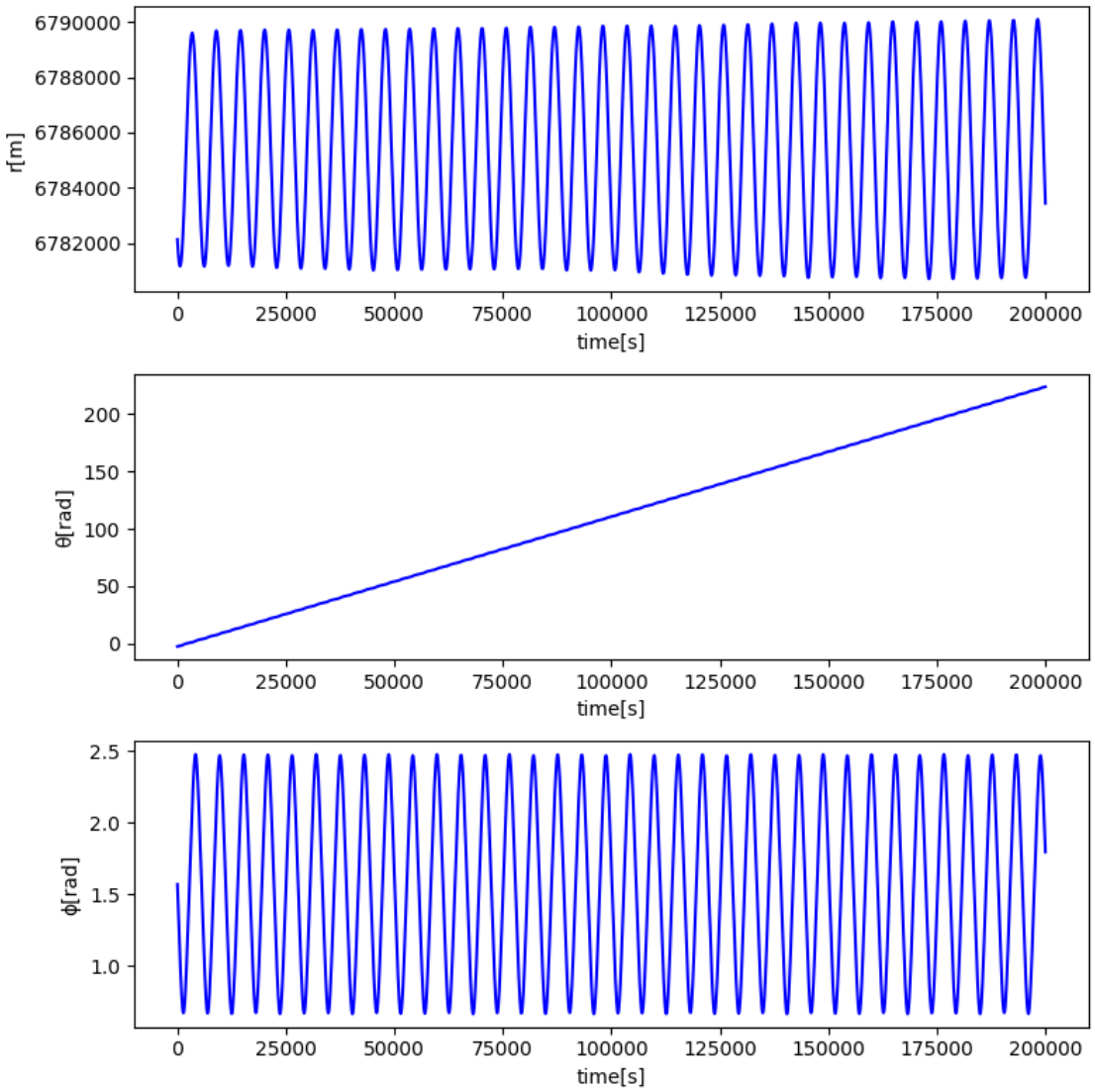


Figure 4.3: Variation of the spherical elements for the Controlled Orbit in function of time

The radial distance of the space vehicle from the center of the Earth is shown to be the same as the Keplerian Orbit with the use of the designed controller, however, like on the Real Orbit

there can't be any certainties towards the control of the angles that compose the spherical coordinates.

One way of finding out if the controller is working correctly is by measuring the error of the spherical coordinates of the Real Orbit from the Keplerian Orbit and compare to the one of the Controlled Orbit from the Keplerian Orbit.

Below, the error of the Real Orbit from the Keplerian Orbit is graphically represented:

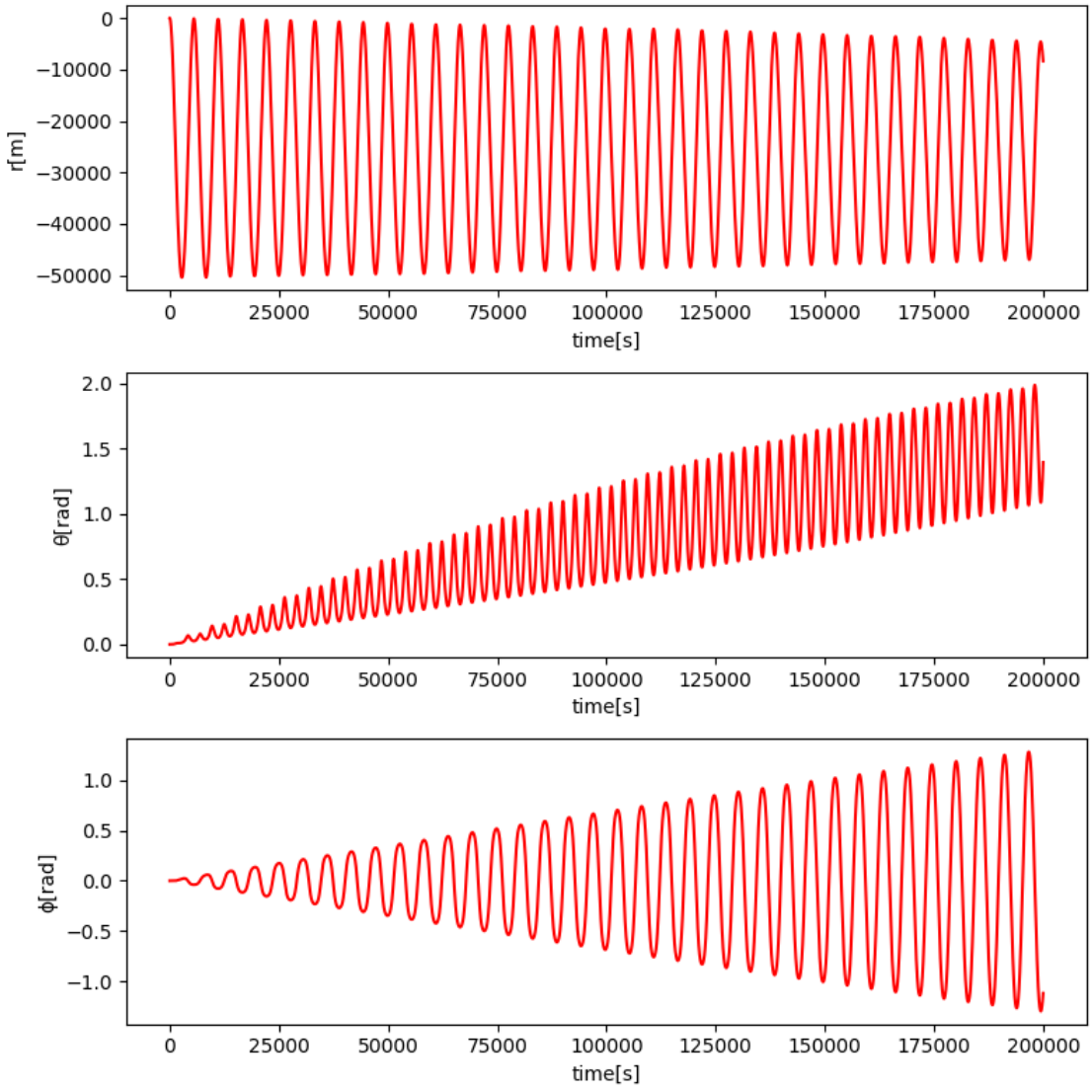


Figure 4.4: Variation of the error of the Real Orbit from the Keplerian Orbit in function of time

As foreseen, the error encountered for the radial distance of the space vehicle from the center of the Earth is troubling, however, and maybe not as expected, the deviations are

even more serious for the angles that compose the spherical coordinates because they are continuously increasing over time.

Now, for the error of the Controlled Orbit from the Keplerian Orbit:

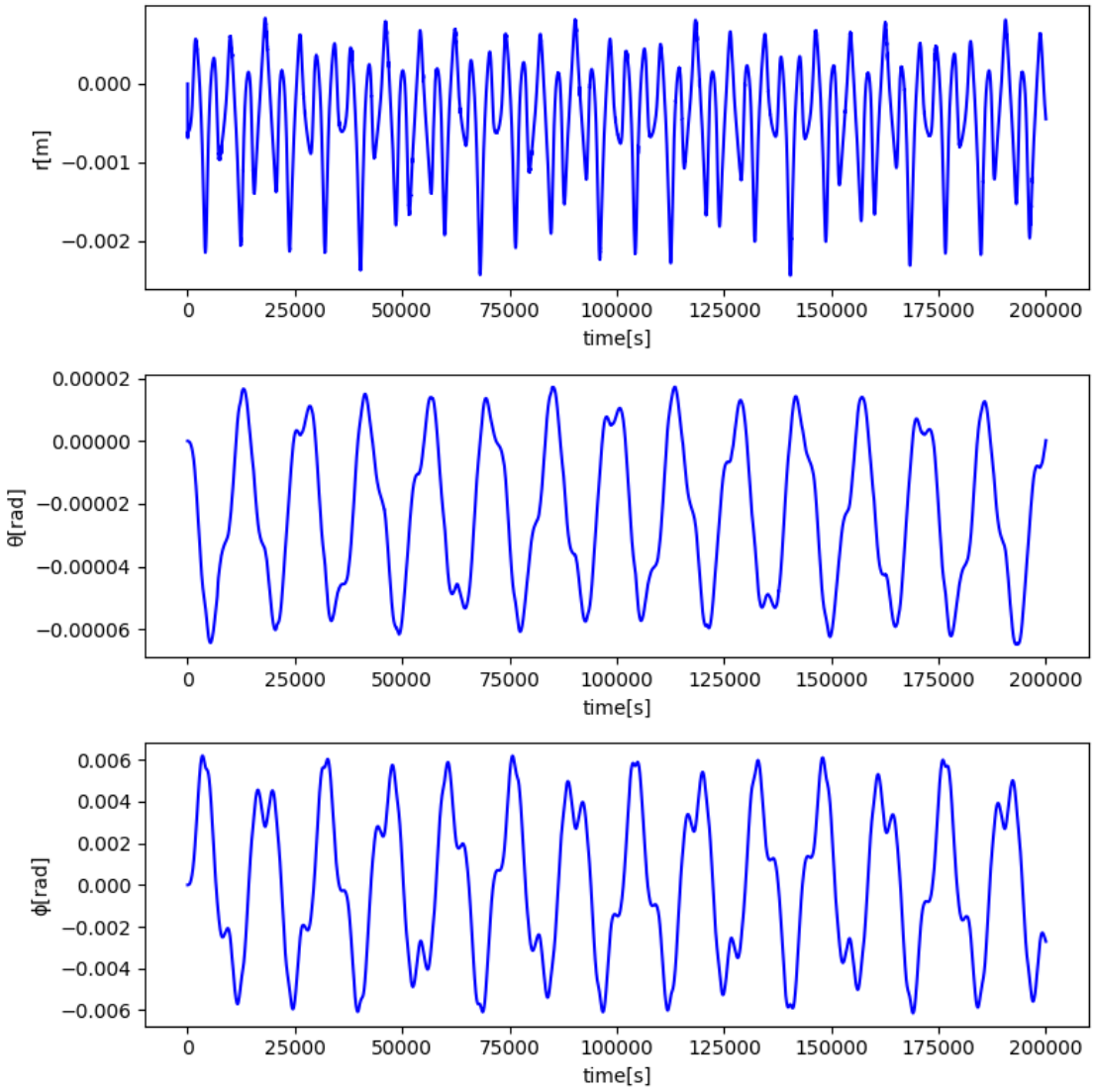


Figure 4.5: Variation of the error of the Controlled Orbit from the Keplerian Orbit in function of time

The error is proven to be insignificant, and, more important than that, it is converging to zero for all the elements composing the spherical coordinates. This proves that the controller is working properly and performing its duty adequately, even for a nonlinear system.

It is also fundamentally important to visualize how the trajectories are propagating in three dimensions to ensure that the dynamic system is properly describing a real case scenario and to compare the Controlled Orbit with the Keplerian Orbit in three dimensions.

After solving the ODEs for the Keplerian, Real and Controlled orbits, three-dimensional graphical representations can be obtained on cartesian coordinates by converting the spherical coordinates into cartesian coordinates for every output of the state of the system, as shown below [31]:

$$\begin{aligned} x_k &= r_k \sin\phi_k \cos\theta_k \\ y_k &= r_k \sin\phi_k \sin\theta_k \\ z_k &= r_k \cos\phi_k \end{aligned} \tag{4.21}$$

The following figure shows the Keplerian Orbit:

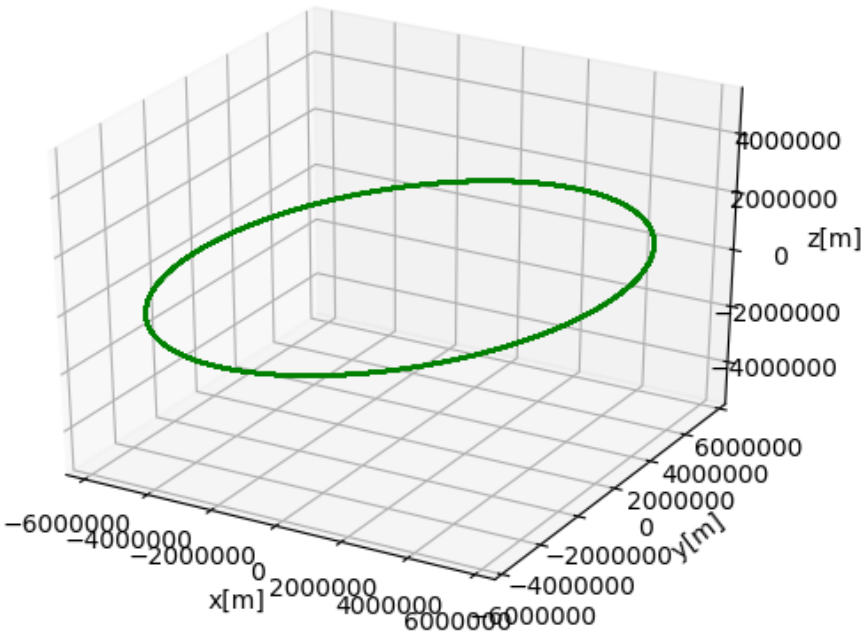


Figure 4.6: 3D Representation of the Keplerian Orbit

For this orbit, it does not matter what the simulation time is, as long as it is enough time for the ISS to perform a single revolution around the Earth, this graphical representation will always look the same, a single closed line. This is expected since a Keplerian Orbit only considers the gravitational forces of the two main bodies, Earth and ISS, and therefore a repetitive orbit is obtained.

The following figure shows the Real Orbit:

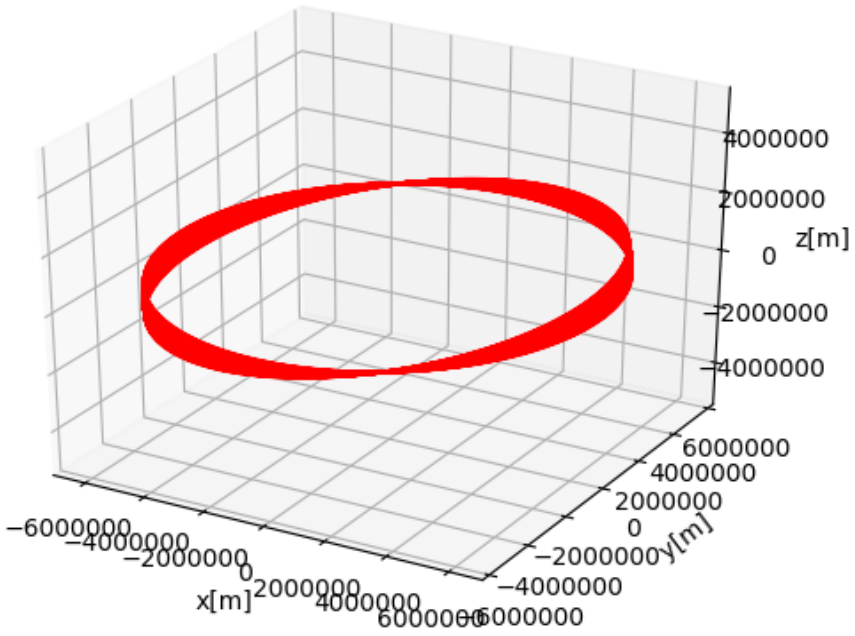


Figure 4.7: 3D Representation of the Real Orbit

This orbit shows clearly the effects of J_2 on a LEO in approximately 2.5 days. To determine whether the dynamic system used is properly representing a real case scenario, this orbit was compared to the one simulated in NASA's General Mission Analysis Tool (GMAT) with the same initial conditions. As already referred in chapter 2 of this dissertation, GMAT considers many other perturbations in space and other important factors like some characteristics regarding the spacecraft to simulate. Therefore, the orbit simulated in GMAT must have excellent accuracy when comparing to a real case scenario. Below, a representation of the Real Orbit in GMAT is displayed for the same simulation time:

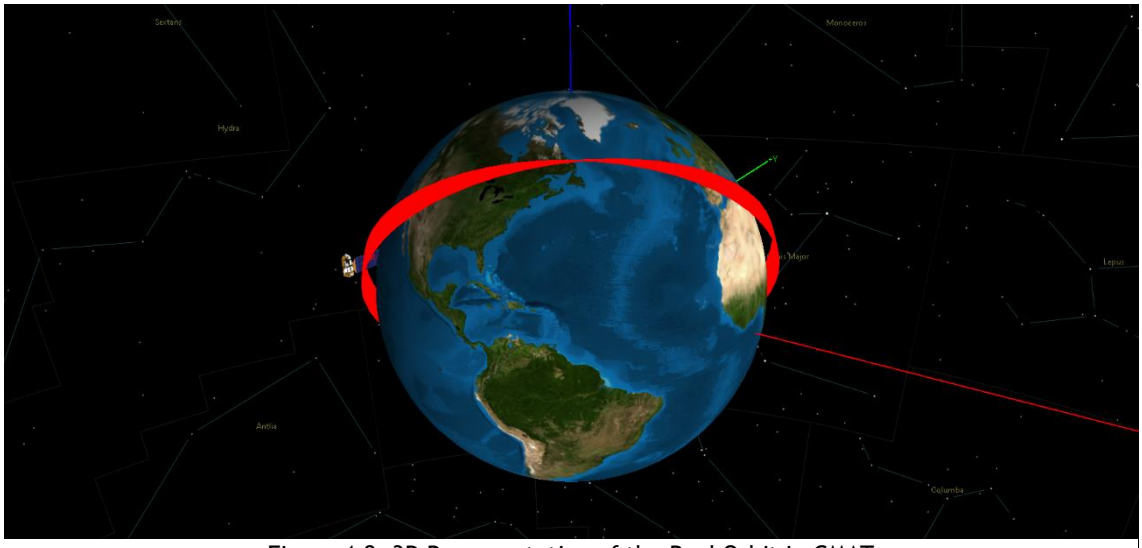


Figure 4.8: 3D Representation of the Real Orbit in GMAT

The representations of the Real Orbit in figure 4.7 and in figure 4.8 are identical, which proves that the dynamic system utilized for this dissertation present in equations (2.11) is a good representation of a real case scenario.

Even for just 2.5 days, approximately, the effects from the J_2 perturbation are clearly noticeable. For this orbit, the longer the simulation time, the heavier the deviation from the Keplerian Orbit, as represented on the following figure that demonstrates the effects of this perturbation for 9.25 days, approximately:

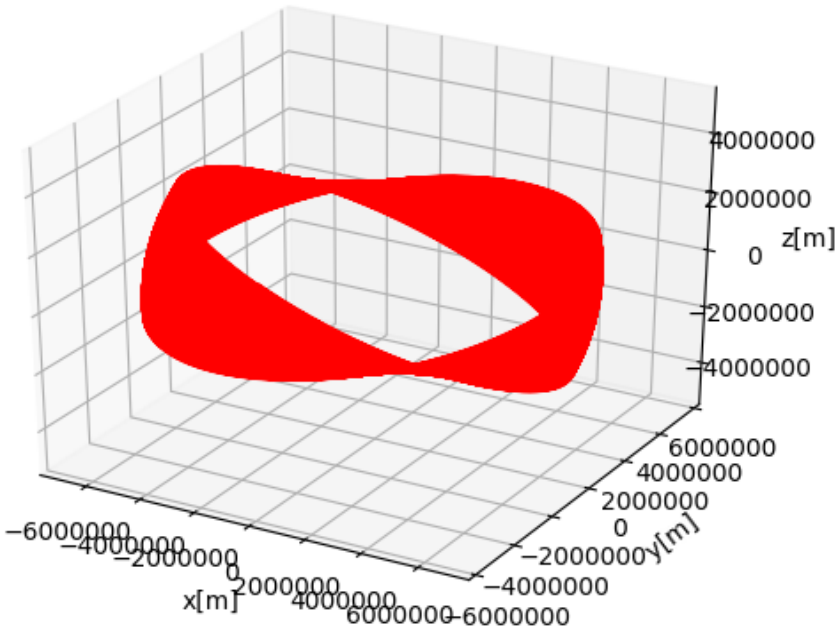


Figure 4.9: 3D Representation of the Real Orbit for 9.25 days

Finally, the following figure shows the Controlled Orbit:

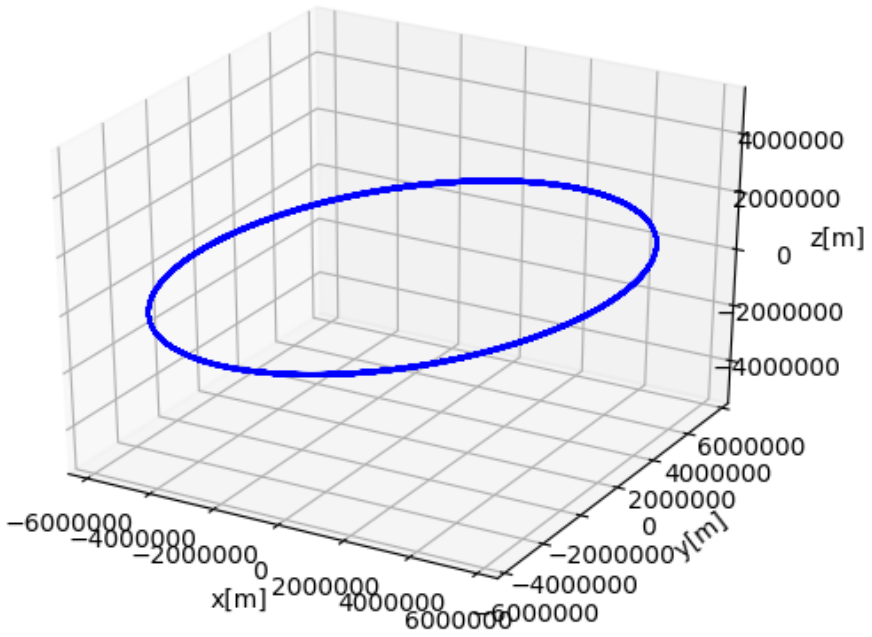


Figure 4.10: 3D Representation of the Controlled Orbit

As expected, the Controlled Orbit is identical to the Keplerian Orbit, which proves the analysis made from figure 4.5.

The designed controller is proven to be robust, because it is designed to work in real life situations. The whole system remains within chosen limits, because it successfully follows the desired trajectory and therefore, it is safe to say that the designed controller also possesses robust stability. Finally, because it can also track the desired state and rejects the considered perturbations it can also be said that the designed controller possesses robust performance.

It is also expected that the controller behaves well even if the system considers random errors or minor disturbances that happen to space vehicles in LEO.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Space vehicles provide important science data, but they are expensive to operate and launch, therefore it is important to control them in order to allow them to execute their missions properly. To do so, it is essential to find which orbit perturbations should be considered and what orbital parameters need to be controlled.

An H_∞ controller was designed in order to successfully alter the orbit of a space vehicle with perturbations, so that it follows a desired orbit with the same initial conditions. The controller showed great overall performance, it was proven to be robust, possess robust stability and robust performance.

For a LEO spacecraft, the only significant perturbations are the atmospheric drag and the Earth's oblateness and these affect the semimajor axis, the eccentricity, the right ascension of the ascending node and the argument of the perigee. The semimajor axis and the eccentricity can be corrected via Hohmann transfer orbits, while the right ascension of the ascending node and the argument of perigee can be adjusted by performing an impulse at the perigee along the radial direction in the orbital plane.

It was also understood that a control system for this case should be robust in order to minimize random errors and to force the space vehicle to follow a pre-determined ideal orbit with few deviations. The H_∞ controller provides the greatest robustness and could minimize the spacecraft's fuel consumption when executing corrective maneuvers compared to other controllers theoretically analyzed.

LEO space vehicles demand great computational power for two reasons. Firstly, an autonomous, on-board controller is preferred instead of the classical approach of sending commands through a ground station due to the characteristics of the orbit. Secondly, the saturation of the satellite's actuators and sensors is a risk when using H_∞ controllers if there is not enough processing power.

Bearing in mind the unstoppable technological advancement, processors are getting smaller, lighter and more powerful each day, therefore, it is expected that more and more LEO satellites utilize on-board H_∞ controllers.

5.2 Future Work

Although the developed controller shows great overall performance and robustness, it would be interesting to compare the obtained results to other control methods like the SMC for the same case. It would also be curious to develop a similar H_∞ controller using LMIs instead of Riccati equations.

An identical controller could be developed for other types of orbits that take other perturbations into account like the lunar-solar effects for geosynchronous altitudes.

Another controller could also be developed for orbits in other planets with different gravitational constants, values for J_2 and equatorial radiuses.

Bibliography

[1] J. R. Wertz, Space Mission Engineering: The New SMAD, 2nd ed. Space Technology Library, California, CA, USA: 2011.

[2] J. Zhang, K. Yang, R. Qi, S. Zhao, Y. Li, "Robustness analysis method for orbit control," in *Acta Astronautica*, vol. 137. Elsevier, 2017, pp. 15-24.

[3] Pixabay, "ISS." [Online]. Available: https://pixabay.com/pt/photos/iss-estação-espacial-11114/ [Accessed: 19-Aug-2019].

[4] C.-C. Chao, Applied Orbit Perturbation and Maintenance. The Aerospace Press, California, CA, USA: 2005.

[5] Leafnode, "Hohmann transfer orbit." [Online]. Available: https://en.wikipedia.org/wiki/Hohmann_transfer_orbit#/media/File:Hohmann_transfer_orbit.svg [Accessed: 27-Mar-2019].

[6] ESA - Space in Images, "The Earth's gravity field (geoid) as it will be seen by GOCE." [Online]. Available: https://www.esa.int/spaceinimages/Images/2004/10/The_Earth_s_gravity_field_geoid_as_it_will_be_seen_by_GOCE [Accessed: 16-Apr-2019].

[7] H. Curtis, Orbital Mechanics for Engineering Students. Elsevier Aerospace Engineering Series, Florida, FL, USA: 2005.

[8] W. J. Larson, J. R. Wertz, Space Mission Analysis and Design, 3rd ed. Space Technology Library, California, CA, USA: 1999.

[9] M. Moghadasian, J. Roshanian, "Approximately optimal manoeuvre strategy for aero-assisted space mission," in *Advances in Space Research* Engineering, vol. 64, no. 2. Elsevier, 2019, pp. 436-450.

[10] M. Willis, K. T. Alfriend, S. D'Amico, "Second-order solution for relative motion on eccentric orbits in curvilinear coordinates," in Proc. AAS/AIAA Astrodynamics Specialist Conference, 2019.

[11] J.-U. Park, K.-H. Choi, S. Lee, "Orbital rendezvous using two-step sliding mode control," in *Aerospace Science and Technology*, vol. 3, no. 4. Elsevier, 1999, pp. 239–245.

[12] Heavens Above, “ISS – Orbit.” [Online]. Available: https://www.heavens-above.com/orbit.aspx?satid=25544 [Accessed: 20-Feb-2019].

[13] H. Kwakernaak, R. Sivan, Linear optimal control systems, 1st ed. Wiley-Interscience, 1972.

[14] A. Iggidr, Controllability, Observability, and Stability of Mathematical Models. Encyclopedia of Life Support Systems (EOLSS), Mathematical Models, UNESCO, France: 2004.

[15] N. S. Nise, Control System Engineering, 6th ed. John Wiley & Sons, California, CA, USA: 2011.

[16] B. M. Chen, Robust & H_∞ Control. Springer-Verlag London, Singapore: 2000.

[17] N.I.A.B.M. Isa, “*DC Motor Controller Using Linear Quadratic Regulator (LQR) Algorithm Implementation on PIC,”* Bs.C. dissertation, Faculty of Electrical & Electronics Engineering, Universiti Malaysia Pahang, Malaysia, 2008.

[18] H. Wang, L. Liao, D. Wang, S. Wen, M. Deng, “Improved Artificial Bee Colony Algorithm and Its Application in LQR Controller Optimization,” in *Mathematical Problems in Engineering*. Hindawi, 2014.

[19] S. Florio, S. Amico, “Optimal Autonomous Orbit Control of Remote Sensing Spacecraft,” in *19th AAS/AIAA Space Flight Mechanics Meeting*. 2009.

[20] C. Chen, “*On the Robustness of the Linear Quadratic Regulator via Perturbation Analysis of the Riccati Equation,”* Ph.D. dissertation, School of Electronic Engineering, Dublin City University, Dublin, 2014.

[21] K. Ogata, Modern Control Engineering, 5th ed. Prentice Hall, 2010.

[22] J. P. Barbot, Sliding Mode Control in Engineering, 1st ed. Marcel Dekker, Cergy-Pontoise, France: 2002.

[23] L. Hui, L. Junfeng, H. Baoyin, “Sliding Mode Control for Low-Thrust Earth-Orbiting Spacecraft Formation Maneuvering,” in *Aerospace Science and Technology*, vol. 10, no. 7. Elsevier, 2006, pp. 636-643.

[24] L. E. Ghaoui, S. Niculescu, Advances in Linear Matrix Inequality Methods in Control. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA: 2000.

[25] L. Yu, J. Chu, “An LMI approach to guaranteed cost control of linear uncertain time-delay systems,” in *Automatica*, vol. 35, no. 6. Elsevier, 1999, pp 1155-1159.

[26] P. Massioni, T. Keviczky, E. Gill, M. Verhaegen, “A Decomposition-Based Approach to Linear Time-Periodic Distributed Control of Satellite Formations,” in *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3. IEEE, 2011, pp 481-492.

[27] K. Neokleous, “*Modeling and Control of a Satellite’s Geostationary Orbit”*, M.Sc. dissertation, Dept. of Space Science, Luleå University of Technology, Prague, 2007.

[28] S. A. Parvez, G-Q. Xing, “Autonomous Orbit Control with Position and Velocity Feedback Using Modern Control Theory,” US Patent 6089507. (1998).

[29] I. Yaesh, U. Shaked, “A Transfer Function Approach to the Problems of Discrete-Time Systems: H_∞-Optimal Linear Control and Filtering,” in *IEEE Transactions on Automatic Control*, vol. 36, no. 11. IEEE, 1991.

[30] S. C. Chapra, R. P. Canale, Numerical Methods for Engineers, 6th ed. McGraw-Hill Higher Education, Boston, Massachusetts, MA, USA: 2006.

[31] H. Vermeille, “Direct transformation from geocentric coordinates to geodetic coordinates,” in *Journal of Geodesy*, vol. 76. Springer, 2002, pp. 451-454.

Appendix

Publication

The research work realized on the production of this dissertation and the dissertation of another author resulted in the following paper:

Pedro M. Dente and Marco D. Martins, "H_∞ Orbital Control and Trajectory Estimation of a Space Vehicle on a Low Earth Orbit," 2019. (To be revised by the Supervisor).

H_∞ Orbital Control and Trajectory Estimation of a Space Vehicle on a Low Earth Orbit

Pedro M. Dente[a], Marco D. Martins[b], Kouamana Bousson[*]

Departamento de Ciências Aeroespaciais, Universidade da Beira Interior, Calçada Fonte do Lameiro, Covilhã 6201-001, Portugal

(Received keep as blank , Revised keep as blank , Accepted keep as blank)

Abstract. A Low Earth Orbit (LEO) is the trajectory of a space vehicle such as a satellite, missile or space shuttle relative to the Earth with an altitude of 1,500 km or less. Keplerian orbits only take in account the gravitational forces between bodies, however, in real life applications there are a lot of perturbations that alter the initial trajectory or the "mathematical ideal orbit" of the space vehicle. The perturbations analyzed in this paper were the atmospheric drag and the oblateness of the Earth (J_2). It is necessary to control this trajectory in order to adjust the orbital elements so that the space vehicle can operate in the surroundings of the initial orbit and perform its mission adequately. It is also of great importance to obtain an estimation of the trajectory as devoid as possible of any type of noise, or uncertainty. The focus of this paper is the development of an H_∞ robust controller that can successfully alter the trajectory of a space vehicle with perturbations, so that it follows a referenced Keplerian trajectory with the same initial conditions, as well as the development of two types of Kalman filters: robust and fading Kalman filters, to obtain a clean, noise free trajectory.

Keywords: atmospheric drag; fading Kalman filter; H_∞ controller; J_2; low Earth orbit; robust Kalman filter; space vehicle; trajectory estimation

1. Introduction

After a space vehicle reaches its desired orbit, minor adjustments must be done to compensate for perturbations so that the orbital elements remain within chosen limits and the spacecraft performs its mission correctly (Wertz 2011). Orbit perturbations change the initially calculated trajectory of a Keplerian orbit into a real orbit. The real orbit of a spacecraft, varying in time and space, is a sequence of osculating Keplerian orbits arranged in chronological order (Zhang *et al*. 2017). These orbit perturbations include the atmospheric drag, Earth's oblateness, solar winds, electromagnetic perturbations, the gravitational force of other celestial bodies (Sun, Moon, other planets and asteroids) and the solar radiation pressure, among others. Keplerian orbits are good approximations for most cases, with some exceptions like the atmospheric drag in Low Earth Orbit (LEO), the rotation of the ascending node and perigee due to Earth's oblateness, and lunar-solar effects at geosynchronous altitudes. This paper will focus on the specific case of a LEO, therefore the perturbations considered are the atmospheric drag and the Earth's oblateness (Wertz 2011).

[a] M.Eng. Student, E-mail: pedro.dente@hotmail.com
[b] M.Eng. Student, E-mail: elin-omar@hotmail.com

The propulsion system controls the space vehicle by modifying the magnitude and direction of its velocity vector. This requires a thruster burn of a specific duration, and often requires a prior attitude change maneuver to properly orient the thruster(s) utilized for trajectory control (Wertz 2011).

Even though the orbit parameters are well calculated, and the main perturbations are considered, random errors in the control system caused by inaccurate orbit determination (due to other less significant perturbations) and the thrust implementation device may lead to failure of the entire spacecraft mission. This means a huge amount of financial resources and valuable science data wasted, thus, it is essential to evaluate the robustness of the controller and to develop a way to keep a spacecraft in a previously assigned orbit through orbital maneuvers executed by thrusters (station-keeping strategy) that tolerate a somewhat large magnitude of errors (Zhang *et al*. 2017).

For these reasons, Kalman filters are used to filter and update the trajectory, and to immediately react when confronted with deviations of great importance. However, because of the delicate nature of spacecraft missions, there is a need for great precision in said filters, with a normal Kalman filter maybe be deemed too unprecise and simple for such role. This paper will implement two types of improved Kalman filters that will try to deal with the two more prevalent problems in trajectory filtering in order to greatly improve precision. The first problem being the uncertainties within the state transition and measurement matrices in the model system, and the second problem being the phenomenon of divergence: when the weight of past data begins to influence newer one, causing the estimated results to ever so diverge from the real measurements

These two problems will be solved respectively via a Kalman filter that is robust against uncertainty, and a fading Kalman filter that gives more prevalence to newer estimated data when faced with older one, with a normal Kalman filter being implemented for comparison.

1.1 Atmospheric Drag

When the orbit altitude is less than 1,500 km, air molecules in the Earth atmosphere interfere with the space vehicle in the direction of its motion. The force that the molecules exert on the spacecraft surface in the opposite direction of its motion is called atmospheric drag. The equation that determines the drag acceleration is:

$$\mathrm{a}_{\mathrm{D}} = -(1/2)\rho V^2 (C_D A/m)\mathbf{i}_v \tag{1.1}$$

where ρ is the atmospheric density; V is the velocity of the spacecraft relative to the atmosphere; C_D is the drag coefficient, a dimensionless quantity; A is the spacecraft effective projection area; m is the mass of the spacecraft; and $\mathbf{i}_v$ is the unit vector of the spacecraft velocity relative to the atmosphere.

Atmospheric drag is directed along the satellite's negative velocity relative to the atmosphere. It gradually decreases the mean orbit altitude and speeds up the satellite, therefore periodic maneuvers are needed to recover the semimajor axis of the satellite's decaying orbit. For simple drag-makeup maneuvers, a single-burn or a Hohmann transfer is performed to recover the orbit altitude when the semimajor axis reaches its lower limit (Chao 2005).

1.2 Earth's Oblateness

The shape of the blue planet is very similar to a sphere. But the centrifugal force due to its

rotation turns it into an oblate spheroid with flattening at the poles and an equatorial bulge (the equatorial radius is greater than the polar radius). At the equator, Earth, resembles a pear. Also, there are a lot of minor mass anomalies such as continents and mountains (Wertz 2011). This lack of symmetry implies that the center of mass of the Earth is not in its geometric center, therefore, the force of gravity on an orbiting body is not perfectly directed towards it. While the gravitational field of a perfectly shaped spherical planet depends only on the distance from its center, oblateness also causes a variation with the angular distance from the equator (latitude). This is called a zonal variation. The dimensionless parameter which quantifies the major effects of oblateness on orbit is J_2, the second zonal harmonic. J_2 is not a universal constant meaning that each planet has its own value depending on its oblateness, rotation rate and mass. For Earth, $J_2 = 1.082 \cdot 10^{-3}$, approximately (Curtis 2005).

The potential generated by the non-sphericity of the Earth causes periodic variations in all the orbital elements, especially in the right ascension of the ascending node (Ω) and the argument of perigee (ω). The variation of the right ascension of the ascending node and the argument of the perigee due to J_2 are:

$$\dot{\Omega}_{J_2} = -1.5 n J_2 (a_e / a)^2 (\cos i)(1 - e^2)^{-2} \tag{1.2}$$

$$\dot{\omega}_{J_2} = 0.75 n J_2 (a_e / a)^2 (4 - 5\sin^2 i)(1 - e^2)^{-2} \tag{1.3}$$

Where n is the mean motion in deg/day, a_e is the Earth's equatorial radius, a is the semimajor axis in km, e is the eccentricity, i is the inclination, $\dot{\Omega}$ and $\dot{\omega}$ are in deg/day.

The oblateness term, J_2, is one of the most significant perturbation acting on Geosynchronous Equatorial Orbit (GEO) and below, which includes altitudes lower than 36,000 km. This perturbation implies major impacts in the initial orbit mainly because of the rotation of several degrees of the right ascension of the ascending node and the argument of perigee per day (Larson and Wertz 1999).

Besides the rotation of the orbit, the principal effect of the non-spherical mass distribution is to cause small changes in the shape of the orbit. Instead of a perfect ellipse, the space vehicle moves in an irregular one, with maximum deviations from a Keplerian orbit on the order of 3 km. In a repeating ground-track orbit, the satellite flies over the same location on the ground, day after day, and it is possible for resonances to build up when the satellite continuously passes (Wertz 2011). It is fundamental to control the spacecraft's trajectory by avoiding these kinds of heavy deviations especially if a repeating ground-track orbit is desired so that the space vehicle can realize its mission properly.

1.3 Orbit Control

There are six orbital elements that define the orbit of a space vehicle. These are: a, semimajor axis, e, eccentricity, i, inclination, Ω, right ascension of the ascending node, ω, argument of the perigee and v, true anomaly.

Only some elements require control for most space missions. True anomaly never needs control because it only defines the current location of the spacecraft of a given orbit. Of the other five elements, the semimajor axis and eccentricity are the ones that require maintenance first and the remaining need control depending on the mission. For LEO, as explained, the main perturbations are the atmospheric drag and the J_2 perturbation. These do not interfere directly with the inclination, however: - The semimajor axis and eccentricity decrease continuously because of the atmospheric drag; - The right ascension of the ascending node and the argument of the perigee may vary several degrees per day depending on the inclination and altitude because of the J_2 perturbation. Therefore, only a, e, Ω and ω require control for this case.

The orbit elements of the reference orbit are denoted by a*, *e**, *i**, Ω*, ω*, and *v**, and the osculating elements are a(t), *e*(t), *i*(t), Ω(t), ω(t), and *v*(t). Then, the deviations of the orbit elements (or the associated error) can be written as $\delta a = a(t) - a^*$, $\delta e = e(t) - e^*$, $\delta\Omega = \Omega(t) - \Omega^*$ and $\delta\omega = \omega(t) - \omega^*$. The orbit elements maximum acceptable variation intervals are defined as Ia, I*e*, IΩ and Iω and its values are attributed before the controller starts working.

I*e* is omitted because its variation is very small, and a circular orbit is considered ideal (the atmospheric drag helps achieving $e = 0$). Also, every time a Hohmann transfer is executed because of the correction of the semimajor axis, the orbit becomes circular even if it is not for a long time. Therefore, it is pointless to define the maximum acceptable variation interval for the eccentricity.

When $\delta a(t) \in Ia$ is violated, a Hohmann transfer is executed to correct the semimajor axis and eccentricity simultaneously. When $\delta\Omega(t) \in I\Omega$ or $\delta\omega(t) \in I\omega$ is violated, an impulse will be performed at the perigee along the radial direction in the orbital plane to correct either the right ascension of the ascending node or the argument of the perigee, depending on which element is required to be corrected (Zhang *et al.* 2017).

To simplify this concept, we can imagine a trajectory tube, where the acceptable deviations from the reference trajectory (ideal Keplerian orbit) are contained, see Fig. 1 (Moghadasian and Roshanian 2019). If the spacecraft trespasses this tube, the designed controller will activate one or more thrusters in order to place the space vehicle back on this imaginary tube.

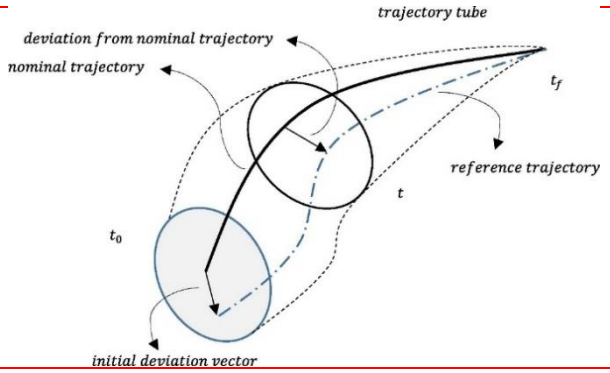


Fig. 1 Conceptual presentation of the trajectory tube (Moghadasian and Roshanian 2019)

2. Dynamic Model of Space Vehicles for Earth Orbits

The equations of motion for a space vehicle revolving around the Earth in a Keplerian orbit, can be given in a spherical coordinate system, represented in Fig. 2, as (Park *et al*. 1999):

$$\begin{cases} \ddot{r} = r\dot{\theta}^2 \sin^2\phi + r\dot{\phi}^2 - \dfrac{\mu}{r^2} + u_r \\ \ddot{\theta} = \dfrac{-2\dot{r}\dot{\theta}}{r} - 2\dot{\theta}\dot{\phi}\cot\phi + \dfrac{u_\theta}{r\sin\phi} \\ \ddot{\phi} = \dfrac{-2\dot{r}\dot{\phi}}{r} + \dot{\theta}^2 \sin\phi\cos\phi + \dfrac{u_\phi}{r} \end{cases} \tag{2.1}$$

Where μ is the Earth's gravitational constant, r is the radial distance of the space vehicle from the center of the Earth, θ is the azimuth angle measured from the X-axis in the XY-plane, and ϕ is

the inclination measured from Z-axis to the vector r. u_r, u_θ and u_ϕ are the trust acceleration components in the i_r, i_θ and i_ϕ directions, respectively. The space vehicle is modeled as a point mass.

Since the perturbations being analyzed in this paper are the atmospheric drag and the Earth's oblateness and a simple, yet effective solution was given to the first one, it is necessary to solve the problem for an orbit under the effects of the second zonal harmonic, J_2. The equations of motion can be rewritten, in order to consider these effects as presented below (Park *et al*. 1999):

$$\begin{cases} \ddot{r} = r\dot{\theta}^2 \sin^2\phi + r\dot{\phi}^2 - \dfrac{\mu}{r^2} + \dfrac{3}{2}\mu J_2 a_e^2 \dfrac{3\cos^2\phi - 1}{r^4} + u_r \\ \ddot{\theta} = \dfrac{-2\dot{r}\dot{\theta}}{r} - 2\dot{\theta}\dot{\phi}\cot\phi + \dfrac{u_\theta}{r\sin\phi} \\ \ddot{\phi} = \dfrac{-2\dot{r}\dot{\phi}}{r} + \dot{\theta}^2 \sin\phi\cos\phi + 3\mu J_2 \dfrac{a_e^2}{r^5}\cos\phi\sin\phi + \dfrac{u_\phi}{r} \end{cases} \tag{2.2}$$

Where μ is $3.986 \cdot 10^{14}$ $m^3 \cdot s^{-2}$, J_2 is $1.082 \cdot 10^{-3}$ and a_e is the equatorial radius of the Earth, $6{,}378 \cdot 10^3$ m approximately.

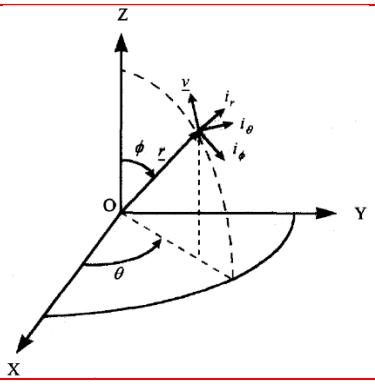


Fig. 2 Motion of a space vehicle orbiting Earth in the spherical coordinate system (Park *et al*. 1999)

2.1 Initial Conditions

Eqs. (2.1) and (2.2) can be rewritten using state space notation, which doubles the number of equations, but transforms them from second order differential equations to first order differential equations. State space notations attributes a new variable, x_n, for each variable from the spherical coordinate system:

$$\begin{cases} x_1 = r \\ x_2 = \dot{r} \\ x_3 = \theta \\ x_4 = \dot{\theta} \\ x_5 = \phi \\ x_6 = \dot{\phi} \end{cases} \tag{2.3}$$

Eq. (2.4) are the state space form of Eq. (2.1) and Eq. (2.5) are the state space form of Eq. (2.2), as seen below:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 x_4{}^2 \sin x_5{}^2 + x_1 x_6{}^2 - \dfrac{\mu}{x_1{}^2} + u_{x1} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -\dfrac{2x_2 x_4}{x_1} - 2x_4 x_6 \cot x_5 + \dfrac{u_{x3}}{x_1 \sin x_5} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -\dfrac{2x_2 x_6}{x_1} + x_4{}^2 \sin x_5 \cos x_5 + \dfrac{u_{x5}}{x_1} \end{cases} \tag{2.4}$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_1 x_4{}^2 \sin x_5{}^2 + x_1 x_6{}^2 - \dfrac{\mu}{x_1{}^2} + \dfrac{3}{2}\mu J_2 a_e{}^2 \dfrac{3\cos x_5{}^2 - 1}{x_1{}^4} + u_{x1} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = -\dfrac{2x_2 x_4}{x_1} - 2x_4 x_6 \cot x_5 + \dfrac{u_{x3}}{x_1 \sin x_5} \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -\dfrac{2x_2 x_6}{x_1} + x_4{}^2 \sin x_5 \cos x_5 + 3\mu J_2 \dfrac{a_e{}^2}{x1^5} \cos x_6 \sin x_6 + \dfrac{u_{x5}}{x_1} \end{cases} \tag{2.5}$$

Proper initial conditions must be given so that the controller can be tested for a real case scenario. These are the ones mentioned in Eq. (2.3) and the chosen vehicle will be the International Space Station (ISS), because it fits with the requirements for this paper, LEO, and because of its great importance for the scientific community.

According to the DLR (German Aerospace Center), on the 20th of February of 2019 at 06:19:32 am (UTC), the orbit data of the ISS was (Heavens Above 2019):

ISS Orbit Data

Eccentricity e	0.0005800
Inclination i	0.9013 rad
Perigee height h_p	403,000 m
Apogee height h_a	411,000 m
Right ascension of the ascending node Ω	3.886 rad
Argument of the perigee ω	0.7564 rad
Mean anomaly M	5.530 rad

The variables from the table above were converted into spherical coordinates. So, the following initial conditions were found for the orbit of the ISS for the spherical state:

Initial Conditions for the Orbit of the ISS in Spherical Coordinates

r	6782,000 m
$\dot{r}$	-3.049 m·s^{-1}
θ	-2.396 rad
$\dot{\theta}$	7.016·10^{-4} rad·s^{-1}
ϕ	1.569 rad
$\dot{\phi}$	-8.865·10^{-4} rad·s^{-1}

3. Controller Design

This paper exposes a motion control problem, where the most important scenario to understand is the Trajectory Tracking. In this scenario, the output of the system, *y(t)* is controlled over time in order to oblige a vehicle to follow a pre-determined and desired trajectory, $y^*(t)$. In this case, the ISS orbit is controlled over time in order to follow a desired trajectory (the Keplerian Orbit), which is used as a reference signal.

3.1 Linear System

Ordinary Differential Equations (ODEs), can model equations of motion and control systems through the state-space approach, like in Eq. (2.4) and (2.5). These can also be described linearly as a state differential equation:

$$\dot{x}(t) = f[x(t), u(t), t] \tag{3.1}$$

Where $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector, u(t) $\in \mathbb{R}^p$ is the input or control variable, also given by a varying vector, f is a real function and t is the time variable [11].

The output variable $y(t) \in \mathbb{R}^q$ of the system can be observed and is given by the following equation:

$$y(t) = g[x(t), u(t), t] \tag{3.2}$$

Where *g* is a real function.

The system described with (3.1) and (3.2) gives the usual linear continuous-time system in the state-space form, also known as a finite-dimensional linear differential system and is represented by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3.3}$$

Where $A \in \mathbb{R}^{n \times n}$ is the system matrix, $B \in \mathbb{R}^{n \times p}$ is the input matrix, $C \in \mathbb{R}^{q \times n}$ is the output matrix and $D \in \mathbb{R}^{q \times p}$ is the feedforward matrix.

3.2 Stability, Controllability and Observability

It is fundamental to determine whether a dynamic system is stable, controllable and observable in the evaluation of a dynamic linear system.

A system is stable if it always returns to an equilibrium point after small disturbances. A system

is marginally stable if it has non attractive characteristics but, despite the circumstances it is still considered stable (Iggidr 2004).

Without appropriate actuators the system can't be properly controllable and without proper sensors, the right states can't be measured and therefore the system can't be properly observed.

3.2.1 Stability

A usual method to verify the stability of a system is to analyze the eigenvalues of its *A* matrix.

The system is linearly stable if all the real parts of its eigenvalues are less than zero, $Re(\lambda) < 0$.

The system is marginally stable if its *A* matrix is critical. The eigenvalues of a critical matrix are all composed of non-positive real parts except for at least one part equal to zero (Iggidr 2004).

3.2.2 Controllability

The dynamic system is controllable if and only if an input can be found that takes every state variable from a desired initial state to a desired final state in a finite amount of time (Nise 2011).

For the system to be controllable, the rank of the controllability matrix C_o needs to be equal to *n*.

$$C_o = [B \; AB \; A^2B \; ... \; A^{n-1}B] \tag{3.4}$$

3.2.3 Observability

The dynamic system is observable if and only if the state of a system *x(t)* can be determined given the input *u(t)* and output *y(t)* without knowing the initial state $x(t_0)$ (Nise 2011).

For the system to be observable, the rank of the observability matrix *O* needs to be equal to *n*.

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3.5}$$

3.3 Control Method

The property that a control system must possess for it to operate properly in realistic situations is defined as robustness. If a controller can be designed such that the whole system to be controlled remains stable when its parameters vary within certain expected limits, the system is said to possess robust stability. Besides, if it can satisfy performance specifications such as steady state tracking, disturbance rejection and speed of response requirements, it is said to possess robust performance (Chen 2000).

As understood in previous chapters, a LEO control system for a spacecraft should not tolerate significant trajectory deviations nor a significant range of random errors. There is no such thing as a perfect system, however, for the presented problem, it is wise to choose a robust controller that is optimized to reduce all kinds of uncertainties because a space vehicle is astronomically costly due to development, fabrication, launch, maintenance and therefore the risks taken should be minimum.

3.3.1 H_∞ Controller

The H_∞ controller contains a system, $\sum$, and its state-space representation. The system has two inputs: *w*, which refers to the disturbances and *u*, which refers to the controlled input. *u* is dependent of the controller, Control and Monitoring Processor (CMP), $\sum_{cmp}$, and the new measurement output, *y*. The two outputs of the system are *h*, that consists of the error signals to be minimized and *y*. A schematic representation of the H_∞ controller is represented in Fig. 6 (Chen 2000).

The standard H_∞ optimization problem is to solve an internally stabilizing proper measurement feedback control law, $\sum_{cmp}$, such that the H_∞ norm of the overall closed-loop transfer matrix function from *w* to *h* is minimized (Chen 2000). This means that we are defining the norm as a measure of the worst-case gain in terms of energy that the system can have (Massioni 2011). If a controller is designed to work in the worst-case scenario it is expected that it will have an excellent overall performance in regular conditions.

When analyzing the robustness of a system, the H_∞ controller is reliable even considering a high noise environment and it could potentially lead to less fuel consumption when executing space maneuvers (Neokleous 2007).

H_∞ enhances the control system performance by minimizing the control error and control effort. It also provides precise control performance and high robustness to model uncertainties and external disturbances. Signals from GPS satellites are used to directly or indirectly determine the satellite orbital state, and is used in real time feed-back loop to continuously estimate the error between a satellite and an ideal pre-defined orbit. For the specific cases of geostationary and LEO satellites, H_∞ provides the greatest robustness to the feedback control loop, increases the operating range of the system and enhances the convergence and stability of the control loop (Parvez and Xing 1998).

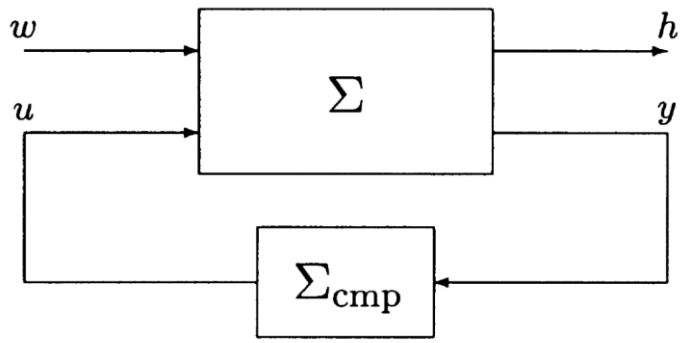


Fig. 3 The structure of the H_∞ control system (Chen 2000)

The linear continuous-time system in the state-space form of an H_∞ controller is given below:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ew(t) \\ y(t) &= Cx(t) \\ x(t_0) &= x_0 \end{aligned} \tag{3.6}$$

Where $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times p}$ is the input matrix, $C \in \mathbb{R}^{q\times n}$ is the output matrix and $E \in \mathbb{R}^{n\times p}$ is the disturbance matrix.

And where $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector, $y(t) \in \mathbb{R}^q$ is the output variable, $u(t) \in \mathbb{R}^p$ is the input or control variable, $w \in \mathbb{R}^p$ is the disturbance vector and x_0 is the state vector referent to the initial conditions.

The D matrix, in equation (3.3), for this paper, is a zero matrix because the system model does not need direct feedthrough.

The system should be stable or marginally stable, controllable and observable.

The first step to develop an H_∞ controller is to linearize the state-space system. This paper uses the one represented in Eq. (2.5).

In the state-space notation of a linear system, the A matrix is given by the Jacobian Matrix, J, of a non-linear system, excluding control or disturbance variables. Finally, the A matrix will be the Jacobian matrix with x_1, x_2, x_3, x_4, x_5 and x_6 equal to their respective equilibrium points.

The B, C and E matrices are composed of the information found on the non-linear system's variables for control, outputs and disturbance respectively.

A discrete-time approach should be considered so that the controller might be implemented on a real spacecraft and perform its tasks according to its digital devices. Therefore, the A, B, C, D and E matrices need to be discretized into A_d, B_d, C_d, D_d, E_d.

The A_d matrix should be nonsingular and for this to occur, the following condition must be secured (Yaesh and Shaked 1991):

$$\det(A_d) \neq 0 \tag{3.7}$$

The objective of the H_∞ controller is to minimize the norm of the transfer function matrix T_{yw} from w to y, given by (Yaesh and Shaked 1991):

$$T_{yw} = C(zI - A + BK)^{-1}E \tag{3.8}$$

Where z is the z-transform, which can be considered the discrete-time equivalent of the Laplace transform.

However, since it is difficult to achieve this goal, the minimum possible H_∞-norm of T_{yw} by γ_0 is denoted and an approximated γ is estimated such that $\gamma - \gamma_0 > 0$ is less than a prespecified tolerance. Also, u_k is determined so that the following condition is true (Yaesh and Shaked 1991):

$$\left\| T_{yw} \right\|_\infty < \gamma \tag{3.9}$$

For this paper the standard value, $\gamma = 1$, is assumed.

The optimal control problem determines the feedback gain, K matrix, of the optimal control vector:

$$u_k = -Kx_k \tag{3.10}$$

A stabilizing solution to (3.9) is found if there exists $P > 0$ that satisfies the following algebraic Riccati equation (Yaesh and Shaked 1991):

$$P = A_d^T P A_d - A_d^T P B_d (I + B_d^T P B_d)^{-1} B_d^T P A_d + P E_d (I + E_d^T P E_d)^{-1} E_d^T P + C_d^T C_d \tag{3.11}$$

P is estimated iteratively. Its initial state, P_0, is an identity matrix of appropriate size and with each iteration, a better solution is found for the problem. The frobenius norm of the difference after every iteration, $\|P_{k+1} - P_k\|_{\mathrm{F}}$, should become lower and lower, however if this is not the case and a convergence problem is encountered, Eq. (3.11) should be rewritten in to:

$$P = P\lambda^k + (1-\lambda^k)(A_d^T P A_d - A_d^T P B_d (I + B_d^T P B_d)^{-1} B_d^T P A_d + P E_d (I + E_d^T P E_d)^{-1} E_d^T P + C_d^T C_d) \tag{3.12}$$

Where λ is an attributed coefficient and should be $]0.5, 1[$ and $k \in \mathbb{N}^0$ is the number of the iteration. This is called a Mann fixed-point iteration.

If there exists a positive definite solution P to (3.11) or (3.12), then (3.9) is guaranteed by the following stabilizing gain matrix (Yaesh and Shaked 1991):

$$K = (I + B_d^T P B_d)^{-1} B_d^T P A_d \tag{3.13}$$

For a trajectory tracking problem, a reference signal must be given to the system so that a desired trajectory is followed. For that to happen, the desired state of the system, x_k^* , should be known for every instant of time that the controller is performing its duty. Eq. (3.10) can be rewritten in order to take this factor into account as seen below:

$$u_k = -K(x_k - x_k^*) \tag{3.14}$$

The control vector is calculated in a way so that the error of the system, $e = x_k - x_k^*$, is minimized over time.

Fig. 4 represents the structure of the designed H_∞ control system, based on Fig. 2.

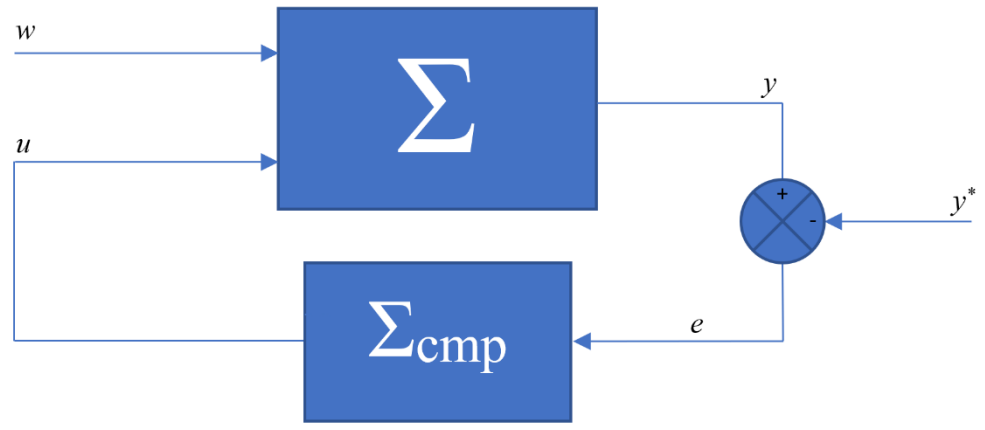


Fig. 4 The structure of the designed H_∞ control system

The structure of the designed H_∞ control system contains the nonlinear system, $\sum$, the controller, $\sum_{cmp}$, two inputs, w and u, one output, y, and the output of the desired state, y^*. The disturbances are given by the vector w, the controlled input is given by the vector u, which is dependent of the controller, $\sum_{cmp}$, and the new measurement output, y. This output, collectively with the desired state output and respective state vectors, x and x^*, will give the current error, which, along with the gain matrix K, will be used to calculate the control vector, u, for every step, dt.

4. Kalman Filter Design

As been said in previous chapters, the implementation of a proper Kalman filter is of great importance when dealing with trajectories that require a high degree of precision. Spacecraft trajectories are known to require greater degrees of precision and stability, such that a normal Kalman filter may not be enough to guarantee reliable and satisfying results.

There is a need, then, for more reliable and precise filters. This paper will classify the problems that can affect the performance of a normal Kalman filter in two different classes: first the uncertainties that exist within the state transition and measurement matrices, and secondly the loss of precision that occurs both with the linearization of more complex systems, as well as when new estimations are confronted with older ones.

Both of these problems will be explained with more detail in their respective sections.

Since all the developed filters will have the normal Kalman filter as their base, its structure will first be considered.

Consider then that the object of study, the ISS, follows a linear and discrete-time system, as described by the following equations for the state transition and measurement, respectively:

$$\begin{aligned} \hat{x}(k+1) &= Fx(k) + w(k) \\ y(k) &= Hx(k) + v(k) \end{aligned} \tag{4.1}$$

$x(k) \in \mathbb{R}^n$ being the state vector, $x(k+1) \in \mathbb{R}^n$ its estimated state, $F \in \mathbb{R}^{n \times n}$ the state transition matrix, $H \in \mathbb{R}^{n \times m}$ the measurement matrix, and $w(k)$ and $v(k)$ being measurement noise vectors with zero mean Gaussian, the covariant matrices being $Q(k) \in \mathbb{R}^{n \times n}$ and $R(k) \in \mathbb{R}^{m \times m}$ respectively.

To the above system will be applied a normal Kalman filter, which will consist of two equally important stages: prediction and update.

The prediction stage will consist of the equations for the predicted state estimate and the predicted error covariance, whose equations are respectively:

$$\begin{aligned} \hat{x}(k)^- &= Fx(k-1)^+ K(k)[y(k) - Hx(k)^+] \\ P(k)^- &= FP(k-1)^+ F^T + Q \end{aligned} \tag{4.2}$$

The update stage will consist of the equations for the updated state estimate and updated covariance error, respectively:

$$\begin{aligned} \hat{x}^+(k) &= \hat{x}^-(k) + K(k)y(k) \\ P^+(k) &= (I - K(k)H)P^-(k) \end{aligned} \tag{4.3}$$

Where K_k is the Kalman gain matrix, obtained by the following equation:

$$K(k) = P(k)^- H^T (R + HP(k)^- H^T)^{-1} \tag{4.4}$$

And $\tilde{y}_k$ is the residual measurement, serving as a comparison between the real measurement y_k and the estimated value:

$$\tilde{y}(k) = y(k) - H\hat{x}(k)^- \tag{4.5}$$

While the normal Kalman filter can be considered incomplete when dealing with spacecraft trajectories, the following filters will be built upon the above equations, with modifications being made in order to improve performance.

4.1. Robust Kalman Filter

The first filter being implemented is the robust Kalman filter.

When constructing a Kalman filter, it is usually assumed the state transition and measurement matrices to be either exactly known, or at least known with some degree of accuracy that will not have a great impact in the overall accuracy of the filter. For filters aiming for better accuracy and performance, that may not be entirely accurate.

Therefore, one can improve the performance of a Kalman filter by guarantying its robustness against plant uncertainty.

For this paper, the uncertainty will be defined as a norm bound parameter within both matrices F and H, with the variance of the filtering error being within certain limits for all possible parameter uncertainties. Matrices Q and R are considered to be known matrices that do not evolve with each iteration:

This paper will follow the equations in (Xia et al. 1994), for the implementation of the Robust Kalman Filter.

Thus, the new equations for state transition and measurement, now accounting for uncertainty become:

$$\begin{aligned} \hat{x}(k+1) &= [F + \Delta F(k)]x(k) + w(k) \\ y(k) &= [H + \Delta H(k)]x(k) + v(k) \end{aligned} \tag{4.6}$$

Where F and H are the known, certain and constant parts of the state transition and measurement matrix, respectably, and ΔF_k and ΔH_k are unknown matrices that represent the uncertainties in matrices F and H, varying with each iteration, following the equation bellow:

$$\begin{bmatrix} \Delta F(k) \\ \Delta H(k) \end{bmatrix} = \begin{bmatrix} H1 \\ H2 \end{bmatrix} M(k)E \tag{4.7}$$

M(k) being an unknown matrix satisfying the following condition:

$$M(k)^T * M(k) \le I \tag{4.8}$$

With k=0, 1, 2…

And *H1*, *H2* and *E* being known, constant matrices that define how the elements of the state transition and measurement matrices are affected by the unknown matrix *M*.

In order to use this model correctly, several assumptions will have to be made:

Firstly, it is necessary that the state transition matrix to be quadratically stable, that is, for a symmetric positive definite matrix P, the following inequation is true:

$$[F + \Delta F(k)]^T P[F + \Delta F(k)] - P < 0\text{, k=0, 1, 2} \tag{4.9}$$

Secondly, the covariant error Q and R are assumed to be nonnegative, and matrix F is assumed to be inverted.

With all assumptions being satisfied, the true goal of the robust Kalman filter can now be accomplished, which is, to create an estimator for newer states with the form:

$$\hat{x}(k+1) = \hat{F}\hat{x}(k) + K(y(k) - \hat{H}\hat{x}(k)) \tag{4.10}$$

Where $\hat{F}$ is the updated state transition matrix, K is the Kalman gain, and $\hat{H}$ is the updated measurement matrix, whose equations will be given at the end of this sub-chapter.

In order to find a stable estimator, it is necessary to first solve two algebraic Riccati equations (ARE):

The first ARE will be solved for matrix $P=P^T>0$,

$$F^T PF - P + F^T P\bar{Q}^{\frac{1}{2}}(I - \bar{Q}^{\frac{1}{2}}P\bar{Q}^{\frac{1}{2}})^{-1}\bar{Q}^{\frac{1}{2}}PF + \varepsilon(E^T E + \nu I) = 0 \tag{4.11}$$

Where ε and v are small, positive and real scalars, and $\bar{Q}$ is obtained by the following equation:

$$\bar{Q} = Q + \frac{1}{\varepsilon}H_1 H_1^T \tag{4.12}$$

If the solution P is a stable solution, then the following inequation is proven true:

$$I - \bar{Q}^{\frac{1}{2}}P\bar{Q}^{\frac{1}{2}} > 0 \tag{4.13}$$

With the first ARE being solved, one can now calculate the following updated matrices, with the uncertainties being taken in consideration

State transition $\hat{F}$ and measurement $\hat{H}$:

$$\begin{aligned} \hat{F} &= F + \bar{Q}(P^{-1} - \bar{Q})^{-1}F \\ \hat{H} &= H + \frac{1}{\varepsilon}H_2 H_1^T (P^{-1} - \bar{Q})^{-1}H_1 H_2^T \end{aligned} \tag{4.14}$$

Covariance matrix of the state transition noise $\hat{Q}$ and measurement noise $\hat{R}$:

$$\hat{Q} = \bar{Q} + \bar{Q}(P^{-1} - \bar{Q})^{-1}\bar{Q} \tag{4.15}$$

$$\hat{R} = R + \frac{1}{\varepsilon}H_2H_2{}^T + \frac{1}{\varepsilon^2}H_2H_1{}^T(P^{-1} - \bar{Q})^{-1}H_1H_2{}^T \tag{4.16}$$

Cross-variance matrix between the state transition and measurement noises M:

$$M = \frac{1}{\varepsilon}(I - \bar{Q}P)^{-1}H_1H_2{}^T \tag{4.17}$$

With these update values, one can now solve the second ARE for $U=U^T>0$

$$\hat{F}U\hat{F}^T - U - (\hat{F}U\hat{H}^T + M)(\hat{R} + \hat{H}U\hat{H}^T)^{-1}(\hat{F}U\hat{H}^T + M)^T + \hat{Q} = 0 \tag{4.18}$$

Thus, Kalman gain can now be calculated by the following expression:

$$K = (\hat{F}U\hat{H}^T + M)(\hat{R} + \hat{H}U\hat{H}^T)^{-1} \tag{4.19}$$

Which then allows for the estimator to be calculated.

$$\hat{x}(k)^- = Fx(k-1)^+ K(k)[y(k) - Hx(k)^+] \tag{4.20}$$

4.2. Fading Kalman Filter

Another problem that threatens the accuracy and performance of the Kalman filter is the possible ever increasing divergence between real data and estimated one. This problem usually tends to happen when, for simplicity sake, a real model is approximated by a linear one, so it can be better used by a linear Kalman filter. For some processes, the errors resulting from that approximation may be deem too unimportant, or the resulting decrease in accuracy they create too little, to be of importance. However, for more delicate and unforgiving processes, such as spacecraft trajectory filtering, those inaccuracies are of great importance. When using a unweighted filter, a poorly estimated state can start a drift between the measured data and the estimated one, which, if left unchecked, can grow with each new iteration, potentially ruining the accuracy of the filter, or at least seriously degrading its performance.

There is a need then for a more dynamic Kalman filter, one that can quickly react to dubiously obtained states, as well as gradually fade out older, more unreliable data, when face with newer one.

One of the solutions adopted is to increase the weight of new data with the implementation of a forgetting factor λ within the equation of the predicted error covariance, becoming:

$$P_k^- = \lambda_{k+1}F_kP_{k-1}{}^+F^T + Q_k \tag{4.21}$$

With $\lambda_k \geq 1$. This way, in each new iteration the newer data is given more importance than older one, thus, even when presented with dubious previous information, the more prevalent new information can again converge the estimated data with the measurements. Of course, the selection of which forgetting factor to use is of great importance.

(Xia et al (1994)) provides three methods to best choose the forgetting factor:

4.2.1 Fading Kalman Filter (First algorithm)

With this algorithm, one can obtain an optimal forgetting factor by successively iterate the following equation:

$$\lambda^{l+1}(k) = \lambda^l(k) - \varphi\frac{\partial f^l(\lambda;k)}{\partial\lambda^l(k)} \tag{4.22}$$

With the initial conditions being:

$$\begin{aligned}\lambda^0(1) &= 1 \\ \lambda^0(k) &= \lambda(k-1)\end{aligned} \tag{4.23}$$

Where k is the number of states, l is the number of iterations within a state. And 0<φ<1 is the step length.

If, at a p^{th} iteration, the following condition proves true:

$$\left|\lambda^p(k) - \lambda^{p-1}(k)\right| < \varepsilon \tag{4.24}$$

Ɛ being a small positive scalar that represents the degree of accuracy in the iterative process, then:

$$\lambda(k) = \max\{1, \lambda^p(k)\} \tag{4.25}$$

The gradient term used in Eq. (4.22) is obtained by the following equation:

$$\frac{\partial f^l(\lambda;k)}{\partial \lambda^l(k)} = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij}{}^l(k)\left(\frac{\partial S^l(k)}{\partial \lambda^l(k)}\right)_{ij} \tag{4.26}$$

Where:

$$S^l(k) = P(k)^- H^T - K(k)C_0(k) \tag{4.27}$$

And:

$$\frac{\partial S^l(k)}{\partial \lambda^l(k)} = FP(k-1)^+ F^T H^T[I - T(k)^{-1}C_0(k)] + K(k)HFP(k-1)^+ F^T H^T[I + T(k)^{-1}C_0(k)] \tag{4.28}$$

Finally, the covariance of the residual, defined in Eq. (4.5), $C0_k$, can be calculated by:

$$C_0(k) = G_1(k)/G_2(k) \tag{4.29}$$

Where:

$$\begin{aligned}G_1(k) &= G_1(k-1)/\lambda(k-1) + y(k)y(k)^T \\ G_2(k) &= G_2(k-1)/\lambda(k-1) + 1\end{aligned} \tag{4.30}$$

With initial conditions $G_1(0)$=0 and $G_2(0)$=0

The biggest drawback of the first algorithm is the use of extensive iterative calculations, this problem being exacerbated by the large number of points being filtered. Therefore, there is a need to obtain a valid equation for the forgetting factor without resorting to iterative calculations.

Thus, two new algorithms are considered, both being a one-step algorithm, meaning the forgetting factor can be calculated using a single equation, without the need for extensive iterations.

Overall, while the first algorithm allows for a higher degree of precision than the second and third algorithms, it will also require a higher computation time than the other two.

4.2.2 Fading Kalman Filter (Second and third algorithms)

Before using the following two algorithms, several assumptions must first be made: Q, R and $P(0)$ must be positive definite matrices, and H must be a full rank matrix.

For the first one-step algorithm, the equation for the forgetting factor is:

$$\lambda(k) = \max\{1, \frac{1}{m} trace[N(k)M^{-1}(k)]\} \tag{4.31}$$

Where

$$M(k) = H(k)FP(k-1)F^T H^T \tag{4.32}$$

$$N(k) = C_0(k) - HQH^T - R \tag{4.33}$$

Whether the second one-step algorithm has the same conditions and equations as the first one-step algorithm, with a slightly different equation for the forgetting factor:

$$\lambda(k) = \max\{1, \frac{1}{m} trace[N(k)] / trace[M(k)]\} \tag{4.34}$$

5. Numerical Results and Discussion

This chapter focuses on presenting the numerical results from the development of the controller and of both the robust Kalman filter as all of the three algorithms used for the fading Kalman filter, which are carefully described in the previous chapters, and the simulation of four trajectories, the reference orbit, also known as the Keplerian Orbit for this paper, the Real Orbit, with perturbations considered but without control of any kind, the Controlled Orbit, with perturbations considered but with an integrated H_∞ controller that obliges the real orbit to follow the referenced one and the Filtered Orbits, with perturbations considered but being filtered by different types of Kalman filters.

Python^{TM} programming language was used for the development of the controller and *MATLAB*^{TM} was used for the development of the Kalman Filter.

5.1 Dynamic Model

The movement of a spacecraft orbiting Earth with its oblateness considered is given by a dynamic model described in the second chapter of this paper. The state vector, representing the position, its respective derivative in spherical coordinates and the control vector, relative to the thrust acceleration components in the directions of every variable relative to the position are given below:

$$x(t) = \left[r, \dot{r}, \theta, \dot{\theta}, \phi, \dot{\phi} \right]^T \tag{5.1}$$

$$u(t) = \left[u_r, u_\theta, u_\phi \right]^T \tag{5.2}$$

The equilibrium points used to determine the A matrix, through the Jacobian one were:

$$x_{eq} = \begin{bmatrix} 6.760 \cdot 10^6 \\ 6.480 \cdot 10^{-1} \\ 0 \\ 1.270 \cdot 10^{-3} \\ 1.571 \\ 5.159 \cdot 10^{-6} \end{bmatrix} \tag{5.3}$$

These were estimated according to the values obtained from simulating the real orbit, which is l ater explained in this chapter.

To discretize, the *Python*^{TM} command *cont2discrete(A,B,C,D)* was used. It can be found in the *signal* library. Since it is an H_∞ control problem and the existence of one more matrix is present, E, a G matrix is used instead of a B matrix. G is a concatenated matrix of B and E.

$$G = \begin{bmatrix} B & E \end{bmatrix} \tag{5.4}$$

Later, the discretized forms of B and E will be given by the expression below:

$$G_d = \begin{bmatrix} B_d & E_d \end{bmatrix} \tag{5.5}$$

5.2 First Analysis for the developed Controller

After obtaining all these matrices, we can also analyze some important properties of the system referred in chapter 3, like stability, controllability and observability.

Through the analysis of the eigenvalues of the A matrix, it was concluded that the system is marginally stable because all the real parts of its eigenvalues are non-positive and at least one of them is zero.

The ranks of the controllability and observability matrices, (3.4) and (3.5) respectively, were also determined. Both have a rank equal to n, which proves the controllability and observability of the system.

It was also concluded that the A_d matrix was nonsingular, as requested by (3.7), for its determinant is different than zero.

For this paper, the simulated systems were always the nonlinearized ones. This is because some variables were too sensitive to be simulated on a simplified system and it was concluded that the system was stable but not attractive. The advantage in using the nonlinear system is that a more robust and realist system is being used to describe the motion of the vehicle, however it takes more time to compute the wanted results. The linearized system was, consequently, only utilized to design the controller.

The systems used for the Keplerian and Real orbits were the ones in (2.4) and (2.5), respectively without the control variable.

For the Controlled Orbit, the Butcher's Method (Chapra and Canale 2006), also known as the fifth order Runge-Kutta algorithm, was the numerical method chosen to simulate the system represented in (2.5) with the control variable varying properly for every step, dt. It was chosen due to its excellent accuracy even though it has a relatively slow execution time.

It is a solver of ODEs in the following form:

$$\dot{x} = f(x,u) \tag{5.6}$$

Where, $x(t) \in \mathbb{R}^n$, is the state of the system given by a varying vector and $u(t) \in \mathbb{R}^p$ is the input or control variable.

Finally, The Butcher's Method is described below:

$$x_{k+1} = x_k + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6) \tag{5.7}$$

With:

$$
\begin{aligned}
k_1 &= dt\ f(x_k, u_k) \\
k_2 &= dt\ f(x_k + \frac{1}{4}k_1, u_k) \\
k_3 &= dt\ f(x_k + \frac{1}{8}k_1 + \frac{1}{8}k_2, u_k) \\
k_4 &= dt\ f(x_k - \frac{1}{2}k_2 + k_3, u_k) \\
k_5 &= dt\ f(x_k + \frac{3}{16}k_1 + \frac{9}{16}k_4, u_k) \\
k_6 &= dt\ f(x_k - \frac{3}{7}k_1 + \frac{2}{7}k_2 + \frac{12}{7}k_3 - \frac{12}{7}k_4 + \frac{8}{7}k_5, u_k)
\end{aligned}
\tag{5.8}
$$

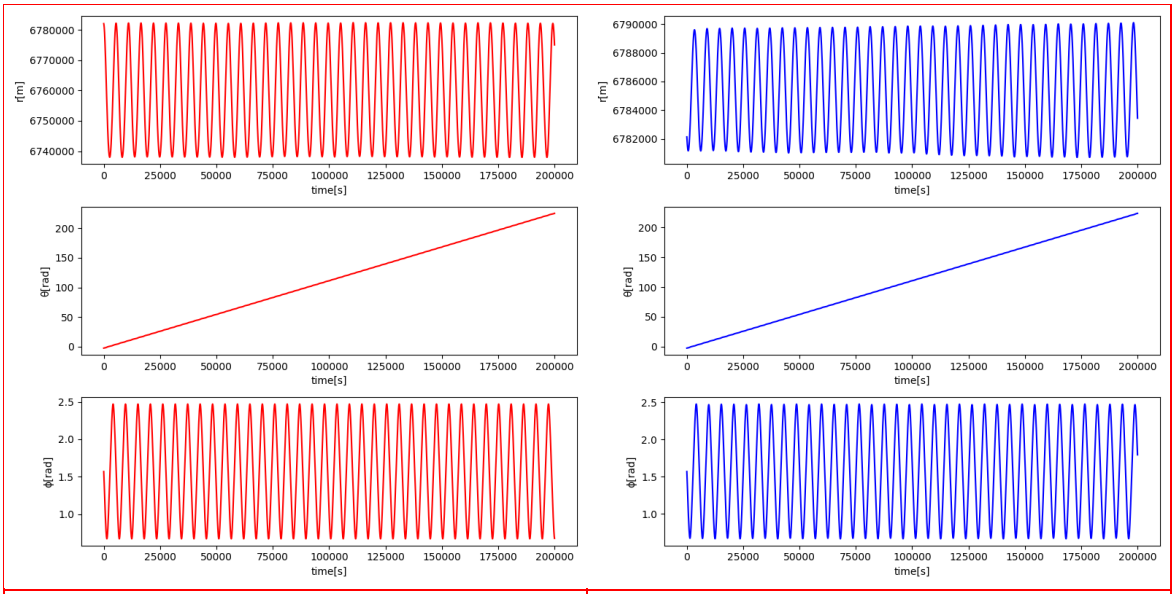
Due to the control actuation frequency, time step used was dt = 0.1 s, which proved to be a small enough step to compute all the results with excellent accuracy, although with the inconvenience, together with the fact of using a complex system (the nonlinearized one) and a robust solver for ODEs, of having a relatively slow computational time. However, it is essential to keep in mind that for the purpose of this paper, the primary objective is to develop a robust controller to work on critical real-life applications with a cutting-edge on-board processor designed for a specific space vehicle.

5.3 Detailed Analysis for the developed Controller

For better visualization and understanding of the problem, a code of colors was implemented. T he green color was attributed to the Keplerian Orbit, the red color was attributed to the Real Orbit and the blue color was attributed to the Controlled Orbit.

The default simulation time used for this paper was t = 200,000 s, which is approximately 2.5 days. This is enough to understand how the Real Orbit changes with time, because, in 2.5 days, the ISS revolves forty times around Earth, approximately. Also, it provides more than enough time to prove the robustness of the designed controller.

Below, the variation of the spherical coordinates for the Real Orbit and Controlled Orbit in function of time are graphically represented:



(a) Real Orbit | (b) Controlled Orbit

Fig. 5 Variation of the error from the Keplerian Orbit in function of time

For the Keplerian Orbit, which graphics are omitted, it is expected that r, the radial distance of the space vehicle from the center of the Earth varies a bit because the eccentricity of the orbit of the ISS is not null. When r = 6,790,000 m, the space vehicle is on the apogee of the orbit, approximately, and when r = 6,782,000 m, the space vehicle is on the perigee of the orbit, approximately. The other spherical elements, θ *and* ϕ vary so that the space vehicle can revolve around Earth. The first, the azimuth angle measured from the X-axis in the XY-plane, increases indefinitely and ϕ, the inclination measured from Z-axis to the vector r decreases and increases constantly.

For the Real Orbit, the apogee is now found at r = 6,780,000 m, approximately and the perigee of the orbit is now found at r = 6,740,000 m, approximately. Although the angles that compose the spherical elements seemed to not variate much in comparison to the Keplerian Orbit, later it is proved that this is not the case.

For the Controlled Orbit below, the radial distance of the space vehicle from the center of the Earth is the same as the Keplerian Orbit with the use of the designed controller, however, like on the Real Orbit there can't be any certainties towards the control of the angles that compose the spherical coordinates.

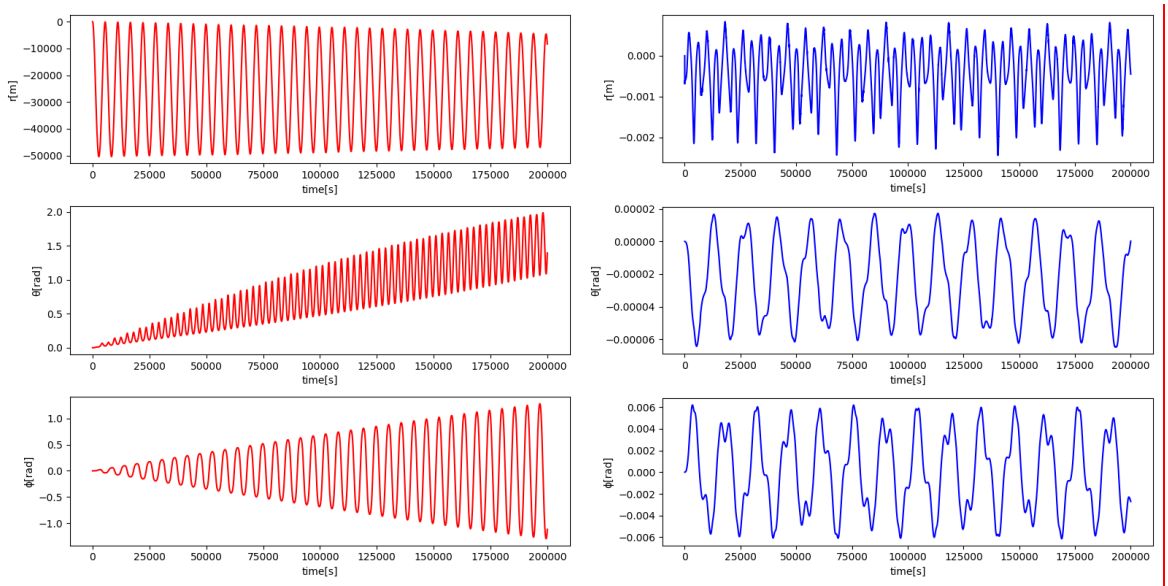
One way of finding out if the controller is working correctly is by measuring the error of the spherical coordinates of the Real Orbit from the Keplerian Orbit and compare to the one of the Controlled Orbit from the Keplerian Orbit.

Below, the error of the Real Orbit from the Keplerian Orbit is graphically represented, at red, in which, as foreseen, the error encountered for the radial distance of the space vehicle from the center of the Earth is troubling, however, and maybe not as expected, the deviations are even more serious for the angles that compose the spherical coordinates because they are increasing over time.

Below, the error of the Controlled Orbit from the Keplerian Orbit is also graphically represented but at blue. Here, the error is proven to be insignificant, and, more important than that,

it is converging to zero for all the elements composing the spherical coordinates. This proves that the controller is working properly and performing its duty adequately, even for a nonlinear system.

It is also fundamentally important to visualize how the trajectories are propagating in three dimensions to ensure that the dynamic system is properly describing a real case scenario and to compare the Controlled Orbit with the Keplerian Orbit in three dimensions.



(a) Real Orbit | (b) Controlled Orbit

Fig. 6 Variation of the error from the Keplerian Orbit in function of time

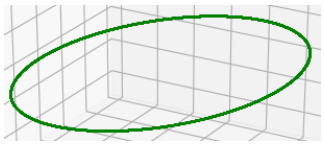
After solving the ODEs for the Keplerian, Real and Controlled orbits, three-dimensional graphical representations can be obtained on cartesian coordinates by converting the spherical coordinates into cartesian coordinates (Vermeille 2002), for every output of the state of the system, as shown below:

$$\begin{aligned} x_k &= r_k \sin\phi_k \cos\theta_k \\ y_k &= r_k \sin\phi_k \sin\theta_k \\ z_k &= r_k \cos\phi_k \end{aligned} \tag{5.9}$$

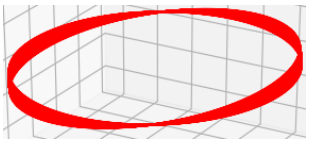
The following figure shows the three-dimensional representation of the simulated orbits. For the Keplerian Orbit. it does not matter what the simulation time is, as long as it is enough time for the ISS to perform a single revolution around the Earth, this graphical representation will always look the same, a single closed line. This is expected since a Keplerian Orbit only considers the gravitational forces of the two main bodies, Earth and ISS, and therefore a repetitive orbit is obtained.

For the Real Orbit, the effects of J_2 on a LEO in approximately 2.5 days are clearly represented.

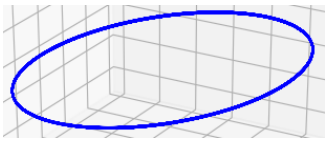
Finally, for the Controlled Orbit and as expected, it is safe to say that it is identical to the Keplerian Orbit, which proves the analysis made from Fig. 5.



(a) Keplerian Orbit



(b) Real Orbit



(c) Controlled Orbit

Fig. 7 Representation of the orbits in 3D

The designed controller is proven to be robust, because it is designed to work in real life situations. The whole system remains within chosen limits, because it successfully follows the desired trajectory and therefore, it is safe to say that the designed controller also possesses robust stability. Finally, because it can also track the desired state and rejects the considered perturbations it can also be said that the designed controller possesses robust performance (Chen 2000).

It is also expected that the controller behaves well even if the system considers random errors or minor disturbances that happen to space vehicles in LEO.

5.4. Analysis of the implemented Kalman Filters

In this section there will be an analysis of the estimation of the ISS trajectory using the different Kalman filters developed in the sections above.

Before they can be used, however, there is a necessity to describe the equations for the state transition and measurement equations:

For the implementation of the robust Kalman filter, the state vector of the ISS trajectory will first be converted from spherical coordinates to Cartesian coordinates according to the conversion equations in Eq. (5.9). This allows for a state matrix of the form $[X \quad Y \quad Z \quad 0 \quad 0 \quad 0]$.

Next a linear, discrete-time system that represents the trajectory of the ISS will be created according to Eq (4.1), where the state transition and measurement matrix are respectively:

$$F=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{and } H=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.10}$$

With the Gaussian white noise from both the state and measurement equations being represented by the following matrices, respectively Q and R:

$$Q=\begin{bmatrix} 0.01^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01^2 \end{bmatrix}, \text{ and } R=\begin{bmatrix} 0.9^2 & 0 & 0 \\ 0 & 0.9^2 & 0 \\ 0 & 0 & 0.9^2 \end{bmatrix}. \tag{5.11}$$

Lastly, the initial error covariance matrix $P(0)$, is assumed to be an identity matrix.

With the above, matrices, and with Eqs. (4.2)-(4.5) a normal Kalman filter was constructed, with the results being present later in this chapter.

For the robust Kalman filter, the following assumptions were made:

First, the uncertain part of the state matrix F, ΔF was assumed to be:

$$\Delta F = \begin{bmatrix} \delta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.12}$$

With $\|\delta\| \leq 0.5$. The measurement matrix, H, was considered to be fully known. The uncertain parameter δ, although it can be considered fairly high, was chosen for the reason that a higher uncertain parameter allows the robust Kalman filter to better stand out from the normal Kalman filter, thus becoming easier to understand how uncertainty can affect trajectory estimation.

The decision to only assume uncertainty in the X axis was also deliberate, so one can better understand how uncertainty affects the estimations of one axis, when compared to the other two.

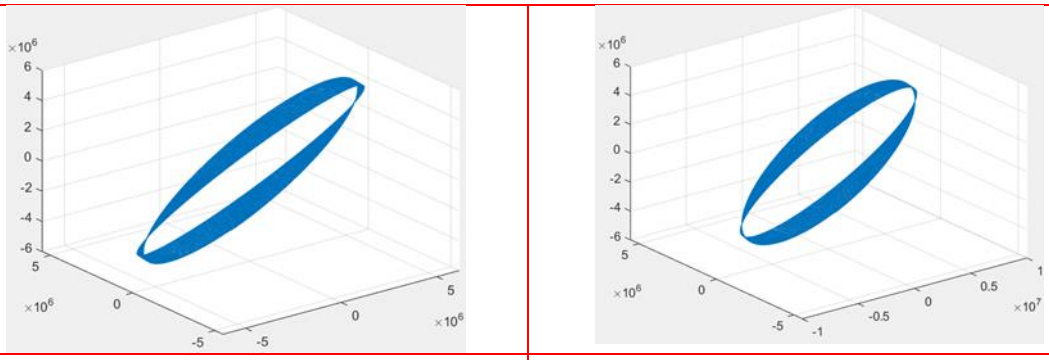
Following Eqs. (4.7)-(4.8) : one can then calculate the following matrices:

$$H_1 = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ and } \quad E = [0.05 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \tag{5.13}$$

Considering ε=0.09 and ν=0.09, both Algebraic Riccati equations in Eq. (4.11) and Eq. (4.18), can be solved for stable results. Thus one can obtain a robust Kalman estimator with:

$$\hat{F} = \begin{bmatrix} 1.1763 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } \quad \hat{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.14}$$

When comparing with Eq. (5.10), one can see how uncertainty can affect the state transition matrix, as the X axis will have the highest deviation when compared to the other two axis, whose values can be considered the same as the original. The effects of uncertainty in trajectory estimation become even more evident when the Robust Kalman filter is presented side by side with a trajectory estimated by a normal Kalman filter:



(a) Normal Kalman Filter Estimation | (b) Robust Kalman Filter Estimation

Fig. 8 Comparison of the developed Kalman Filters

As for the Fading Kalman filter, all three algorithms were tested, whose results are posted along with an estimation from a normal Kalman filter for comparison. Because of the scale and the complex and overlapping nature of the trajectory, it was decided to present the results of the trajectory estimation for the first forty thousand points, as they present a simpler and more streamlined trajectory. It is to note, however, that the implemented Kalman filters are fully capable of estimating the trajectory for the complete two million point trajectory.

Under these circumstances, the following graphic depicts the trajectory estimation of the normal Kalman filter:

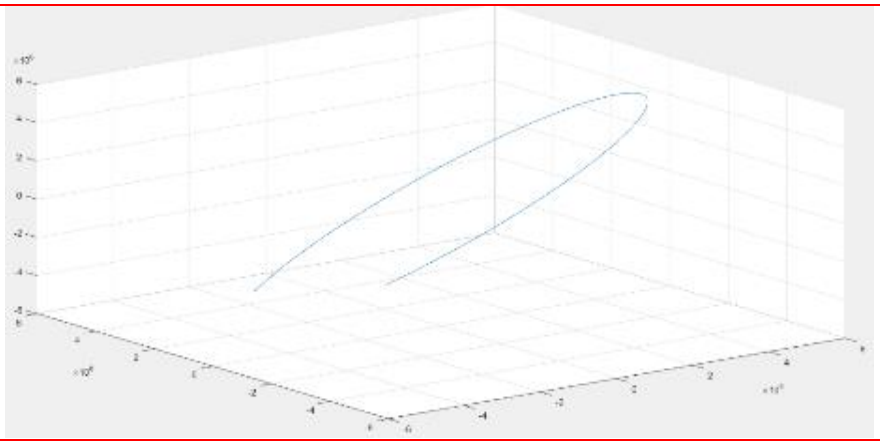


Fig. 9 Trajectory estimation for a Normal Kalman Filter

With the results from the three implemented Kalman filters being presented as the following graphics:

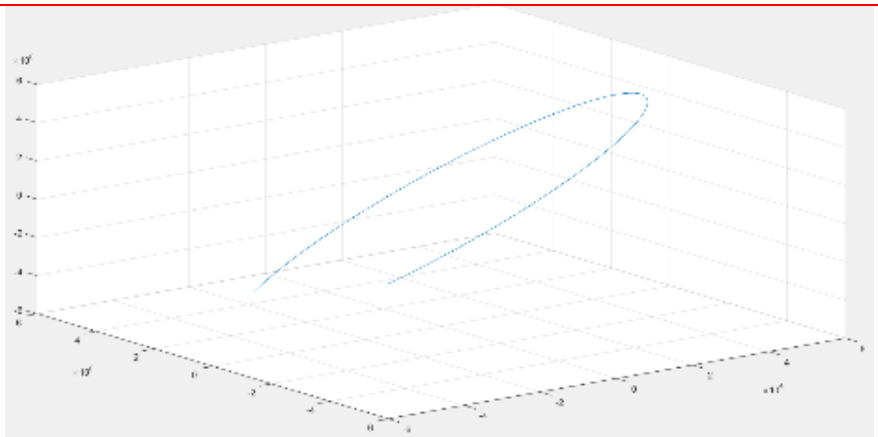


Fig. 10 Trajectory estimation for a Fading Kalman filter, first algorithm

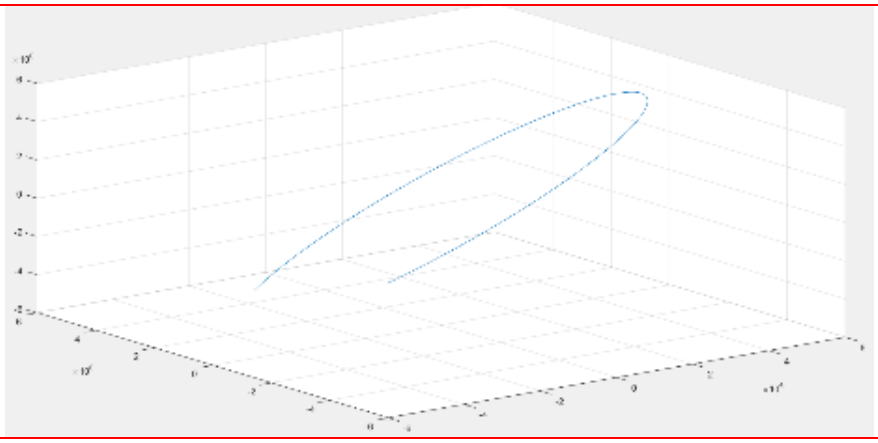


Fig. 11 Trajectory estimation for a Fading Kalman filter, second algorithm

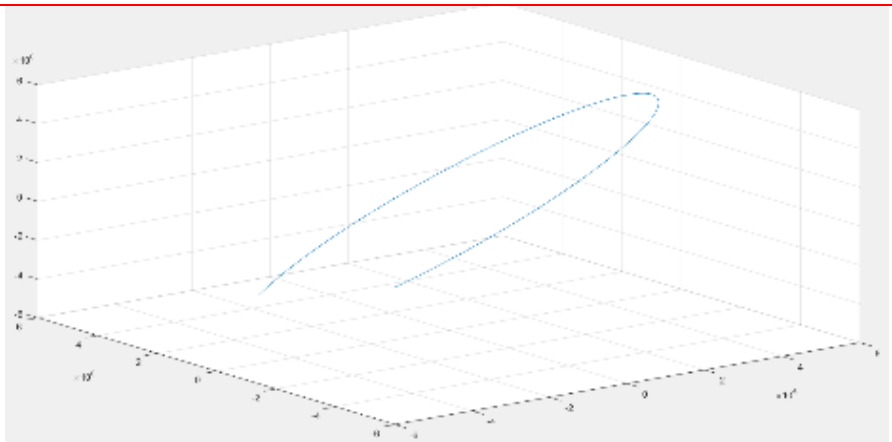


Fig. 12 Trajectory estimation for a Fading Kalman filter, third algorithm

From the observed results it can be noted that the three algorithms behaved exactly like a normal Kalman filter. This phenomenon can happen when the linear model chosen to represent the trajectory greatly resembles the real one, so the imperfections and noise that can happen from this simplification simply do not exist, or are incredibly small if they do, so there is no need to overweight newer states to eliminate the effect of older ones, or to force the filter to converge again. This way, the forgetting factor λ will always be 1.

6. Conclusions

An H_∞ controller was designed in order to successfully alter the orbit of a space vehicle with perturbations, so that it follows a desired orbit with the same initial conditions. The controller showed great overall performance, it was proven to be robust, possess robust stability and robust performance.

The two types of Kalman filter being implemented were both successful in tackling their respective problems: the implementation of a robust Kalman filter against uncertainty improved the precision and performance of the trajectory estimation, and the implementation of the fading Kalman filters, although ultimately made redundant by the precise linearization of the system equations, were nonetheless correctly built, and can be applied to other less precisely linearized systems for improved performance.

Acknowledgements

This research work was conducted in the Laboratory of Avionics and Control of the university of Beira Interior (Covilhã, Portugal) and supported by the Portuguese Foundation for Sciences and Technology (FCT) and the Aeronautics and Astronautics Research Center (AeroG) of the Associated Laboratory for Energy, Transports and Aeronautics (LAETA).

References

Massioni, P., Keviczky, T., Gill, E. and Verhaegen, M. (2011), "A decomposition-based approach to linear time-periodic distributed control of satellite formations", *IEEE Transactions on Control Systems Technology*, **19**(3), 481-492. https://doi.org/10.1109/TCST.2010.2051228.

Yaesh, I. and Shaked, U. (1991), "A transfer function approach to the problem of discrete-time systems: H_∞-optimal linear control and filtering", *IEEE Transactions on Automatic Control*, **36**(11). https://doi.org/10.1109/9.100935.

Xia, Q., Rao, M., Ying, Y. and Shen, X. (1994), "Adaptative Kalman filter with an application", *Automatica*,**30**,8, 1333-1338. https://doi.org/10.1016/0005-1098(94)90112-0.

Moghadasian, M. and Roshanian, J. (2019), "Approximately optimal manoeuvre strategy for aero-assisted space mission", *Advances in Space Research Engineering*, **64**(2), 436-450. https://doi.org/10.1016/j.asr.2019.04.003.

Parvez, S.A. and Xing, G-Q. (1998), "Autonomous orbit control with position and velocity feedback using modern control theory", US Patent 6089507.

Vermeille, H. (2002), "Direct transformation from geocentric coordinates to geodetic coordinates", *Journal of Geodesy*, **76**(8), 451-454.

https://doi.org/10.1007/s00190-002-0273-6.
Park, J-U., Choi, K-H. and Lee, S. (1999), "Orbital rendezvous using two-step sliding mode control", *Aerospace Science and Technology*, **3**(4), 239-245. https://doi.org/10.1016/S1270-9638(99)80046-7.
Xie, L., Soh, Y, C. and Souza, C, E. (1994) "Robust Kalman filtering for uncertain discrete-time systems", *IEEE transactions on automatic control*, **39**(6), 1310-1314. https://doi.org/10.1109/9.293203.
Zhang, J., Yang, K., Qi, R., Zhao, S. and Li, Y. (2017), "Robustness analysis method for orbit control", *Acta Astronautica*, **137**, 15-24. https://doi.org/10.1016/j.actaastro.2017.03.034.
Chao, C-C. (2005), "*Applied Orbit Perturbation and Maintenance*", Aerospace Press, California, CA, USA.
Curtis, H. (2005), "*Orbital Mechanics for Engineering Students*", Elsevier Aerospace Engineering Series, Florida, FL, USA.
Nise, N.S. (2011), "*Control System Engineering*", John Wiley & Sons, CA, USA.
Iggidr, A. (2004), "*Controllability, Observability, and Stability of Mathematical Models",* Encyclopedia of Life Support Systems (EOLSS), Mathematical Models, UNESCO, France.
Chapra, S.C. and Canale, R.P (2006), "*Numerical Method for Engineers*", McGraw-Hill Higher Education, Massachusetts, MA, USA.
Chen, B. (2000), "*Robust & H_∞ Control*", Springer- Verlag London, Singapore.
Larson, W.J. and Wertz, J.R. (1999), "*Space Mission Analysis and Design*", Space Technology Library, California, CA, USA.
Wertz, J.R. (2011), "*Space Mission Engineering: The New SMAD*", Space Technology Library, California, CA, USA.
Neokleous, K. (2007), "Modeling and control of a satellite's geostationary orbit", M.Sc. Dissertation; Luleå University of Technology, Prague, Czech Republic.
Heavens Above (2019), ISS – Orbit; https://www.heavens-above.com/orbit.aspx?satid=25544