Thrusting System for Electrostatic Space Debris Control

Master Thesis Report

AE5810: Thesis Space

Guillem Rueda Oller

Delft University of Technology

Thrusting System for Electrostatic Space Debris Control

Master Thesis Report

by

Guillem Rueda Oller

in partial fulfillment to obtain the degree of Master of Science
at Delft University of Technology.

Presented and defended on
9th May 2023

Student number:	5006538	
Project duration:	January 2022 – April 2023 (AE5810)	
	Jan. 2022 to Apr. 2022 full-time	
	May 2022 to Apr. 2023 part-time (0.25 FTE)	
Thesis supervisor:	Dr. Jian Guo,	TU Delft, supervisor
	Dr. Hanspeter Schaub,	CU Boulder, supervisor

Cover Image: Space Debris Ring. Image credit: MIT.edu

TU Delft

Abstract

The geosynchronous large debris reorbiter concept is an active debris removal method proposed for defunct satellites in the geostationary orbit using the electrostatic tractor. The electrostatic tractor, or tug, is a spacecraft that controls a mutual electrostatic force with a target and uses this force to slowly accelerate the target towards or away from the tug. The challenging task of station keeping a few dozen meters away between the tug and the debris spacecraft must be done with thrusters that can only intermittently fire. In between the firing, the electron gun is used to control the tug and debris potential. This research investigates different thruster options and discrete control algorithms for this application of the electrostatic tractor. The objective is to devise and simulate a reliable propulsion system for the geosynchronous large debris reorbiter concept by simulating and analyzing different micro-propulsion thrusting options and developing a discrete thrusting solution using pulsed thrust and active charging control.

Guillem Rueda Oller
Delft, May 2023

Acknowledgements

First, I would like to truly thank Dr. Hanspeter Schaub for hosting me in his Autonomous Vehicle Systems laboratory. With the guidance and support I received during my stay at the University of Colorado Boulder, I managed to carry out this project. It has been a pleasure to work with you.

Many thanks to all the members of the Autonomous Vehicle Systems laboratory, who made my day for roughly half year. Also, I am very grateful for all the wonderful hours spent with other students, interns and researchers I met throughout my stay in Boulder.

I would like to specially thank Dr. Jian Guo. Your advice when I had to make difficult decisions and your guidance when I had to write this thesis report were key to my success. Special thanks to the University of Colorado Boulder, the Delft University of Technology and the Fundatie van de Vrijvrouwe van Renswoude te Delft for funding my research stay at the University of Colorado Boulder.

Also, I want to thank the tremendous support received from my family and friends during this time. As always, a huge thank you to my parents and my brother for always encouraging me to follow my dreams.

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the concept of the geosynchronous large debris reorbiter for space debris removal. Motivation for this research is provided, including an overview of the amount of space debris in geostationary orbit, current active debris removal techniques and the thrusting challenge of the geosynchronous large debris reorbiter concept. Next, a research objective is defined and a main research question is presented, followed by research sub-questions. The last section of this chapter describes the thesis structures for the rest of the thesis report.

1.1. Concept

The geosynchronous large debris reorbiter concept is an active debris removal method proposed for defunct satellites in the geostationary orbit using the electrostatic tractor. The electrostatic tractor is a spacecraft (here referred as the tug) that controls a mutual electrostatic force with a target and uses this force to slowly accelerate the target towards or away from the tug. The target of an electrostatic tractor spacecraft can be an asteroid or another spacecraft. The challenging task of station keeping a few dozen meters away between the tug and the debris spacecraft must be done with thrusters that can only intermittently fire. In between the firing, the electron gun is used to control the tug and debris potential. In order to advance on making the geosynchronous large debris reorbiter a reality, this work investigates different thruster options and discrete control algorithms for this application of the electrostatic tractor. Thruster characteristics and uncertainties must be modelled and taken into account by the control algorithms. The objective of this thesis project is to devise and simulate a reliable propulsion system for the geosynchronous large debris reorbiter concept by simulating and analyzing different micro-propulsion thrusting options and developing a discrete thrusting solution using pulsed thrust and active charging control.

1.2. Motivation

This section introduces the current situation and challenges of space debris in GEO. A survey of possible active debris capturing and removal methods for space debris in GEO is presented. The Geosynchronous Large Debris Reorbiter concept is presented as a feasible and suitable solution. However, in order to make this concept a reality, research on its propulsion system is necessary. This research project devises and proposes this propulsion system.

1.2.1. Space Debris in GEO

A geosynchronous orbit is an Earth-centered orbit with an orbital period equal to Earth's spin period, which is about 23 hours 56 minutes 4 seconds (one sideral day). A geostationary orbit, also called the Geosynchronous Equatorial Orbit (GEO), is a circular geosynchronous orbit in Earth's equatorial plane. This definition makes it appear in a fixed position in the sky to ground observers. Theoretically, only one geostationary orbit exists and it is located at 35,786 kilometres altitude from Earth's Equator (42,164 kilometres in radius from Earth's center). A craft moving continuously using power could maintain a very different geostationary height, but this would not be very efficient. Therefore, the GEO and its vicinity are extremely valuable and unique for applications such as communications, meteorology and navigation, which make use of geostationary satellites. Currently, there are over 1,600 objects in GEO –including operational satellites and space debris–, making it a very crowded place [1]. Each satellite occupies a different position in GEO. Some of these valuable "slots" remain occupied by satellites that become inoperable or reach their end-of-life while in GEO. Moreover, solar radiation pressure and lunar gravitational perturbations may drift these uncontrolled satellites to other slots which would threaten other geostationary satellites. Geostationary uncontrolled satellites and their detached parts are unwanted space debris in GEO.

Initially, space debris was not considered a serious issue by satellite operators [2]: there were too few objects in space and large distances between them to the point that probabilities of collision were insignificant. Moreover, operators could easily track space debris and reorbit operational satellites when necessary in order to avoid collisions. Today, the high amount of space debris orbiting Earth is a matter of worldwide concern. Kessler states that more debris is created when a critical debris density has been reached, even if new artificial satellites are no longer launched into space [3]. Therefore, space debris threatens to destroy operational satellites that are considered vital for mankind nowadays.

Dysfunctional satellites and upper rocket stages, high- and low-intensity explosions and hypervelocity impacts with spacecraft are examples of different ways that generate space debris. In 1975, NASA had investigated the mass distribution of orbiting man-made space debris, which resulted in estimation models with power and exponential terms [4]. This implies that space debris generation increases with a higher rate over time. Table 1.1 shows that as of December 16, 2021, there are more than 24,000 known objects in space with sizes larger than 10 cm, and that the 67% is space debris.

Table 1.1: Orbital population till December 16, 2021 [5].

Status	Payloads	Debris	All
On orbit	8,059	16,347	24,406
Decayed	3,739	21,994	25,733
Total	11,798	38,341	50,139

In this situation, current practice is to follow the 25-year safety standard: the satellite retains sufficient end-of-life fuel to either lower its orbit and re-enter, or raise itself to a graveyard orbit, within the 25 years after the mission ends [6]. Geostationary satellites raise to a super-synchronous graveyard orbit for disposal, typically 250 to 300 kilometres beyond the geostationary orbit [7]. Discarded satellites in these orbits are not able to reach geostationary altitude again within 25 years, even under lunar and solar radiation disturbances [8]. However, there are still many older satellites, current malfunctioning satellites and space debris in GEO. The Annual Space Environment Report issued in 2021 by the European Space Agency discloses that, from over the 1,600 objects in GEO during 2020, less than 550 were under control [1]. Figure 1.1 shows that uncontrolled objects in GEO were either drifting, in libration orbits, or in highly-inclined orbits. Uncontrolled objects in GEO pose a hazard to operational satellites. Consequently, a need to capture and remove space debris objects exists.

1.2.2. Active Debris Capturing and Removal in GEO

Active Debris Removal (ADR) targets in the geostationary orbit are very challenging. Contact or close proximity with these targets is difficult because of the uncontrolled attitude (including uncontrolled rotation) of the debris object. Also, the non-cooperative target poses a significant danger to the space-tug craft because of the unknown structural integrity of the debris object and its rotation. The defunct satellite could break apart when only pulled at the contact point. Moreover, space debris in GEO must be removed using orbit raising maneuvers, which can require significant amounts of fuel.

Reference [9] provides a good review and comparison of the existing technologies on active space debris capturing and removal. The paper presents first the methods for capturing space debris. As

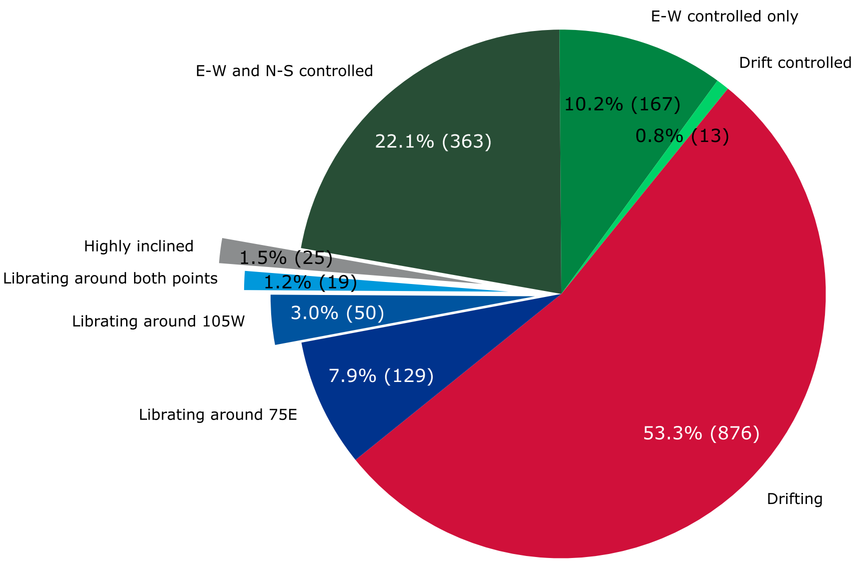


Figure 1.1: Classification of objects in GEO during 2020 [1].

represented in Figure 1.2, these methods are divided into two categories: contact and contactless capturing methods. For the scenario given in Section 1.2.1, contact capturing methods are not desirable for debris in GEO that have uncontrolled attitude and unknown structural integrity. Uncontrolled satellites in geostationary orbit are particularly heavy, making them more dangerous to any other spacecraft that tries physical contact. The authors of [9] consider that contactless capture methods are primarily thought for asteroid deflection and therefore do not further discuss them.

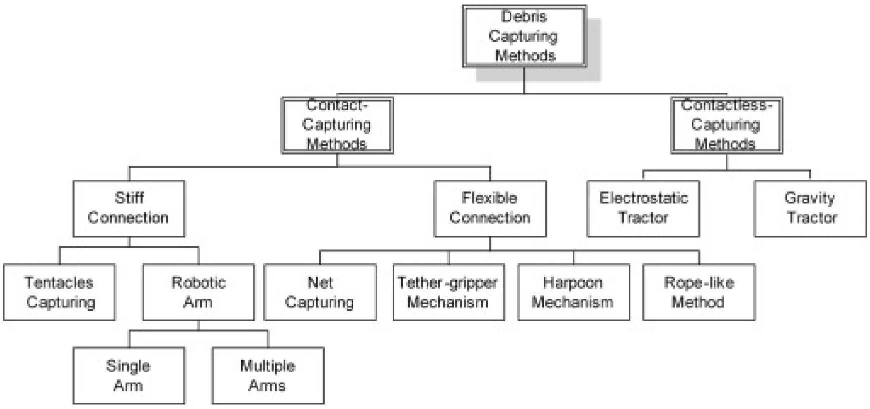


Figure 1.2: Concept diagram of capturing methods [9].

Figure 1.3 presents the methods for space debris removal. Drag augmentation systems, electro-dynamic tethers, contactless removal methods and contact removal methods are considered the most relevant and promising active debris removal methods [9]. However, drag augmentation systems do not apply to debris in GEO because these debris objects have to raise to a graveyard orbit, whereas drag can only be used to lower orbits. Also, electro-dynamic tether removal methods cannot treat targets beyond Low Earth Orbit (LEO) due to the insufficient geomagnetic intensity [10]. Artificial atmosphere influence, laser systems and Ion Beam Shepherd are the three contactless removal methods presented in reference [9]. The first two methods cannot raise orbits and therefore are not of interest for GEO space debris. The Ion Beam Shepherd is a concept to lower debris altitude by ejecting highly collimated neutralized plasma beam onto the debris object. Besides, Kitamura suggested that this shepherd could be used to reorbit space debris in GEO too [11]. Finally, as assessed for contact capturing methods, contact removal methods are not recommended for uncontrolled satellites in GEO.

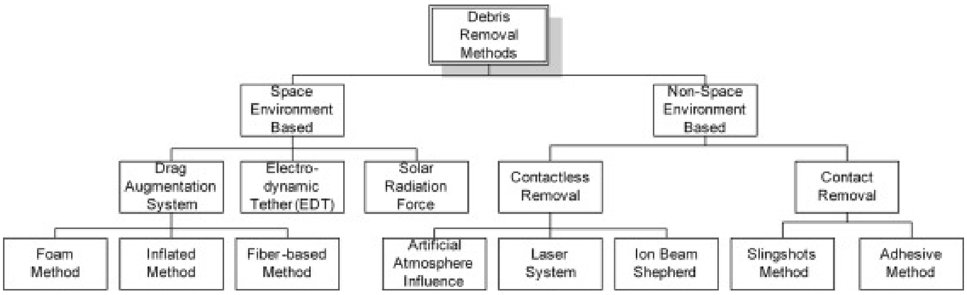


Figure 1.3: Concept diagram of removal methods [9].

1.2.3. Geosynchronous Large Debris Reorbiter

The Electrostatic Tractor (ET) has been proposed to touchlessly remove space debris from GEO by taking advantage of inter-craft Coulomb forces [8]. Referred as the Geosynchronous Large Debris Reorbiter (GLiDeR) concept in the original publication, the controlled spacecraft, referred to as the servicer or tug, emits an electron beam onto the space debris. This makes the tug and debris potentials to increase several 10s of kilovolts, with positive charge for the tug and negative charge for the debris. The resulting attractive Coulomb force in the order of milli-Newtons between the two craft allows the tug to pull the debris to a graveyard orbit without any physical contact. This method can be used to raise the altitude of an average geostationary retired satellite by 300 kilometres in two months [12]. To this day, extensive research led by Dr. Hanspeter Schaub has been carried out on this proposal.

Thrusting Solution for the GLiDeR

No research on possible thrusting solutions for the geosynchronous large debris reorbiter concept has been published yet. All research on this concept assumes that the tug spacecraft can generate instantaneously any thrust force vector with infinite resolution and no maximum force magnitude. Particularly, previous research assumed that the tug propels exactly as dictates the required thrust control acceleration in Equation 2.13. However, we will see that this is far from the real requirements on the field. The original team that worked on the geosynchronous large debris reorbiter concept decided to start the research using this simple model for easiness. Today, the author of this thesis is part of the current team working on the GLiDeR concept.

In order to make progress in the research on the ET concept for GEO debris removal, it is necessary to find a realistic thrusting solution for the tug. First, it is necessary to simulate, analyze and select which thrust systems work well on the ET. Each thrusting option performs differently depending on its thrusting technology (e.g., cold gas) and the thruster characteristics (e.g., specific impulse). As will be shown when formulating the relative motion dynamics in Section 2.1, the thrust on the Electrostatic Tractor is in the same order of magnitude as the electrostatic force, which is in the order of milli-Newtons [12]. Therefore, micro-thrusters are necessary to achieve that low level of thrust.

Furthermore, it is necessary to design, test and evaluate a discrete thrusting solution for the ET propulsion system. The performance and suitability of a thruster system will also depend on the relative motion control algorithm. Whereas a continuous and ideal propulsion system was considered in previous research, this research has to work with algorithms that consider thrusters either open or close, instead of being able to provide any thrust. The Literature Study report shows that the ET micro-thrusters and the mounted electron beam cannot work at the same time, as the thruster exhaust interferes with the electron beam functioning [13, 14]. The propulsion system and control strategy proposed and designed in this research will have to ensure that the micro-thrusters and the electron beam are not working simultaneously.

It is necessary to figure out which propulsion system and control strategy are suitable for the geosynchronous large debris reorbiter concept. The goal of this research is to devise and simulate a reliable propulsion system for the geosynchronous large debris reorbiter concept. This research proposes and compares thrusting options and designs and simulates a relative motion control algorithm for this application of the Electrostatic Tractor.

1.3. Research Objective and Questions

The research objective of the thesis project is:

To devise and simulate a reliable propulsion system for the geosynchronous large debris reorbiter concept **by** surveying different micro-propulsion thrusting options and developing a discrete thrust control algorithm **using** pulsed-width modulation (PWM) thrust and active charging control.

In order to achieve the research objective, the following research question has to be answered:

What micro-propulsion thrusting options and thrust control strategy on the electrostatic tractor are suitable for the geosynchronous large debris reorbiter concept?

The research team of the GLiDeR finds useful to answer the sub-questions below in order to achieve the research objective. The following sub-questions will be answered during this research:

1. What are the general guidelines and recommendations for the propulsion system of the geosynchronous large debris reorbiter concept?
2. Can the electrostatic tractor remove GEO debris by using current micro-propulsion options?
3. What thruster configurations improve the performance of the GLiDeR concept?
4. What PWM control algorithm can do optimal station keeping with on-off thrusters?

1.4. Thesis Structure

The rest of this thesis is structured as follows:

Chapter 2 provides the physics model of the Electrostatic Tractor for GEO space debris removal. Relative motion dynamics of the ET and the debris object are modelled, and an initial feedback control is presented. The electrostatic force generated by the electron beam is modelled too.

Chapter 3 reviews and compares the propulsion system options available to use in the Electrostatic Tractor. Thrust-related parameters of propulsion systems are reviewed, simulated and tested. The simulation helps to determine how several parameters affect the propulsion system.

In Chapter 4, the author proposes and formulates a discrete thrust control algorithm for the Electrostatic Tractor trajectory planning problem. The chapter starts assessing which control algorithm can be used for this application. Then, the algorithm is mathematically formulated. Finally, the algorithm is particularized to the case of the Electrostatic Tractor thrust controller.

In Chapter 5, the physics model and the discrete thrust control algorithm designed in the previous chapter are simulated. The simulation results for different scenarios, tuned controller parameters and multiple micro-thrusters configurations are shown, plotted and analyzed.

Finally, Chapter 6 summarizes the work conducted in this thesis, identifies its main findings and contributions, and highlights opportunities for future work.

2

Physics Model

The Electrostatic Tractor (ET) has been proposed to touchlessly remove space debris from GEO by taking advantage of inter-craft Coulomb forces [8]. Referred as the Geosynchronous Large Debris Reorbiter (GLiDeR) concept in the original publication, the controlled spacecraft, referred as the servicer or tug, emits an electron beam onto the space debris. This makes the tug and debris potentials to increase several 10s of kilovolts, with positive charge for the tug and negative charge for the debris. The resulting attractive Coulomb force in the order of milli-Newtons between the two craft allows the tug to pull the debris to a graveyard orbit without any physical contact. This method can be used to raise the altitude of an average geostationary retired satellite by 300 kilometres in two months [12]. To this day, extensive research led by Dr. Hanspeter Schaub has been carried out on this proposal. This chapter provides the physics model of the Electrostatic Tractor for GEO space debris removal.

2.1. Relative Motion Dynamics

Reference [12] derives the relative motion dynamics of the system. A Hill frame $\mathcal{H} : \left\{\hat{\boldsymbol{h}}_r, \hat{\boldsymbol{h}}_\theta, \hat{\boldsymbol{h}}_h\right\}$ with origin at the tug's center of mass is defined by

$$\hat{\boldsymbol{h}}_r = \frac{\boldsymbol{r_T}}{r_T}, \qquad \hat{\boldsymbol{h}}_\theta = \hat{\boldsymbol{h}}_h \times \hat{\boldsymbol{h}}_r, \qquad \hat{\boldsymbol{h}}_h = \frac{\boldsymbol{r_T} \times \dot{\boldsymbol{r}}_{\boldsymbol{T}}}{|\boldsymbol{r_T} \times \dot{\boldsymbol{r}}_{\boldsymbol{T}}|} \tag{2.1}$$

where $\boldsymbol{r_T}$ is the inertial position vector of the tug, $\dot{\boldsymbol{r}}_{\boldsymbol{T}}$ is the inertial velocity vector of the tug, and $r_T = |\boldsymbol{r_T}|$. Being $\boldsymbol{r_D}$ the inertial position of the debris, vector $\boldsymbol{\rho} = \boldsymbol{r_D} - \boldsymbol{r_T}$ is the relative position vector of the debris with respect to the tug.

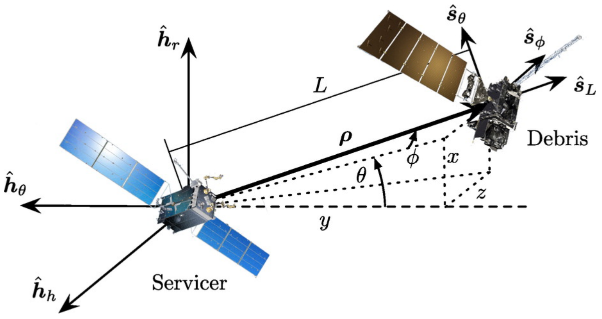


Figure 2.1: Hill frame $\mathcal{H}$ and spherical frame $\mathcal{S}$ [15].

Taking two inertial time derivatives of the relative position vector $\boldsymbol{\rho}$, the relative acceleration vector is $\ddot{\boldsymbol{\rho}} = \ddot{\boldsymbol{r}}_{\boldsymbol{D}} - \ddot{\boldsymbol{r}}_{\boldsymbol{T}}$. The inertial accelerations of tug ($\ddot{\boldsymbol{r}}_{\boldsymbol{T}}$) and debris ($\ddot{\boldsymbol{r}}_{\boldsymbol{D}}$) are

$$\ddot{\boldsymbol{r}}_{\boldsymbol{T}} = -\frac{\mu}{r_T^3}\boldsymbol{r}_{\boldsymbol{T}} + \frac{\boldsymbol{F}_{\boldsymbol{c}}}{m_T} + \boldsymbol{u}_{\boldsymbol{T}} \tag{2.2a}$$

$$\ddot{\boldsymbol{r}}_{\boldsymbol{D}} = -\frac{\mu}{r_D^3}\boldsymbol{r}_{\boldsymbol{D}} - \frac{\boldsymbol{F}_{\boldsymbol{c}}}{m_D} \tag{2.2b}$$

where $\mu = 6.674 \times 10^{-11}$ m^3 kg^{-1} s^{-2} is the gravitational constant for Earth and m_T and m_D are the masses of tug and debris respectively. The first term of the right hand side of both equations is the gravitational acceleration. The second term is the attractive electrostatic acceleration caused by Coulomb force $\boldsymbol{F}_{\boldsymbol{c}}$. The third term $\boldsymbol{u}_{\boldsymbol{T}}$ acting on the tug is the thruster control acceleration that tug's inertial thrusters generate. Now the relative acceleration vector $\ddot{\boldsymbol{\rho}}$ can be written as

$$\ddot{\boldsymbol{\rho}} = -\frac{\mu}{r_D^3}\boldsymbol{r}_{\boldsymbol{D}} + \frac{\mu}{r_T^3}\boldsymbol{r}_{\boldsymbol{T}} - \frac{\boldsymbol{F}_{\boldsymbol{c}}}{m_D} - \frac{\boldsymbol{F}_{\boldsymbol{c}}}{m_T} - \boldsymbol{u}_{\boldsymbol{T}} \tag{2.3}$$

Equation 2.3 forms the relative Equations of Motion (EOM) in the Hill frame $\mathcal{H}$. If the electrostatic force accelerations and the thruster control acceleration are grouped under the total control acceleration vector $\boldsymbol{u}$ such that

$$\boldsymbol{u} = -\boldsymbol{F}_{\boldsymbol{c}}\left(\frac{1}{m_T} + \frac{1}{m_D}\right) - \boldsymbol{u}_{\boldsymbol{T}} \tag{2.4}$$

then the EOM is expressed as the Clohessy-Wiltshire-Hill (CWH) relative EOM:

$$\ddot{\boldsymbol{\rho}} = -\frac{\mu}{r_D^3}\boldsymbol{r}_{\boldsymbol{D}} + \frac{\mu}{r_T^3}\boldsymbol{r}_{\boldsymbol{T}} + \boldsymbol{u} \tag{2.5}$$

Equation 2.5 can be linearized using the relative position vector ${}^{\mathcal{H}}\boldsymbol{\rho} = [x, y, z]^T$ expressed in cartesian coordinates in the Hill frame and obtain

$$\ddot{x} - 2n(t)\dot{y} - 3n^2(t) = u_x \tag{2.6a}$$

$$\ddot{y} + 2n(t)\dot{x} = u_y \tag{2.6b}$$

$$\ddot{z} + n^2(t) = u_z \tag{2.6c}$$

Semi-major axis a changes over time and therefore the same does the mean motion $n = \sqrt{\mu/a^3}$. The orbit angular acceleration $\dot{n}$ is in the order of n^3. However, $\dot{n}$ is neglected because the thrust is in the same order of magnitude as the electrostatic force, which is in the order of milli-Newtons [12].

For a more convenient form of the EOM for control design, a spherical frame $\mathcal{S} : \{\hat{\boldsymbol{s}}_{\boldsymbol{L}}, \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{s}}_{\boldsymbol{\phi}}\}$ with origin at the tug's center of mass is introduced. Length L is the separation distance between tug and debris, θ is the in-plane rotation angle and ϕ is the out-of-plane rotation angle, as illustrated in Figure 2.1. As this is a 3-2 Euler angle rotation sequence with respect to the Hill frame, the direction cosine matrix (DCM) that maps from $\mathcal{H}$ to $\mathcal{S}$ is

$$[SH] = \begin{bmatrix} \cos\phi\sin\theta & -\cos\phi\cos\theta & -\sin\phi \\ \cos\theta & \sin\theta & 0 \\ \sin\theta\sin\phi & -\cos\theta\sin\phi & \cos\phi \end{bmatrix} \tag{2.7}$$

The relations between the Hill frame and the spherical frame coordinates are

$$L = \sqrt{x^2 + y^2 + z^2} \tag{2.8a}$$

$$\theta = \arctan\left(\frac{x}{-y}\right) \tag{2.8b}$$

$$\phi = \arcsin\left(\frac{-z}{L}\right) \tag{2.8c}$$

and

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [SH]^T \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L\sin\theta\cos\phi \\ -L\cos\theta\cos\phi \\ -L\sin\phi \end{bmatrix} \tag{2.9}$$

Taking two time derivatives of Equation 2.9 and substituting into Equation 2.6 yields the following spherical relative Equations of Motion:

$$\begin{bmatrix} \ddot{L} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \left[\boldsymbol{F}\left(L, \theta, \phi, \dot{L}, \dot{\theta}, \dot{\phi}\right)\right] + [\boldsymbol{G}(L, \theta)]\, {}^{\mathcal{S}}\boldsymbol{u} \tag{2.10}$$

with the total control acceleration vector ${}^{\mathcal{S}}\boldsymbol{u} = [u_L, u_\theta, u_\phi]^T$ expressed in spherical coordinates, and

$$[\boldsymbol{F}] = \begin{bmatrix} \frac{1}{4}L\left(n^2\left(-6\cos(2\theta)\cos^2\phi + 5\cos(2\phi) + 1\right) + 4\dot{\theta}\cos^2(\phi)\left(2n(t) + \dot{\theta}\right) + 4\dot{\phi}^2\right) \\ 3n^2\sin(\theta)\cos(\theta) + 2\dot{\phi}\tan(\phi)\left(n + \dot{\theta}\right) - 2\frac{\dot{L}}{L}\left(n + \dot{\theta}\right) \\ \frac{1}{4}\sin(2\theta)\left(n^2\left(3\cos(2\theta) - 5\right) - 2\dot{\theta}\left(2n + \dot{\theta}\right)\right) - 2\frac{\dot{L}}{L}\dot{\phi} \end{bmatrix} \tag{2.11a}$$

$$[\boldsymbol{G}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{L\cos\phi} & 0 \\ 0 & 0 & -\frac{1}{L} \end{bmatrix} \tag{2.11b}$$

2.2. Relative Motion Control Design

A relative-control algorithm for the EOM in spherical frame is designed in reference [12]. The globally asymptotically stabilizing feedback control is

$$ {}^{\mathcal{S}}\boldsymbol{u} = [\boldsymbol{G}(L,\theta)]^{-1}\left(-[P]\,\dot{\boldsymbol{X}} - [K]\,(\boldsymbol{X} - \boldsymbol{X_r}) - \left[\boldsymbol{F}\left(L,\theta,\phi,\dot{L},\dot{\theta},\dot{\phi}\right)\right]\right) \tag{2.12} $$

where $\boldsymbol{X} = [L,\theta,\phi]^T$ is the state vector, $\boldsymbol{X_r} = [L_r,\theta_r,\phi_r]^T$ is the desired steady-state vector and $[K]$ and $[P]$ are positive definite gain matrices.

Finally, the required thruster control acceleration is computed by isolating variable ${}^{\mathcal{S}}\boldsymbol{u_T}$ in Equation 2.4, resulting in

$$ {}^{\mathcal{S}}\boldsymbol{u_T} = -{}^{\mathcal{S}}\boldsymbol{u} - {}^{\mathcal{S}}\boldsymbol{F_c}\left(\frac{1}{m_T} + \frac{1}{m_D}\right) \tag{2.13} $$

where the feedback term ${}^{\mathcal{S}}\boldsymbol{u}$ is obtained from Equation 2.12 and the feed-forward term of the electrostatic force ${}^{\mathcal{S}}\boldsymbol{F_c}$ is estimated.

2.3. Equilibrium of the Relative Motion Control

If the gain matrices $[K]$ and $[P]$ are selected to be diagonal, the spherical relative EOM in Equation 2.10 decouple as

$$ \ddot{L} + P_L\dot{L} + K_L\,(L - L_r) = 0 \tag{2.14a} $$

$$ \ddot{\theta} + P_\theta\dot{\theta} + K_\theta\,(\theta - \theta_r) = 0 \tag{2.14b} $$

$$ \ddot{\phi} + P_\phi\dot{\phi} + K_\phi\,(\phi - \phi) = 0 \tag{2.14c} $$

The response of the system for each coordinate is equivalent to a simple damped harmonic oscillator. The damped nature and settling time of the response can be controlled by selecting the gains. When the electric potential is estimated, Equation 2.14a becomes

$$ \ddot{L} + P_L\dot{L} + K_L\,(L - L_r) = (F_c - F_{c,\text{est}})\left(\frac{1}{m_T} + \frac{1}{m_D}\right) \tag{2.15} $$

where F_c and $F_{c,\text{est}}$ are the actual and estimated Coulomb force respectively. Reference [12] proofs

$$ K_L \geq \frac{27\mu_{TD}k_c\,|\Delta Q|}{4L_r^3} \tag{2.16} $$

with $\mu_{TD} = m_T^{-1} + m_D^{-1}$ is required to guarantee that an equilibrium solution with a positive value of L exist. Therefore, this ensures that tug and debris spacecraft do not collide. Note that the formulation in Equation 2.16 considers a charge estimation error of $\Delta Q = q_Tq_D - q_{T,\text{est}}q_{D,\text{est}}$. Here, q_T and q_D are the actual charges of tug and debris spacecraft respectively, whereas $q_{T,\text{est}}$ and $q_{D,\text{est}}$ are the estimated charges.

2.4. Electrostatic Force Model

An isolated object in vacuum with charge q has an electric potential V such that

$$V = \frac{q}{C} \tag{2.17}$$

where C is the object's capacitance. When two objects are in close proximity, charge distribution on both objects change due to their mutual capacitance. For example, charges q_1 and q_2 and potentials V_1 and V_2 of two spheres with radii R_1 and R_2 and separation distance L are related by [16]

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = k_c \begin{bmatrix} 1/R_1 & 1/L \\ 1/L & 1/R_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{2.18}$$

where $k_c = 8.988 \times 10^9$ N m^2 C^{-2} is the Coulomb constant. If the two electric potentials are known or estimated from sensors data, charges q_1 and q_2 are obtained by inverting Equation 2.18. Finally, electrostatic force is obtained with Coulomb's law:

$$F_c = k_c \frac{q_1 q_2}{L^2} \tag{2.19}$$

This force acts along the vector connecting the centers of the spheres. The force is repulsive when both spheres are charged to the same polarity and attractive otherwise.

Knowing charge distribution on the two objects is necessary to compute torques. The Multi-Sphere Model (MSM) is proposed because a single sphere cannot model accurately general 3D geometries. The MSM uses a number of spheres to model 3D geometries with any shape and to approximate the charge distribution on the objects [17, 18]. For multiple spheres, the charge to voltage relationship is

$$\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = k_c \begin{bmatrix} 1/R_1 & 1/r_{1,2} & \cdots & 1/r_{1,n} \\ 1/r_{2,1} & 1/R_2 & \cdots & 1/r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/r_{n,1} & 1/r_{n,2} & \cdots & 1/R_n \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{bmatrix} \tag{2.20}$$

where V_i, Q_i and R_i are the potential, charge and radius of the ith sphere respectively, $r_{i,j} = |\boldsymbol{r_{i,j}}|$ and $\boldsymbol{r_{i,j}}$ is the vector from the jth to the ith sphere. Also, Equation 2.20 can be written as

$$\boldsymbol{V} = [S]\,\boldsymbol{Q} \tag{2.21}$$

where $[S]$ is the elastance matrix. For two charged objects modelled with multiple spheres, Equation 2.20 has the form

$$\begin{bmatrix} \boldsymbol{V_1} \\ \boldsymbol{V_2} \end{bmatrix} = \begin{bmatrix} S_1 & S_M \\ S_M^T & S_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q_1} \\ \boldsymbol{Q_2} \end{bmatrix} \tag{2.22}$$

where S_M is the mutual capacitance block in the elastance matrix and changes with the relative position of the two objects. For rigid bodies, diagonal blocks S_1 and S_2 are constant [19, 20]. If the electric potentials are known, the charge of each sphere is computed by inverting Equation 2.20. Then, the total Coulomb force $\boldsymbol{F_c}$ and torque $\boldsymbol{L_c}$ about point O acting on object 1 are

$$\boldsymbol{F_{c_1}} = -k_c \sum_{j=1}^{n_1} Q_{1_j} \left(\sum_{i=1}^{n_2} \frac{Q_{1_i}}{r_{i,j}^3} \boldsymbol{r_{i,j}} \right) \tag{2.23a}$$

$$\boldsymbol{L_{c_{1,O}}} = -k_c \sum_{j=1}^{n_1} \boldsymbol{r_j} \times Q_{1_j} \left(\sum_{i=1}^{n_2} \frac{Q_{1_i}}{r_{i,j}^3} \boldsymbol{r_{i,j}} \right) \tag{2.23b}$$

where $\boldsymbol{r_j}$ is the vector from point O to the jth sphere. For the electrostatic tractor, using Equation 2.23a and Equation 2.23b allows to take into account the debris attitude and rates. For example, debris attitude control can be designed to touchlessly detumble the debris object.

However, tug and debris attitudes are not taken into account in this research project. For simplicity, the tug and debris objects will be modelled as spheres of known radius. Additionally, in this research the spacecraft will be charged at constant and known voltages. Section 2.3.5 of the Literature Study report [13] provides details of a complete spacecraft charging model that can be used in future research.

3

Micro-propulsion Thrusters

This chapter reviews the propulsion system options available to use in the Electrostatic Tractor. Thrust-related parameters of propulsion systems are reviewed, simulated and tested. An example scenario simulation of the Electrostatic Tractor for GEO debris removal helps to determine how several parameters affect the propulsion system. The chapter ends with the conclusion about “the hardware” of the proposed space debris removal concept.

3.1. Propulsion System Types

As deducted in the literature study report [13], the thrust of the Electrostatic Tractor (ET) for GEO debris removal must be in the same order of magnitude as the electrostatic force. The electrostatic force is in the order of milli-Newtons [12]. Therefore, micro-thrusters are necessary to achieve that low level of thrust.

In order to survey the available micro-thruster types and technologies for the ET concept, this section provides an overview on the current state-of-the-art of micro-propulsion systems. The information presented in this section is based on the course reader of A. Cervone [21]. The course reader puts together the work of several research groups [22, 23, 24, 25, 26, 27, 28].

In this research project, micro-thrusters are grouped into three categories: (1) Chemical propulsion, (2) Electric propulsion, and (3) Cold gas and resistojets. Propellant-less propulsion systems, such as solar sails, are not considered in this research because they have a low degree of control and also need to interact with the surrounding environment (e.g., planetary atmosphere). This makes propellant-less propulsion systems too complicated for GEO debris removal with the Electrostatic Tractor.

Chemical propulsion systems for low-thrust propulsion applications are miniaturized versions of traditional mono-propellant, bi-propellant and solid propellant systems. However, at the present time, these propulsion systems are not recommended for the electrostatic tractor due to several reasons. The combustion temperature in bi-propellant systems is normally significantly higher than mono-propellant systems. This causes more significant thermal issues. Solid propellants operate with a thrust of 10's of Newtons, while micro-thrusters working in the milli-Newton level are required for the ET concept. Moreover, once a solid propellant system starts burning its solid fuel, it cannot be stopped. Finally, miniaturized mono-propellant systems may be an option to consider for the ET concept in the future. However, today, no mono-propellant micro-propulsion systems have been demonstrated in orbit [21]. The high dry mass of a chemical propulsion system and the high risk involved in the chemical reaction are major drawbacks in small satellites.

Electric propulsion systems include ion thrusters, Radio-Frequency electric thrusters, Hall Effect thrusters, electrospray thrusters and Pulsed Plasma Thrusters. The inertial thruster requirement No. 2 in Table 3.1 of the Literature Study report [13] states that "Thrusters shall operate at the differential potential between tug and debris spacecraft." In other words, thrusters must be able to operate at high potentials. For our application, this means tens of kilovolts. In general, current low-thrust electric propulsion systems are unable to operate properly in environment with high differential potentials.

At this point, the best micro-propulsion system types for the electrostatic tractor concept removing GEO debris are cold gas systems and (micro-)resistojets. These two types of micro-propulsion systems are examined below.

3.1.1. Cold Gas systems

In cold gas systems, the propellant is stored at high pressure and accelerated in a nozzle without any additional heating or energy input. If the propellant is stored as a liquid, it is vaporized before reaching the nozzle. Given the simple design and extreme simplicity of the concept, the system mass is usually small and a limited number of components are required. Cold gas micro-thrusters are the ones which have been most widely developed and operated in space. Figure 3.1 shows a schematic representation of the working principle of a typical cold gas micro-thruster.

In absence of a pressure regulator, the gaseous propellant storage pressure in the tank continuously decreases while propellant is extracted from it (blow-down operation). Consequently, the thrust level

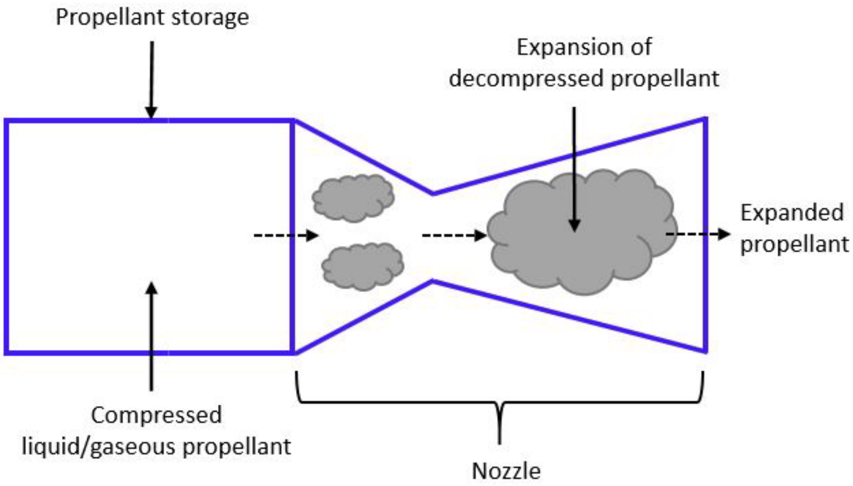

Propellant storage
Expansion of decompressed propellant
Expanded propellant
Compressed liquid/gaseous propellant
Nozzle

Figure 3.1: Working principle of a cold gas micro-thruster [23].

continuously decreases with time during operation in orbit. If the extraction of propellant is not too fast, it can be assumed that the blow-down process is isothermal and, therefore, pressure and mass of propellant in the tank change proportionally. The low temperature of the propellant at the nozzle inlet limits the jet velocity and specific impulse.

Typical propellants for cold gas micro-thrusters are Isobutane, refrigerants, Sulfur Dioxide, Sulfur Hexafluoride, Nitrogen, Argon and Xenon.

3.1.2. Micro-Resistojets

Resistojets can be seen as an intermediate concept between electrical and chemical propulsion since, as schematically shown in Figure 3.2, the propellant is heated electrically (typically by means of a resistance) and accelerated in a convergent-divergent nozzle. In principle, any propellant can be used, stored in any phase (liquid, solid or gaseous); in practice, however, liquid propellants are the most widely used. An alternative to liquid propellants are the so-called *warm gas* thrusters, which are basically cold gas systems allowing for additional (usually limited) heating of the gaseous propellant before being accelerated in the nozzle.

In terms of components and operational characteristics, resistojets are very similar to cold gas thrusters: the propellant is stored in a tank, pressurized (when in the liquid phase) by a pressurant gas and injected in the heating chamber by opening a thrust valve. A pressure regulator is usually not included, meaning that the operation is typically blow-down and the pressure (and thrust) provided by the system are decreasing over its lifetime. The specific impulse, although higher than cold gas systems due to the higher temperature of the propellant and the nozzle inlet, is still limited due to the

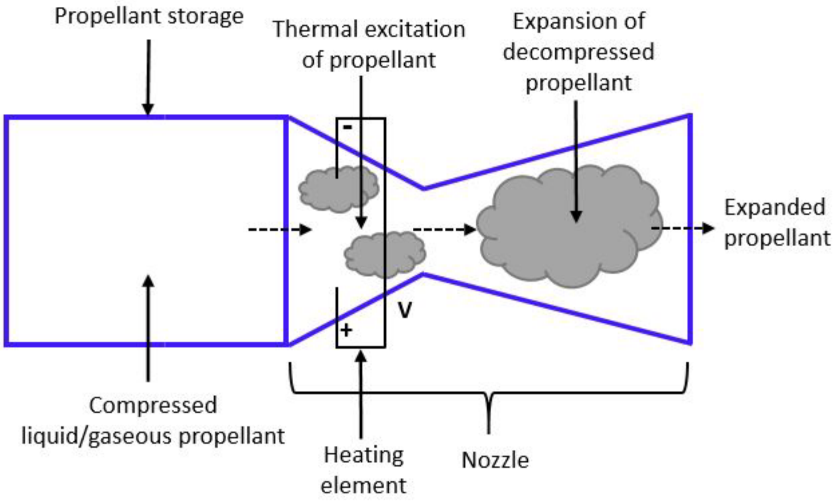

Propellant storage
Thermal excitation of propellant
Expansion of decompressed propellant
-
+
V
Expanded propellant
Compressed liquid/gaseous propellant
Heating element
Nozzle

Figure 3.2: Working principle of a micro-resistojet thruster [23].

limitations in the available heating power.

There is a direct relationship not only between heating power and chamber temperature (as expected), but also between heating power and propellant pressure. This means, in other words, that given a desired temperature at which the propellant has to be heated, the required heating power to achieve that temperature will be a function of the propellant pressure and, therefore, in a system without pressure regulator it will vary over the lifetime of the system. This complicates the design of the control electronics and, at the same time, poses additional limitations on the achievable thrust and specific impulse levels.

3.2. Propulsion System Parameters

Key parameters of a propulsion system are the operating thrust level, the specific impulse, the total impulse, the power consumption and the mass of the whole system. Dimensions of the propulsion system is important too. Besides, another set of parameters for the micro-thrusters in the electrostatic tractor concept are: **thrust bias, noise and resolution**, minimum and maximum thrust, and configuration of existing commercial micro-propulsion systems available. These parameters may have a major impact for the electrostatic tractor because the ET needs precise thrust control. The following sections of this chapter present an initial assessment of the impact of these parameters for GEO debris removal using the electrostatic tractor concept.

3.3. Propulsion System Requirements

Initially, five system requirements for the ET thrusters were defined in Table 3.1 of the Literature Study report [13, 29]. These inertial thruster requirements are:

1. **Thrust Magnitude:** On the same order as the expected tractor force between tug spacecraft and the debris object, which is of 1 to 10 milli-Newtons.
2. **Operate at high potentials:** Thrusters shall operate at the differential potential between tug and debris spacecraft, which is of tens of kilovolts.
3. **Thrust Resolution:** Small and subtle inertial thrust variations of 10-50 micro-Newton are required to achieve the meter level relative positioning.
4. **Minimize momentum and charge flux:** Avoid imparting a momentum or charge flux onto the debris spacecraft which might reduce the effectiveness of the electrostatic tractor.
5. **Long-Term Operation:** The inertial thrusting must run continuously for the 3-4 month reorbiting period.

Section 3.5 will show that Requirement No. 3 (Thrust Resolution) shall be removed.

3.4. Propulsion System Integration and Simulation

The Electrostatic Tractor for space debris removal is simulated in *Basilisk*.[1]A Basilisk C++ module that computes the control thrust force of the Electrostatic Tractor Relative Motion Control is already available [30]. This module implements the model presented in Chapter 2. A servicing satellite and a debris are charged to different electrostatic potentials, resulting in an attractive force between the two craft. The Electrostatic Tractor concept uses this attractive force to tug the debris to another orbit [12]. Figure 3.3 shows an illustration of the Electrostatic Tractor functioning to tug space debris without physical contact.

A basic modeling of thruster limitations (based on parameters of Section 3.2) is added to this C++ module. Here, T is the real thrust and T_{des} is the desired thrust. The following sequential instructions are integrated into the module.

1. Thruster only in one direction $\rightarrow$ if $T_{des} < 0$ then $T := 0$
2. Minimum thrust T_{min} required $\rightarrow$ if $T_{des} < T_{min}$ then $T := 0$
3. Maximum thrust T_{max} available $\rightarrow$ if $T_{des} > T_{max}$ then $T := T_{max}$
4. Thruster resolution $res \rightarrow T := T_{min} + res \cdot \lfloor (T - T_{min}) / res \rceil$. This operation rounds down the value of $T - T_{min}$ to the nearest multiple of res in order to impose the thruster resolution.

[1] *Basilisk* is an open-source software framework capable of both faster-than real-time spacecraft simulations and real-time options for hardware-in-the-loop simulations. More information and download in `https://hanspeterschaub.info/basilisk/`

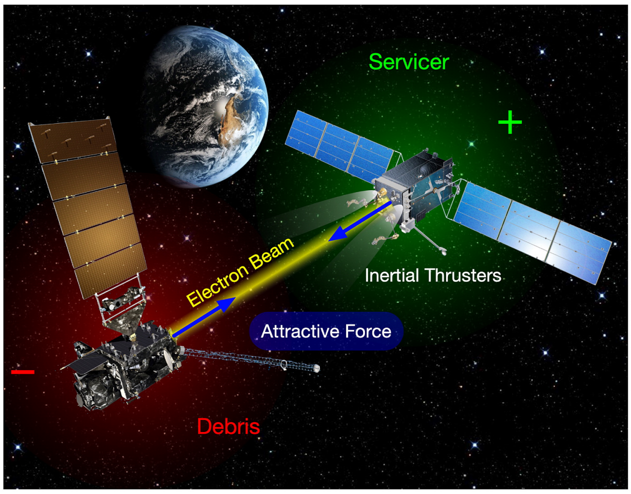


Figure 3.3: Electrostatic tractor in operation [15].

Physical characteristics for thruster bias and noise are included by adding $T := T \cdot (1 + w_1 + b_1) + w_2 + b_2$, where w_1 and w_2 are thrust noise and b_1 and b_2 are thrust bias of the micro-propulsion system. This equation adds biases and noises both in the order of magnitude of the thrust level and independent of that order of magnitude. These biases and noises correspond to the unmodelled effects of the thrusters and the unmodelled physics of the spacecraft.

The simulation is defined in a Python script [31]. This script demonstrates a basic debris reorbit scenario from GEO using the Electrostatic Tractor (ET) concept with a simple proportional-derivative (PD) controller. The example considered here consists of a tug of 500 kilograms, sphere radius of 2 meters and it is charged at 25 kV. The debris object has 2,000 kilograms, sphere radius of 3 meters and it is charged at -25 kV. The electron beam in the tug is responsible to charge both tug and debris spacecraft to the specified potentials. The tug is initially 50 meters ahead the debris, whereas the desired separation distance is set to 30 meters.

The additional physical parameters and considerations in this example are: Thruster limited to operate in the direction to separate tug and debris, minimum thrust $T_{min} = 75$ µN, maximum thrust $T_{max} = 1.45$ mN and thrust resolution $res = 25$ µN. Additionally, noises are $w_1 \sim N\left(0, \sigma = 16.7 \text{ µN/N}\right)$ and $w_2 \sim N\left(0, \sigma = 22.7 \text{ µN}\right)$ and biases are $b_1 = -2.94$ µN/N and $b_2 = 5$ µN. These minimum thrust, maximum thrust and thrust resolution values correspond to the commercial IFM Micro 100 Thruster [32]. Noises and biases were selected randomly.

This example scenario is first simulated as an ideal case, were all the physical constraints set in the previous paragraph are not taken into account. Then, a second simulation is executed using all the physical parameters and considerations defined above. Therefore, the second simulation uses a more realistic model than the ideal case of the first simulation.

The dynamics and controls used in these simulations were defined in Chapter 2. The orbital dynamics are simulated by propagating tug and debris states (position and velocity) using the inertial accelerations obtained in Equation 2.2. The thrust control acceleration applied is computed in Equation 2.13. Finally, the electrostatic force in these simulations is obtained from Equations 2.18 and 2.19.

3.5. Analysis of Simulation Results

From the results of both simulations, the separation distance between tug and debris over time is calculated. This is the most meaningful variable in order to compare the behaviour between the ideal scenario and the scenario with physical constraints.

Figure 3.4 shows the separation distance going from the initial 50 meters to the desired 30 meters. The desired separation distance is reached in less than a tenth of the orbit period.

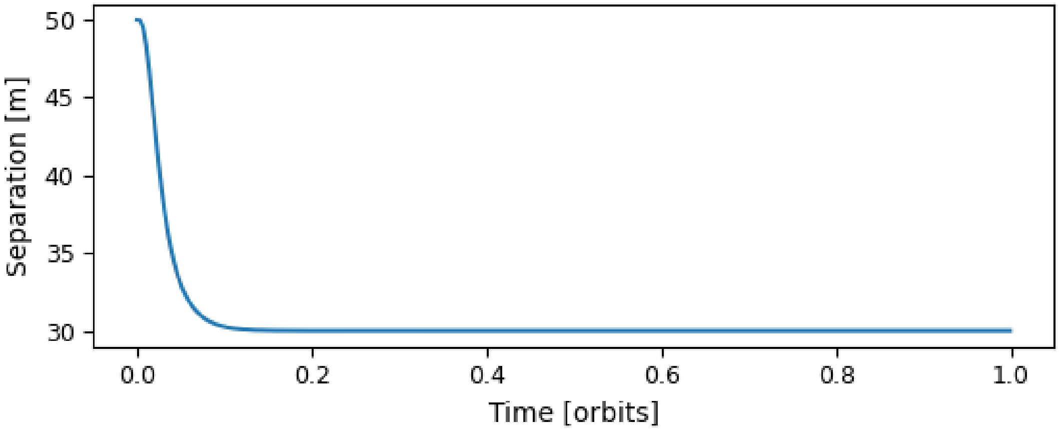


Figure 3.4: Separation distance between tug and debris (without physical constraints).

Figure 3.5 plots the same variable and the shape is very similar to the previous figure. It must be highlighted that the added thrust noise, bias, resolution and minimum and maximum thrust have not really affected the behaviour of the system. These parameters were selected according to the current options available in the market and particularly correspond to the IFM Micro 100 Thruster [32], which has average values of the micro-thrusters commercially available at this moment. It is an advantage that these physical parameters do not really affect the behaviour of the electrostatic tractor in this scenario (GEO debris removal). While only the thrust resolution was a requirement in Section 3.3, both this requirement as well as considerations of noises and biases can be discarded.

Another effect visible when comparing Figure 3.4 and Figure 3.5 is the constraint that states that a single thruster can only fire in one direction, and not the opposite, makes the tug to take more time to get close to the debris. This happens because the Coulomb force is the only one acting in this second scenario. So in Figure 3.5, the desired separation distance of 30 meters is not reached yet in a tenth of the orbit period.

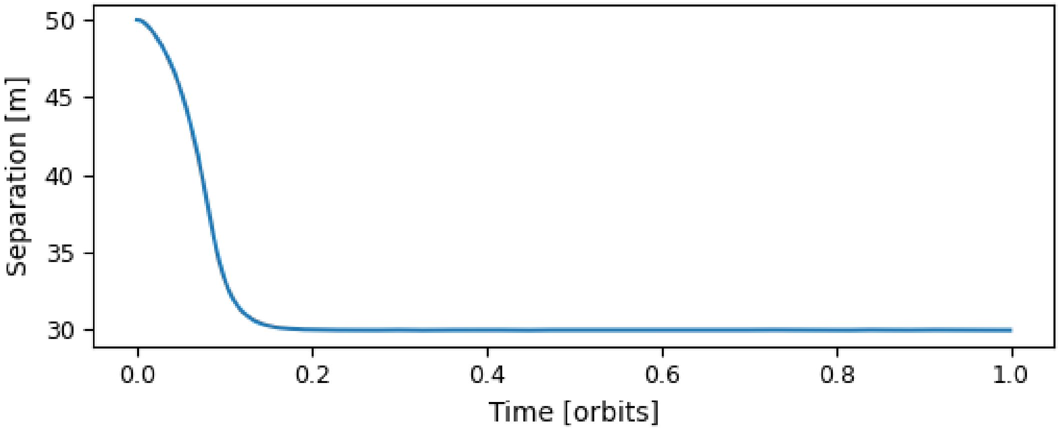


Figure 3.5: Separation distance between tug and debris (with physical constraints).

3.6. Conclusions on the Propulsion System

First, the chapter enumerated the technologies for low-thrust propulsion. It reviewed the two types of micro-thrusters that better meet the propulsion system requirements: cold gas thrusters and micro-resistojet thrusters. Regardless of the thruster used, physical limitations slightly modify the thrust level. The most important are thrust bias, noise and resolution. Propulsion system requirements defined in the Literature Study report are reviewed here too.

Section 3.4 explained how to integrate the physical limitations in the Basilisk software. A general example of the Electrostatic Tractor removing space debris in GEO is proposed. This example is simulated twice: first the ideal scenario where there are no disturbances and then the scenario with them. The analysis of Section 3.5 determined that parameters such as thrust noise, bias and resolution of the thruster do not meaningfully affect the performance and behaviour of the ET for removing GEO debris. However, plots showed that, when there is only one thruster –which can only thrust in one direction (no negative thrust)–, the tug and debris spacecraft take more time to come closer to the desired separation distance. This is because only the attractive Coulomb force is acting at that moment.

For the rest of the thesis, the physical limitations of thrust noise, bias and resolution are ignored when designing the thrust control algorithm. This is because these limitations do not play an important role in the ET scenarios of GEO debris removal. To simplify the rest of the research project, thrusters are considered to have two states only: full on or full off. Additionally, there is no need to specify which commercial micro-thruster is needed, as with the descriptions done in this chapter, there are many commercial cold gas micro-thruster and micro-resistojets that can be used in the Electrostatic Tractor.

4

Discrete Thrust Control Algorithm

In this chapter, the author proposes and formulates a discrete thrust control algorithm for the Electrostatic Tractor trajectory planning problem. The chapter is divided into three sections: First, Section 4.1 discusses the problem and introduces the control algorithm for the Electrostatic Tractor. From there, Section 4.2 mathematically formulates the chosen algorithm—a predictive control optimization problem for linear time-varying systems based on impulsive control signals. Finally, Section 4.3 particularizes the optimization problem to the trajectory planning problem of the Electrostatic Tractor. To do this, the spacecraft rendezvous problem is formulated and the effects of the Coulomb force between spacecraft are modelled and added to the optimization problem.

4.1. Control Algorithm for the Electrostatic Tractor

The relative motion dynamics modelled in Section 2.1 together with the control objective presented there (maintaining debris and tug at a fixed separation distance), makes the Electrostatic Tractor problem equivalent to a spacecraft rendezvous problem. The spacecraft rendezvous problem consists of two nearby spacecraft where one approaches the other. This is usually done in docking operations, where both spacecraft much reach the same point and at no relative velocity in order to make contact. The Electrostatic Problem is the same as the spacecraft rendezvous problem, but both spacecraft must reach and maintain a reference separation distance, always keeping a safety distance to avoid collision.

The spacecraft rendezvous problem, and therefore the Electrostatic Tractor problem, is a trajectory planning problem. For example, when using a body-fixed reference frame in the debris spacecraft, the problem becomes to solve which path has the ET follow (relative to the debris) in order to reach the desired separation distance, subject to constraints. Equation 2.10 shows the equation of motion between tug and debris in spherical coordinates. By expressing this equation in Cartesian coordinates, the Clohessy–Wiltshire equations appear. These equations correspond to a simplified model of orbital relative motion where the target is in a circular orbit. In this chapter, Section 4.3 puts the Clohessy–Wiltshire equations applicable to the ET problem in the form of a linear time-varying system (LTV). This step can be done if the Coulomb force between tug and debris is considered part of the control input, as was expressed in Equation 2.4. The definition of a linear time-varying system is presented in Section 4.2.1.

As mentioned in Chapters 1 and 2 of this thesis report and explained in the Literature Study report [13], thrusters are considered to have two states only: full on or full off. This choice makes the problem easier to handle. Also, the Literature Study report shows that the ET micro-thrusters and the mounted electron beam cannot work at the same time, as thruster flames interfere the electron beam functioning [13, 14]. Therefore, the control input u has form of pulses, which can be modelled with Pulse Width Modulation (PWM). Note that the total control input u includes both the Coulomb force and the thrust, as expressed in Equation 2.4, even though both cannot work simultaneously.

4.2. Impulsive Predictive Controller

Reference [33] explains how to solve the control problem of an LTV system that has PWM control inputs. This method to stabilize an LTV system at $x = 0$ (where x is the state vector of the system) by using PWM control inputs is used in this thesis project (with additional modifications) in order to solve the trajectory planning problem of the Electrostatic Tractor. For simplicity, impulsive control inputs at the beginning of each time interval will be considered. Here, Section 4.2.2 explains how to formulate the problem as a convex quadratic optimization problem and why this is the appropriate method to solve the trajectory planning problem of the Electrostatic Tractor. The algorithm will need to predict future states in order to make a global optimization of present and future states. With all this, now we formulate an Impulsive Predictive Controller for LTV systems.

This section starts with the general solution of a linear time-varying system. Then, the solution is particularized to impulsive control inputs applied at the beginning of each time interval. Considering impulsive control inputs simplifies the mathematical problem while it is a good model of short-duration thrusts. Next, a planning optimization problem over a fixed planning horizon is formulated. For this purpose, several constraints and an objective function are added to the problem. Afterwards, an algorithm is formulated to convert the instant impulses to equivalent pulse width-modulated control signals. The full sequence of the complete control algorithm for LTV systems is then presented. The section ends with a comment on model predictive control.

4.2.1. Linear Time-Varying System Model with Impulsive Control

A linear time-varying system is defined by

$$\dot{x} = A(t)x + B(t)u \tag{4.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input vector, and $A(t)$ and $B(t)$ are $n \times n$ and $n \times m$ matrices respectively, which depend on time $t \geq 0$.

If initial conditions $x(t_0) \in \mathbb{R}^n$ are given at time t_0 and the input is known, the solution to Equation 4.1 for $t \geq t_0$ is

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^{t} \Phi(t, s)B(s)u(s)\,\mathrm{d}s \tag{4.2}$$

where $\Phi(t, t_0)$ is the state transition matrix [34]. This matrix is the unique solution to the differential equation

$$\dot{\Phi}(t, t_0) \;= A(t)\Phi(t, t_0) \text{ for } t > t_0 \tag{4.3a}$$

$$\Phi(t_0, t_0) = I_n \tag{4.3b}$$

In this chapter, time intervals start at t_k and end at $t_{k+1} = t_k + T$, where T is the sample period. Sampling times are $t_k = t_0 + kT$, $k = 0, 1, ...$ Then, Equation 4.2 is rewritten as

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k)x(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, s)B(s)u(s)\,\mathrm{d}s \tag{4.4}$$

Impulsive control

For simplicity, impulsive control inputs at the beginning of each time interval are considered. Therefore, control input is $u(t) = u_k \cdot \delta(t - t_k)$ for $t \in [t_k, t_{k+1})$, where $u_k \in \mathbb{R}^m$ and $\delta(t)$ is Dirac's delta function. Then, Equation 4.4 becomes

$$x_{k+1} = \Phi_{k+1,k} \cdot x_k + \Phi_{k+1,k} \cdot B_k \cdot u_k \tag{4.5}$$

where $x_{k+1} = x(t_{k+1})$, $x_k = x(t_k)$, $B_k = B(t_k)$ and $\Phi_{k+1,k} = \Phi(t_{k+1}, t_k)$.

Compact notation

It is possible to obtain x_{k+j+1} for any $j \geq 0$ when Equation 4.5 is updated to

$$x_{k+j+1} = \Phi_{k+j+1,k} \cdot x_k + \sum_{i=k}^{k+j} \Phi_{k+j+1,i} \cdot B_i \cdot u_i \tag{4.6}$$

where $\Phi_{k+j+1,k} = \Phi_{k+j+1,k+j} \cdot \Phi_{k+j,k+j-1} \cdots \Phi_{k+1,k}$. Also, $\Phi_{k+j+1,k} = \Phi(t_{k+j+1}, t_k)$ is fulfilled.

Now, $X_k \in \mathbb{R}^{n \cdot N_p}$ and $U_k \in \mathbb{R}^{m \cdot N_p}$ are defined as stack vectors of N_p state and input vectors, respectively, such that

$$X_k = \begin{bmatrix} x_{k+1} \\ \vdots \\ x_{k+N_p} \end{bmatrix} \tag{4.7a}$$

$$U_k = \begin{bmatrix} u_k \\ \vdots \\ u_{k+N_p-1} \end{bmatrix} \tag{4.7b}$$

Parameter $N_p \in \mathbb{N}_1$ is the planning horizon. The planning horizon is how many steps ahead the algorithm predicts the future states and takes them into account when optimizing. Note that X_k spans from time t_{k+1} to t_{k+N_p} while U_k spans from time t_k to t_{k+N_p-1}. At this point, one can write that

$$X_k = F_k\, x_k + G_k\, U_k \tag{4.8}$$

where

$$F_k = \begin{bmatrix} \varPhi_{k+1,k} \\ \varPhi_{k+2,k} \\ \vdots \\ \varPhi_{k+N_p,k} \end{bmatrix} \tag{4.9a}$$

$$G_k = \begin{bmatrix} \varPhi_{k+1,k} & 0 & \cdots & 0 \\ \varPhi_{k+2,k} & \varPhi_{k+2,k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \varPhi_{k+N_p,k} & \varPhi_{k+N_p,k+1} & \cdots & \varPhi_{k+N_p,k+N_p-1} \end{bmatrix} \cdot \begin{bmatrix} B_k & 0 & \cdots & 0 \\ 0 & B_{k+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{k+N_p-1} \end{bmatrix} \tag{4.9b}$$

Matrices F_k and G_k have size $n{\cdot}N_p \times n$ and $n{\cdot}N_p \times m{\cdot}N_p$ respectively.

4.2.2. Formulation of the Planning Problem

Next, the planning problem is formulated by introducing constraints and the objective function.

Constraints on the planning problem

The state can be subject to inequality constraints, which can change along the planning horizon. In general, state constraints can be formulated as $C_{S,k}\, X_k \leq d_{S,k}$. Using Equation 4.8, these constraints can be written in terms of inputs as $C_{S,k}\, G_k\, U_k \leq d_{S,k} - C_{S,k}\, F_k\, x_k$.

For an impulsive controller, inputs u_k are limited below and above. Therefore, $\underline{U}_k \leq U_k \leq \overline{U}_k$ where $\underline{U}_k$ and $\overline{U}_k$ are stacked vectors containing the lower and upper bounds of the impulses exerted by each input at sampling times $t_k, t_{k+1}, ..., t_{k+N_p-1}$.

Objective function

The objective function to be minimized, now denoted as J_k, is chosen to be a weighted combination of the 1-norm of the control input and the squared 2-norm of the state, denoted as $J_{U,k}$ and $J_{X,k}$ respectively, both over the planning horizon. Moreover, it is appropriate to include the 1-norm of the control input as this directly reflects the amount of control effort required. This choice allows to globally optimize state (ET trajectory) and control effort (ET fuel consumption) at the same time. When the control input is thrust, $J_{U,k}$ gives an estimation of the total fuel consumption over $t \in \left[t_k, t_{k+N_p}\right)$. Then,

$$J_k = J_{X,k} + \alpha \cdot J_{U,k} \tag{4.10}$$

where α is a positive constant used to give a relative weight between input cost and state error. The greater α is, the lower the control effort is. For the ET it means less fuel consumption but more time needed to reach the desired state. On the other hand, an $\alpha = 0$ will provide the fastest way to reach the desired state regardless of the control effort it will require. Note that α is a tunning parameter that has to be chosen and adjusted experimentally, depending on the system responses obtained in simulations. This objective function makes the desired state to be $x = 0$ (for all components of the state vector).

Next, $J_{X,k}$ is computed as

$$J_{X,k} = X_k^T \, Q_k \, X_k \tag{4.11}$$

where Q_k is a positive-semidefinite matrix. In this research Q_k is fixed to the identity matrix. Using Equation 4.8, one can write

$$J_{X,k} = x_k^T \, F_k^T \, Q_k \, F_k \, x_k + 2 \, x_k^T \, F_k^T \, Q_k \, G_k \, U_k + U_k^T \, G_k^T \, Q_k \, G_k \, U_k \tag{4.12}$$

Notice that the term $x_k^T \, F_k^T \, Q_k \, F_k \, x_k$ is constant and can be neglected. Therefore, the state error is redefined as $J_{X,k}^{'} = 2 \, x_k^T \, F_k^T \, Q_k \, G_k \, U_k + U_k^T \, G_k^T \, Q_k \, G_k \, U_k$.

Finally, for impulsive control inputs, $J_{U,k}$ is computed as

$$J_{U,k} = \sum_{j=k}^{k+N_p-1} \|u_j\|_1 = \|U_k\|_1 \tag{4.13}$$

The objective function to be minimized can be now defined as $J_k^{'} = J_{X,k}^{'} + \alpha \cdot J_{U,k}$.

Planning optimization problem

At this point, one can formulate the planning optimization problem over a planning horizon N_p, starting from initial conditions x_k at time t_k. This problem is formulated as

$$\begin{aligned} \min_{U_k} \quad & 2\, x_k^T\, F_k^T\, Q_k\, G_k\, U_k + U_k^T\, G_k^T\, Q_k\, G_k\, U_k + \alpha\, \|U_k\|_1 \\ \text{s.t.} \quad & C_{S,k}\, G_k\, U_k \leq d_{S,k} - C_{S,k}\, F_k\, x_k \\ & \underline{U}_k \leq U_k \leq \overline{U}_k \end{aligned} \tag{4.14}$$

Note that $G_k^T\, Q_k\, G_k$ must be ensured to be a positive-semidefinite matrix. In order to avoid the 1-norm term in the objective function in Problem (4.14), our research defines a new optimization problem where

$$U_k^{'} = \begin{bmatrix} U_k^+ \\ U_k^- \end{bmatrix} \geq 0 \text{ with } U_k = \underbrace{\begin{bmatrix} I_{(m\cdot N_p)}, & -I_{(m\cdot N_p)} \end{bmatrix}}_{P} \cdot U_k^{'} = U_k^+ - U_k^- \tag{4.15a}$$

$$\overline{U}_k^{'} = \begin{bmatrix} \overline{U}_k^+ \\ \overline{U}_k^- \end{bmatrix} = \begin{bmatrix} \overline{U}_k \\ -\underline{U}_k \end{bmatrix} \geq 0 \tag{4.15b}$$

Note that with this formulation we are requiring that $\underline{U}_k \leq 0 \leq \overline{U}_k$.

The final optimization problem is formulated as

$$\begin{aligned} \min_{U_k^{'}} \quad & \left(2\, x_k^T\, F_k^T\, Q_k\, G_k\, P + [\alpha, ..., \alpha]\right) U_k^{'} + U_k^{'T} \left(P^T\, G_k^T\, Q_k\, G_k\, P\right) U_k^{'} \\ \text{s.t.} \quad & C_{S,k}\, G_k\, P \cdot U_k^{'} \leq d_{S,k} - C_{S,k}\, F_k\, x_k \\ & 0 \leq U_k^{'} \leq \overline{U}_k^{'} \end{aligned} \tag{4.16}$$

Note that the solution will satisfy $U_{k,i}^+ \cdot U_{k,i}^- = 0$ for $\forall i \in \{1, ..., m\cdot N_p\}$. From solution vector $U_k^{'}$, vector $U_k = U_k^+ - U_k^-$ is computed, thus obtaining the sequence of control inputs $u_k, u_{k+1}, ..., u_{k+N_p-1}$ that must be applied at times $t_k, t_{k+1}, ..., t_{k+N_p-1}$, respectively.

The choice of objective function made in this section leads to a convex Quadratic Programming (QP) problem, as expressed in Problem (4.16). Therefore, an algorithm to solve convex QP problems is required. General QP problems can be solved with several methods, such as interior-point methods, active set, augmented Lagrangian, conjugate gradient, gradient projection, and extensions of the simplex algorithm [35, 36]. Problem (4.16) is a convex QP problem with linear constraints only, also known as a non-quadratically constrained convex QP problem. For these problems, using either the interior-point method (also known as barrier method) or the simplex method is the common approach.

The results when choosing the barrier method or the simplex method to solve Problem (4.16) should be very similar in both cases. While the internal steps of these methods are not relevant to this research project, it is useful to highlight their features to know which algorithm to choose. Simplex and barrier optimizers work well on different types of problems. The barrier optimizer works well on large sparse problems. In contrast, the simplex optimizers will probably perform better on problems where the constraint matrix and the resulting Cholesky factor are relatively dense. Also, the barrier optimizer is easier to parallelize.

Simulations in Chapter 5 use the Gurobi Optimizer in Matlab. For non-quadratically constrained convex QP problems, Gurobi can be executed using the barrier method (interior-point method) or the simplex method. We select the barrier method run in parallel, which is the default method in Gurobi for this type of problems [37].

4.2.3. Impulsive-to-PWM Algorithm

The control input was modelled as $u(t) = u_k \cdot \delta(t - t_k)$ for $t \in [t_k, t_{k+1})$. However, thrusters are unable to provide an instant impulse. Therefore, as explained in reference [33], here we choose to convert each control input to an equivalent pulse width-modulated (PWM) control signal. These will have the shape shown in Figure 4.1.

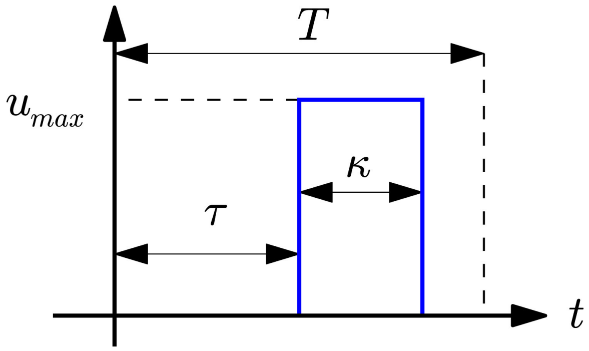


Figure 4.1: PWM variables [33].

As mentioned earlier, impulses were considered at the beginning of each time interval. Therefore, the equivalent PWM pulses will start at the beginning of each time interval too ($\tau = 0$). Then, each pulse time width $\kappa_i^+ \geq 0$ or $\kappa_i^- \geq 0$ is obtained from $\kappa_i^+ = U_{k,i}^+ / u_{max,i}^+$ or $\kappa_i^- = U_{k,i}^- / u_{max,i}^-$, respectively, for $\forall i \in \{1, ..., m{\cdot}N_p\}$. In these expressions, all $u_{max,i}^+ \geq 0$ and $u_{max,i}^- \geq 0$ are parameters set in the problem statement.

In order to keep the equivalence between impulses and PWM pulses, pulse time widths $\kappa << T$. Therefore, maximum pulse time widths $\kappa_{max,i}^+ \geq 0$ and $\kappa_{max,i}^- \geq 0$ must be set. Consequently, $\overline{U}_k^{'}$ –defined in Equation 4.15b– is obtained from $\overline{U}_{k,i}^+ = \kappa_{max,i}^+ \cdot u_{max,i}^+$ and $\overline{U}_{k,i}^- = \kappa_{max,i}^- \cdot u_{max,i}^-$.

4.2.4. Full Algorithm Sequence

The full algorithm sequence of this Impulsive Predictive Controller for any dynamic system that is modelled as an LTV system $\dot{x} = A(t)x + B(t)u$ is as follows:

STEP 1. Read current state x_k at current time t_k.

STEP 2.1. Formulate Optimization Problem (4.16). To do so, the following elements are needed:

(i) Controller parameters T, N_p, $\kappa^{+/-}_{max,i}$ and α. These tuning parameters are set by the user and must satisfy that $T > 0$; $N_p \in \mathbb{N}_1$; $T \geq \kappa^{+/-}_{max,i} \geq 0$; and $\alpha \geq 0$.

(ii) LTV model of the system. The model is fully defined by $A(t)$ and $B(t)$ matrices for $t \geq t_k$. Alternatively, the state transition matrix $\Phi(t, t_0)$ for $t \geq t_0 \geq t_k$ as defined by Equation 4.3 can be provided instead of $A(t)$ matrix. Similarly, only $B(t)$ at times $t_k, t_{k+1}, ..., t_{k+N_p-1}$ are needed.

(iii) State constraints $C_{S,k}\, X_k \leq d_{S,k}$. These linear constraints enforce requirements defined by the user to the state variables.

(iv) Lower and upper bounds $u^{+/-}_{max,i} \geq 0$ for control inputs.

(v) Current state x_k obtained in Step 1.

The LTV model (ii), the imposed state constraints (iii), and the calculation of the lower and upper bounds for control inputs (iv) specific to the Electrostatic Tractor problem will be defined in Section 4.3.

STEP 2.2. Solve Optimization Problem (4.16). To do so, an existing algorithm that solves non-quadratically constrained convex QP problems is executed for Problem (4.16). As explained in Section 4.2.2, using either the interior-point method (barrier method) or the simplex method is the common approach. In our simulations we use the barrier method run in parallel.

STEP 2.3. Calculate times $\kappa_i^+ = U^+_{k,i}/u^+_{max,i}$ and $\kappa_i^- = U^-_{k,i}/u^-_{max,i}$ for the current time t_k, which means for $i = 1, ..., m$ only. Note that $\kappa_i^+ \kappa_i^- = 0$ where $\kappa_i^+ \geq 0$ and $\kappa_i^- \geq 0$. Define $\kappa_i = \max\{\kappa_i^+, \kappa_i^-\}$.

STEP 3. Turn on control input $i = 1, ..., m$ for κ_i seconds. Then turn off. Turning on control input i means it takes value $u^+_{max,i}$ if $\kappa_i^+ > 0$ or value $-u^-_{max,i}$ if $\kappa_i^- > 0$. Turning off means it takes value 0.

STEP 4. Wait until next sampling time $t_{k+1} = t_k + T$. When t_{k+1} is reached (so T seconds have elapsed since Step 1), set $t_k := t_{k+1}$ and go to **STEP 1**.

The loop introduced by Step 4 makes Problem (4.16) to be formulated and solved at each time step. By repeatedly solving this optimization problem and applying the beginning of the result (corresponding to the current time t_k), we establish a feedback that closes the loop.

Additionally, it is possible to solve Problem (4.16) faster if $U_{k+1}^{'}$ is initialized by using the values of the previous solution $U_{k}^{'}$. Values corresponding to the time intervals starting at t_{k+1} to t_{k+N_p-1} in $U_{k+1}^{'}$ are set equal to their corresponding values obtained in $U_{k}^{'}$. Values corresponding to the time interval starting at t_{k+N_p} (the last interval in $U_{k+1}^{'}$) can be guessed by using some simple control law.

4.2.5. Model Predictive Control

The controller for the Electrostatic Tractor needs to control a dynamical system over a period of time such that an objective function is optimized. The ET controller must maintain the desired separation distance with the target while minimizing fuel consumption and satisfying some constraints. Therefore, an optimal control method is likely the best control method for our problem.

Predictive controllers are a group of model-based controllers that predict future states. A model of the system is required in order to design a controller for the system. Predictive control is not the only model-based control design method available. Other model-based controllers are pole-placement and linear quaratic (LQ) control methods. Since our problem needs constraints, Model Predictive Control (MPC) is the best option. MPC is a type of optimal control.

The theory of optimal control is concerned with operating a dynamic system at minimum cost. We say we have a linear quadratic (LQ) problem when the system dynamics are described by a set of linear differential equations (including linear time-varying systems) and the cost is described by a quadratic function. In case that the system dynamics are described by linear time-invariant (LTI) equations only, the solution is called a linear–quadratic regulator (LQR). It is common to add linear constraints that restrict the values that the state variables and control inputs can take. Then, we will talk of a constrained LQ problem (for LTV systems) or a constrained LQR (for LTI systems). Therefore, we see that the formulated Problem (4.16) likely corresponds to a finite-horizon discrete-time constrained LQ optimal control problem. Then, since this problem is solved repeatedly with a receding horizon, the full control algorithm designed in this section is a form of Model Predictive Control.

Although the explanation given above is appropriate, Problem (4.16) is technically not a LQ optimal control problem as typically defined. This is the case because its objective function $J_{k}^{'}$ contains the 1-norm of the control input, while the classical definition of the LQ problem does not. Therefore, we will simply say that the designed Impulsive Predictive Controller uses model predictive control.

As explained in Section 4.2.2, model predictive control allows to define the objective function as a weighted combination of the 1-norm of the control input and the squared 2-norm of the state over the planning horizon. Using the 1-norm of the control input –instead of its squared 2-norm–, directly reflects the amount of control effort required, which for the Electrostatic Tractor corresponds to the amount of fuel used. Finally, using the aforementioned weighted combination allows to globally optimize state (ET trajectory) and control effort (ET fuel consumption) simultaneously.

4.3. Application to the Electrostatic Tractor

In this section, this research work particularizes the problem presented in Section 4.2 to the Electrostatic Tractor problem. First, the spacecraft rendezvous problem is formulated, including the state transition matrix of the system. Then, the effects of the Coulomb force between spacecraft are modelled and added to the problem.

4.3.1. Spacecraft Rendezvous Model

In order to solve the spacecraft rendezvous problem, the Local-Vertical-Local-Horizontal (LVLH) frame in Figure 4.2 is considered.

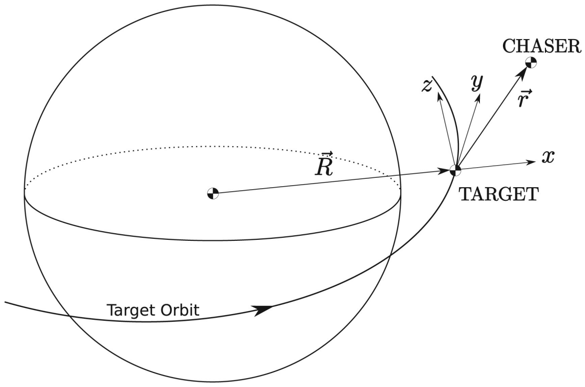


Figure 4.2: Target Local-Vertical-Local-Horizontal frame [33].

Then, the state vector at time t_k is defined as $x_k = [r_{k,x}, r_{k,y}, r_{k,z}, v_{k,x}, v_{k,y}, v_{k,z}]^T$, where $r_k = (r_{k,x}, r_{k,y}, r_{k,z})$ and $v_k = (v_{k,x}, v_{k,y}, v_{k,z})$ are, respectively, the relative position and relative velocity of the chaser with respect to the target. The control input vector at time t_k is defined as $u_k = [u_{k,x}, u_{k,y}, u_{k,z}]^T$. As defined in Section 4.2.1, u_k is the instant impulse vector applied at time t_k. The input vector has units of velocity. These vectors are written in target's LVLH coordinate frame system.

The discretized spacecraft rendezvous problem can be written as

$$x_{k+1} = \mathit{\Phi}_{k+1,k} \cdot x + B_k \cdot u_k \tag{4.17}$$

where $B_k = \begin{bmatrix} 0_3 & I_3 \end{bmatrix}^T$ is a constant 6×3 matrix. For a circular orbit, $\mathit{\Phi}$ is the state transition matrix:

$$\Phi(t,t_0) = \begin{bmatrix} 4-3\,c & 0 & 0 & \frac{s}{n} & \frac{2\,(1-c)}{n} & 0 \\ 6\,(s-n\,(t-t_0)) & 1 & 0 & \frac{2\,(c-1)}{n} & -3\,(t-t_0)+\frac{4\,s}{n} & 0 \\ 0 & 0 & c & 0 & 0 & \frac{s}{n} \\ 3\,n\,s & 0 & 0 & c & 2\,s & 0 \\ 6\,n\,(c-1) & 0 & 0 & -2\,s & 4\,c-3 & 0 \\ 0 & 0 & -s\,n & 0 & 0 & c \end{bmatrix} \tag{4.18}$$

where $s = \sin(n\,(t-t_0))$, $c = \cos(n\,(t-t_0))$ and the orbital motion is $n = \sqrt{\mu/a^3}$, where μ is the gravitational parameter of the central body and a is the semi-major axis of the orbit.

Now, from B_k, Φ and considering sampling times $t_k = t_0 + kT$, $k = 0, 1, ...$, it is possible to construct matrices F_k and G_k as defined in Equations 4.9a and 4.9b, respectively. Also, it is possible to add the following state constraint in order to make sure that the chaser in Figure 4.2 is never behind the target (so $r_y \geq 0$):

$$C_{S,k} = \begin{bmatrix} c_S & & \\ & \ddots & \\ & & c_S \end{bmatrix} \text{ and } d_{S,k} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.19}$$

where $c_S = [0, -1, 0, 0, 0, 0]$ so $C_{S,k}\,X_k \leq d_{S,k}$. Reference [38] shows more complicated constraints that could be considered. The other constraints to add are $0 \leq U_k^{'} \leq \overline{U}_k^{'}$. For example, if each of the six thrusters necessary for 3D motion provides an acceleration of $u_{max,i} = 1$ m/s^2 to the chaser, the sampling period T is 60 seconds and the maximum pulse width $\kappa_{max,i}$ for all control signals is 10 seconds, then $\overline{U}_{k,i} = \kappa_{max,i} \cdot u_{max,i} = 10$ m/s for $\forall i \in \{1, ..., 3{\cdot}N_p\}$. This means that each thruster is limited to provide a maximum delta-V of 10 m/s by thrusting up to 10 seconds at nominal 1 m/s^2, starting at the beginning of each 60-second cycle.

Finally, matrix Q_k can be set to the identity matrix. With all this, Problem 4.16 can be solved as a closed-loop MPC controller as explained in Section 4.2.5. This way, the classical spacecraft rendezvous problem is solved using an optimal controller.

4.3.2. Problem Formulation with Contactless Active Charging Control

In order to use the impulsive predictive controller to solve the Electrostatic Tractor problem, the electrostatic force between the tug and debris spacecraft must be considered. Before, the state vector is redefined as $x_k = [r_{k,x}, r_{k,y} - L_r, r_{k,z}, v_{k,x}, v_{k,y}, v_{k,z}]^T$ where $L_r > 0$ is the desired steady-state separation distance between tug and debris spacecraft.

The electrostatic force is obtained with Coulomb's law:

$$F_c = k_c \frac{q_1 q_2}{L^2} \tag{4.20}$$

where $k_c = 8.988 \times 10^9$ N m^2 C^{-2} is the Coulomb constant, q_1 and q_2 are the charges of tug and debris spacecraft, and L is the distance between them. This force is approximated to act along the vector connecting the centers of the spacecraft. This is because the distance vectors $\boldsymbol{r_{i,j}}$ in Equation 2.23a, which calculates the Coulomb force, are very similar among them due to the safety separation distance between spacecraft. Therefore, instead of multiple distance vectors, an average distance vector is considered. Also, the controller handles the relative distance between spacecraft continuously. Therefore, torques that may be generated due to the Coulomb force not acting exactly in the center of mass of the debris spacecraft can be ignored, as they do not heavily affect the distance control if sufficient separation distance is kept. As mentioned in Section 2.4, using spheres as spacecraft in this research project is a good initial approach, as it is sufficient for the ET thrust control algorithm to work as Chapter 5 demonstrates. The force is repulsive when both spheres are charged to the same polarity and attractive otherwise.

Equation 4.15a defined the solution vector $U_k^{'}$. Now, a matrix $P^{''}$ and the following modified solution vector $U_k^{''}$ are used instead:

$$U_k^{''} = \begin{bmatrix} U_k^+ \\ U_k^- \\ U_k^C \end{bmatrix} \text{ with } U_k = \underbrace{\begin{bmatrix} I_{(m \cdot N_p)}, & -I_{(m \cdot N_p)}, & I_{(m \cdot N_p)} \end{bmatrix}}_{P''} \cdot U_k^{''} = U_k^+ - U_k^- + U_k^C \tag{4.21}$$

Here, $U_k^C \in \mathbb{R}^{m \cdot N_p}$ is a stack vector representing the effect of the Coulomb force from time t_k to t_{k+N_p-1} such that

$$U_k^C = \begin{bmatrix} u_k^C \\ \vdots \\ u_{k+N_p-1}^C \end{bmatrix} \tag{4.22}$$

where u_k^C is the delta-V vector that the Coulomb force produces in cycle k, of period T. In this problem, it is considered that the electron beam that charges both spacecraft can only be used after the $\kappa_{max,i} = 10$ seconds of thrusting. Also, the Coulomb force vector $\boldsymbol{F_{c,k}}$ is considered to be constant during the rest of the cycle. Consequently,

$$u_k^C = (T - \kappa_{max,i}) \cdot \boldsymbol{F_{c,k}} \left(\frac{1}{m_T} + \frac{1}{m_D} \right) \tag{4.23}$$

where m_T and m_D are masses of tug and debris, and force vector $\boldsymbol{F_{c,k}} = F_{c,k} \frac{r_k}{\|r_k\|}$. Note that for the Electrostatic Tractor problem, F_c is a negative number because both spacecraft are charged with opposite polarity.

Finally, all $F_{c,k}$ are obtained from global state vector X_k and using Equation 4.20. New input constraints are added to the problem in order to fix the components of U_k^C in $U_k^{''}$ to the computed values. The problem is iterated until it converges to a solution.

4.3.3. Modified Full Algorithm Sequence

The modified full algorithm sequence of this Impulsive Predictive Controller for the Electrostatic Tractor, which is modelled as an LTV system, is as follows:

STEP 1. Read current relative position $[r_{k,x}, r_{k,y}, r_{k,z}]^T$ and current relative velocity $[v_{k,x}, v_{k,y}, v_{k,z}]^T$ at current time t_k. Define current state $x_k = [r_{k,x}, r_{k,y} - L_r, r_{k,z}, v_{k,x}, v_{k,y}, v_{k,z}]^T$ where $L_r > 0$ is the desired steady-state separation distance between tug and debris spacecraft set by the user.

STEP 2.1. Formulate Optimization Problem (4.16) with additional modifications, using:

(i) Controller parameters T, N_p, κ_{max} and α. These tuning parameters are set by the user and must satisfy that $T > 0$; $N_p \in \mathbb{N}_1$; $T \geq \kappa_{max} \geq 0$; and $\alpha \geq 0$. Also, a new threshold $\beta > 0$ is defined by the user. This parameter β should be smaller than thrusters response time.

(ii) LTV model of the system. The ET model is defined by the state transition matrix $\Phi(t, t_0)$ as defined in Equation 4.18 and by $B(t) = \begin{bmatrix} 0_3 & I_3 \end{bmatrix}^T$.

(iii) State constraints $C_{S,k}\, X_k \leq d_{S,k}$ defined in Equation 4.19.

(iv) Lower and upper bounds $u^{+/-}_{max,i} \geq 0$ for maximum thruster acceleration on each direction.

(v) Current state x_k obtained in Step 1.

(vi) **New Step: Predicted Coulomb acceleration vectors** $u^C_k, ..., u^C_{k+N_p-1}$**.** These vectors are calculated using Equation 4.23. To use this equation, it is necessary to calculate the global state vector X_k using Equation 4.8. Equation 4.8 requires the stacked input vector U_k. We use U_k obtained in the previous iteration of the inner loop or 0 if this is the first iteration of the inner loop for t_k.

STEP 2.2. Solve Optimization Problem (4.16) as explained in Sections 4.2.2 and 4.2.4.

STEP 2.3. Calculate times $\kappa^+_i = U^+_{k,i}/u^+_{max,i}$ and $\kappa^-_i = U^-_{k,i}/u^-_{max,i}$ for the current time t_k, which means for $i = 1, ..., m$ only. Note that $\kappa^+_i \kappa^-_i = 0$ where $\kappa^+_i \geq 0$ and $\kappa^-_i \geq 0$. Define $\kappa_i = \max\{\kappa^+_i, \kappa^-_i\}$.

STEP 2.4. **New Step:** If this is the first iteration of the inner loop for t_k, or if the maximum number of iterations of the inner loop for t_k has not been reached and $\sum_{i=1}^{m} \left(\kappa^{+/-}_{old,i} - \kappa^{+/-}_i\right)^2 \geq \beta$, then set $\kappa^{+/-}_{old,i} := \kappa^{+/-}_i$ and go to **STEP 2.1.vi** *(inner loop)*. Otherwise continue.

STEP 3. Turn on control input $i = 1, ..., m$ for κ_i seconds. Then turn off. Turning on control input i means it takes value $u^+_{max,i}$ if $\kappa^+_i > 0$ or value $-u^-_{max,i}$ if $\kappa^-_i > 0$. Turning off means it takes value 0.

STEP 4. Wait until next sampling time $t_{k+1} = t_k + T$. When t_{k+1} is reached (so T seconds have elapsed since Step 1), set $t_k := t_{k+1}$ and go to **STEP 1** *(outer loop)*.

5 Simulations and Results

Appendix A contains the code that has been implemented in this research project. The code does a simulation based on the physics modelled in Chapter 2 and the discrete thrust control algorithm designed in Chapter 4. This chapter provides and analyzes the simulation results by executing several cases. First, we give an explanation on how the code is verified and validated. Second, different example scenarios for the Geosynchronous Large Debris Reorbiter concept are presented. Third, simulations with tuned values for controller parameters are carried out. Fourth, different combinations (or configurations) of micro-thrusters put in the Electrostatic Tractor are proposed to be simulated. Finally, results obtained from the simulations of the different scenarios, tuned controller parameters and multiple micro-thrusters configurations are shown, plotted and analyzed.

5.1. Verification and Validation

The final algorithm proposed in Chapter 4 is implemented in Matlab. In order to verify that the specifications are correctly implemented by this code, a software verification procedure is carried out. Since the author was the only person available to verify the code, simple verification methods had to be used. On the one hand, the author reviewed the code to detect bad coding practices and adhere to code conventions. Simultaneously, the author performed desk checks. A desk check is a manual technique of reviewing source code without executing the code. Here, the programmer carefully follows the logic and calculations of the code while acting as the computer. This helps to check the correctness of the code, especially by checking what logical sequence would be executed and what values the variables would have at each moment.

On the other hand, the author dynamically verified the code by executing it. The code was debugged while running tests. Single functions were tested with unit tests. Unit tests automatically check whether the output of a function to a given input is correct. For larger pieces of code or complex functions, such as the code that solves the optimization problem, the system and integration test results were manually analyzed to see if they made sense and were close to values obtained in previous studies.

The code is validated by analyzing whether the results obtained meet what was originally desired. Section 5.3 presents and analyzes the results obtained when running this code.

5.2. Simulation Parameters

5.2.1. Geosynchronous Large Debris Reorbiter Scenarios

The following scenarios consider the spacecraft to be in a geostationary circular orbit with a period of one sidereal day (23 h 56 min 4.0905 s). The tug spacecraft has to pull the debris spacecraft 300 kilometres higher. These parameters correspond to the nominal scenario for which the Geosynchronous Large Debris Reorbiter concept is designed for.

In these simulations, the tug has 500 kilograms, sphere radius of 3 meters and it is charged at 20 kV. The debris object has 1,000 kilograms, sphere radius of 3 meters and it is charged at -20 kV. The tug is equipped with six 15-mN micro-thrusters. Each micro-thruster thrusts in one sense of one axis of tug's body-fixed reference frame. From all run simulations, simulations presented in this report correspond to the most interesting results to analyze. Tables 5.1, 5.2 and 5.3 show the simulation parameters.

Table 5.1: Distance between tug and debris spacecraft.

	Initial distance (L_0)	**Desired distance (L_r)**	**Minimum distance (L_{min})**
Scenario 1	25 m	20 m	15 m
Scenario 2	20 m	20 m	15 m
Scenario 3	15 m	20 m	15 m
Scenario 4	40 m	35 m	15 m
Scenario 5	35 m	35 m	15 m
Scenario 6	30 m	35 m	15 m

Both spacecraft start with no relative velocity. Table 5.1 lists six different scenarios with different initial distance or desired distance between tug and debris. These distances are in the direction of movement only. A constraint similar to Equation 4.19 guarantees a minimum distance of 15 meters for all scenarios at all times.

5.2.2. Controller Parameters

Table 5.2 shows different sets of values for parameters that tune the control algorithm designed in the previous chapter. These parameters are the algorithm period T, the planning horizon N_p, the maximum pulse width κ_{max} and the relative weight α between input cost and state error, as were defined in Chapter 4.

Table 5.2: Tuning of control algorithm parameters.

	Algorithm period (T)	**Planning horizon (N_p)**	**Maximum pulse time width (κ_{max})**	**Weight: input cost and state error (α)**
Tuning 1	30 s	20	10 s	10
Tuning 2	60 s	10	20 s	10
Tuning 3	90 s	7	30 s	10
Tuning 4	30 s	20	12 s	10
Tuning 5	30 s	20	8 s	10
Tuning 6	30 s	20	10 s	1
Tuning 7	30 s	20	10 s	1000

5.2.3. Micro-thrusters Configurations

In this section, different combinations of micro-thrusters provide different possible configurations of the thrusting system of the electrostatic tractor spacecraft (the tug). Table 5.3 lists these configurations by specifying the maximum thrust that the thruster can generate on each direction. Thrusters are oriented along the axes of the LVLH coordinate frame system that was previously illustrated in Figure 4.2. Remember that the thrust on $+$**y** axis direction is the one that separates tug and debris objects in the direction of movement of the orbit.

Table 5.3: Micro-thrusters configurations on the ET: Maximum thrust on each direction.

	$+$**x** / $-$**x**	$+$**y** / $-$**y**	$+$**z** / $-$**z**
Config. 1	15 mN / 15 mN	15 mN / 15 mN	15 mN / 15 mN
Config. 2	– / –	15 mN / 15 mN	– / –
Config. 3	– / –	15 mN / –	– / –
Config. 4	– / –	1 mN / –	– / –

The dash “–” means there is no thruster on that direction.

5.3. Analysis of Results

5.3.1. Multiple Scenarios

The six scenarios listed in Table 5.1 are simulated for a 24-hour period each. To start with, these simulations use the parameters in rows “Tuning 1” and “Configuration 1” from Tables 5.2 and 5.3, respectively. Figure 5.1 shows the evolution of the distance between tug and debris for the first 3 hours. The initially unstable tug-debris systems (Scenarios 1, 3, 4 and 6) stabilize after 1 hour approximately. The scenarios experience a pull that brings both spacecraft closer due to the attractive Coulomb force. This effect increases when tug and debris get closer while being already pretty close, such as in Scenario 1. With the designed controller, the obtained response corresponds to a second-order system. These results may slightly differ in reality if the target spacecraft has an irregular shape and spins at a high rate. Nevertheless, the obtained results are good enough for this research. Also, lines in Figure 5.1 are very smooth because the simulation time steps are 30, 60 and 90 seconds as indicated in Table 5.2, while the plot spans to 3 hours.

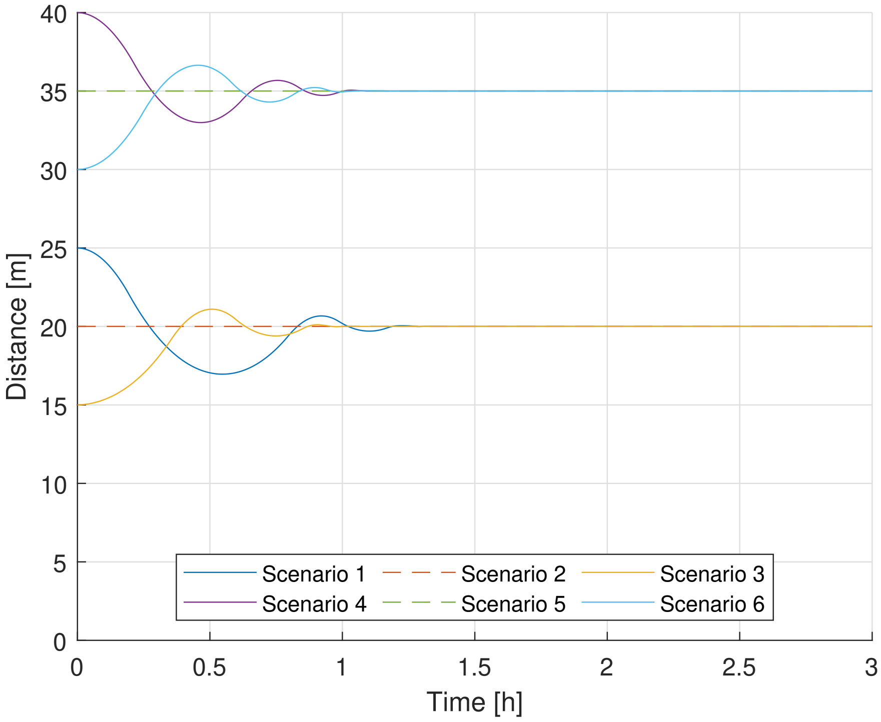


Figure 5.1: Evolution of separation distance between tug and debris spacecraft.

Figure 5.2 plots the change in orbital radius and semi-major axis (SMA) of the debris object. Only Scenario 2 ($L_0 = L_r = 20$ m) and Scenario 5 ($L_0 = L_r = 35$ m) are plotted in the figure as the other four scenarios give the same results. The figure clearly shows that the closer the tug and debris are maintained, the faster the debris is re-orbited. In Scenario 5 ($L_r = 35$ m) the Electrostatic Tractor needs almost 16 months to pull the debris spacecraft 300 kilometres higher while in Scenario 2 ($L_r = 20$ m) it takes about 4.5 months only.

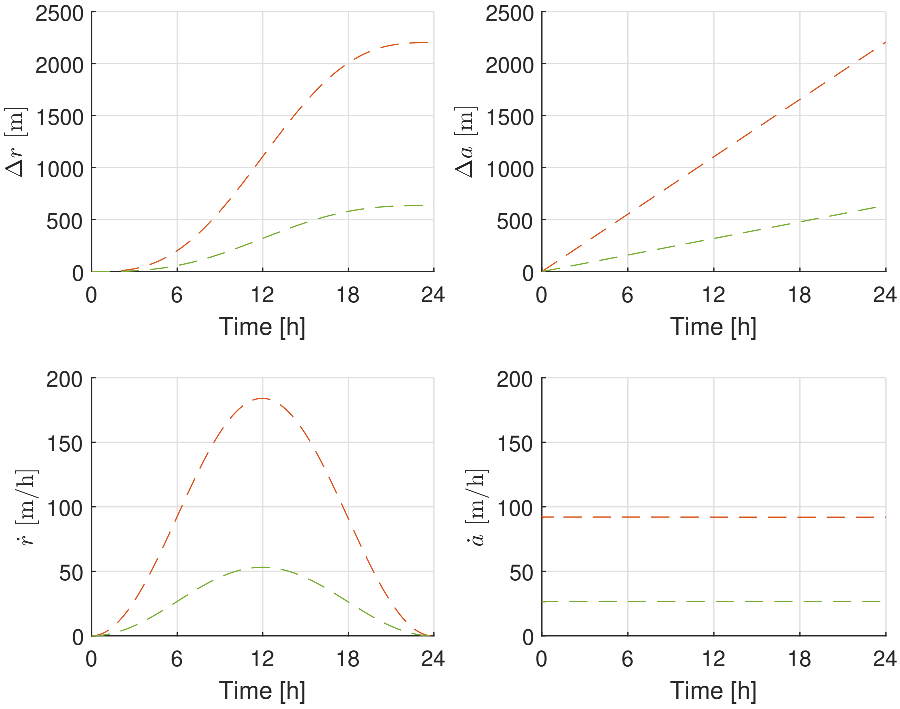


Figure 5.2: Change in orbital radius and semi-major axis of the debris object in scenarios 2 (red-dashed) and 5 (green-dashed).

Reference [8] proofs that the SMA change Δa produced by the Electrostatic Tractor over one orbit revolution can be estimated with the formula

$$\Delta a \approx \frac{4\pi}{n^2} \frac{R_{tug} R_{debris} V_{tug} V_{debris}}{k_c m_{debris} L^2} \tag{5.1}$$

where n is the mean orbital motion, R_{tug} and R_{debris} are the approximated radii of tug and debris respectively, m_{tug} and m_{debris} are the masses of tug and debris respectively, $k_c = 8.988 \times 10^9$ N m^2 C^{-2} is the Coulomb constant and L is the separation distance between tug and debris spacecraft.

According to Equation 5.1, our Scenario 2 would result in a 2.5-kilometer SMA increase per orbit. With the full simulation presented here, the debris experiences a 2.2-kilometer SMA increase per orbit. These values help validate the simulation and controller implementation in this research work. Figure 5.2 also shows that, while the semi-major axis increases at a constant rate, the orbital radius does not. This happens because the orbital eccentricity changes while the debris object is being pulled. However, orbital radius and SMA are the same at the end of the 24 hours (after one orbit).

Figures 5.3a and 5.3b in the next page show the delta-V generated by the thrusters of the Electrostatic Tractor (the tug spacecraft) in Scenarios 1 to 6. The magnitude of delta-V ($\|\Delta V\|$) is linear with the fuel burnt. Therefore, the cumulative $\|\Delta V\|$ provides an indicator of the total fuel consumption at each time. Its derivative $\|\dot{\Delta V}\|$ provides an indicator of the consumption rate. The former indicator is equivalent to the total energy consumed until a specific time, while the latter indicator is equivalent to the consumption power at a specific time.

It is possible to distinguish the transition phase followed by the steady-state phase. The transition phase in Scenarios 1 to 6 lasts between 1 and 1.5 hours. This phase shows an almost linear increase of the cumulative $\|\Delta V\|$ at an average rate of 0.036 m/s per hour. Nevertheless, this delta-V rate ($\|\dot{\Delta V}\|$) has important perturbations during this phase. Also, the delta-V rate is greater for the first minutes: it is 0.072 m/s per hour.

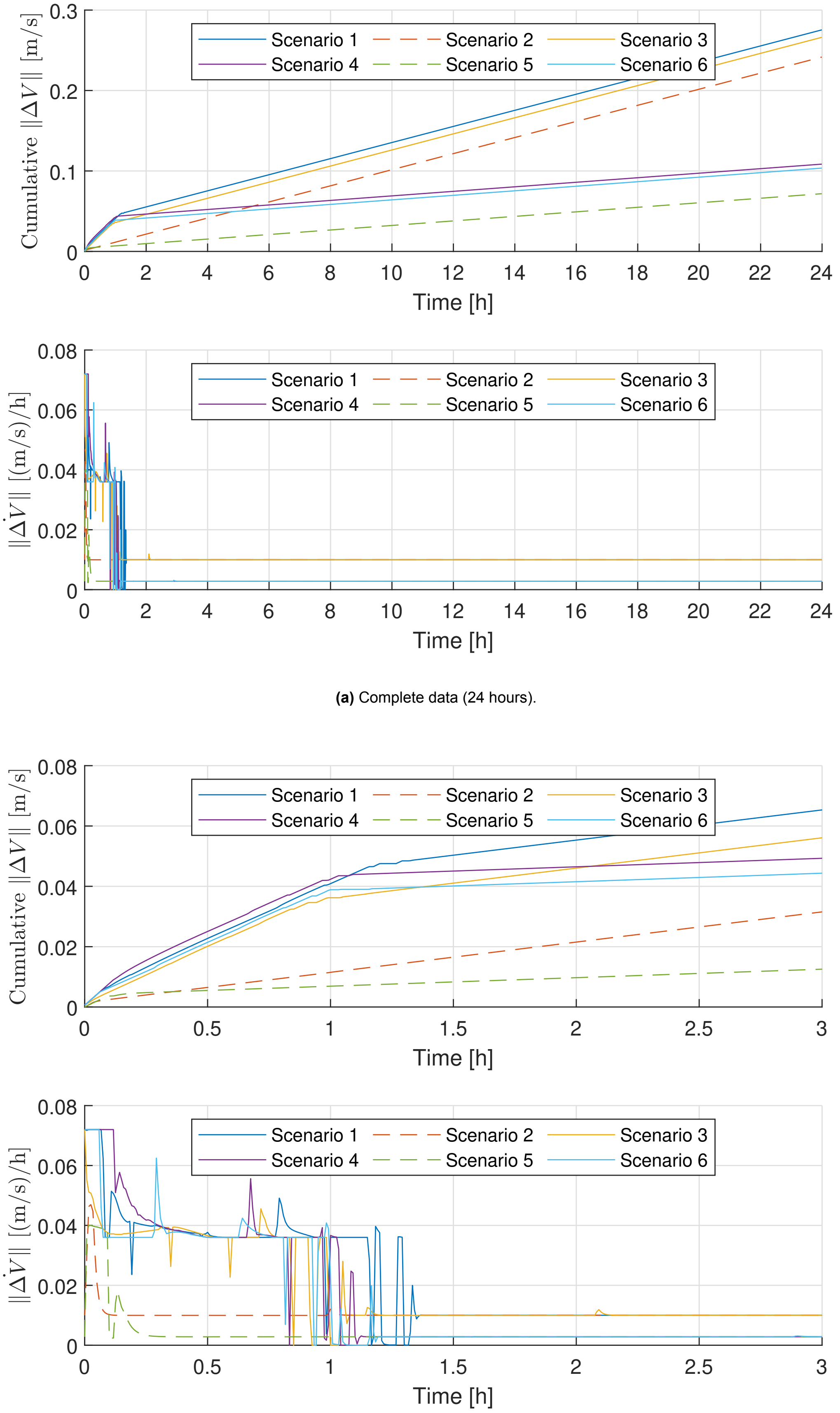
The steady-state phase is clearly visible in Figure 5.3a. In this phase, the delta-V rate is the same in Scenarios 1, 2 and 3 and in Scenarios 4, 5 and 6. This is because the tug-debris system is held at the desired separation distance L_r. Therefore, skipping the first 1.5 hours of tugging, the cumulative $\|\Delta V\|$ lines plotted for Scenarios 1 to 3 ($L_r = 20$ m) are parallel, as well as for Scenarios 4 to 6 ($L_r = 35$ m). Also, a closer desired separation distance results in a greater delta-V rate. For example, the steady-state delta-V rate is 0.003 m/s per hour when $L_r = 35$ m. In contrast, the steady-state delta-V rate increases to 0.010 m/s per hour when $L_r = 20$ m.

The semi-major axis change rate $\dot{a}$, the delta-V rate $\|\dot{\Delta V}\|$ and the desired tug-debris separation distance L_r are constant in the steady-state phase. These three values can be considered a measure of mission speed, fuel consumption and mission risk, respectively. In the Geosynchronous Large Debris Reorbiter concept simulated here there is a relation between these three factors. Reference [8] proves analytically that $\dot{a} \propto F_c \propto 1/L_r^2$ when thrusters and electron beam are working continuously and simultaneously. Also, we can say that $\|\dot{\Delta V}\| \propto F_c$. Therefore, taking any pair i and j of the six scenarios considered in this section, the following relation is satisfied:

$$\frac{\dot{a}_i}{\dot{a}_j} \approx \frac{\|\dot{\Delta V}\|_i}{\|\dot{\Delta V}\|_j} \approx \frac{{L_r}_j^2}{{L_r}_i^2} \tag{5.2}$$

This equation is only valid in the steady-state phase.

Considering the results obtained from the simulations of Scenarios $i = 2$ and $j = 5$, the first two quotients in Equation 5.2 take a value of 3.5 rounded. The third quotient results in 3.1 approximately. Deviations are caused because the ET micro-thrusters and the mounted electron beam cannot work at the same time. In the simulations shown above, thrusters are on up to the first 10 seconds (κ_{max}) in every 30-second period (T). Then, the electron beam is on for the last 20 seconds.



(a) Complete data (24 hours).

(b) Zoom in first 3 hours.

Figure 5.3: Delta-V generated by tug thrusters for re-orbiting.

5.3.2. Tuning Controller Parameters

Next, Scenario 1 ($L_0 = 25$ m, $L_r = 20$ m, $L_{min} = 15$ m) is simulated seven times using different tuning parameters of the control software. These simulations take parameters from Tunings 1 to 7 in Table 5.2. Figure 5.4 shows the evolution of the separation distance between tug and debris for each simulation. The figure is divided in three plots, each plot showing three results.

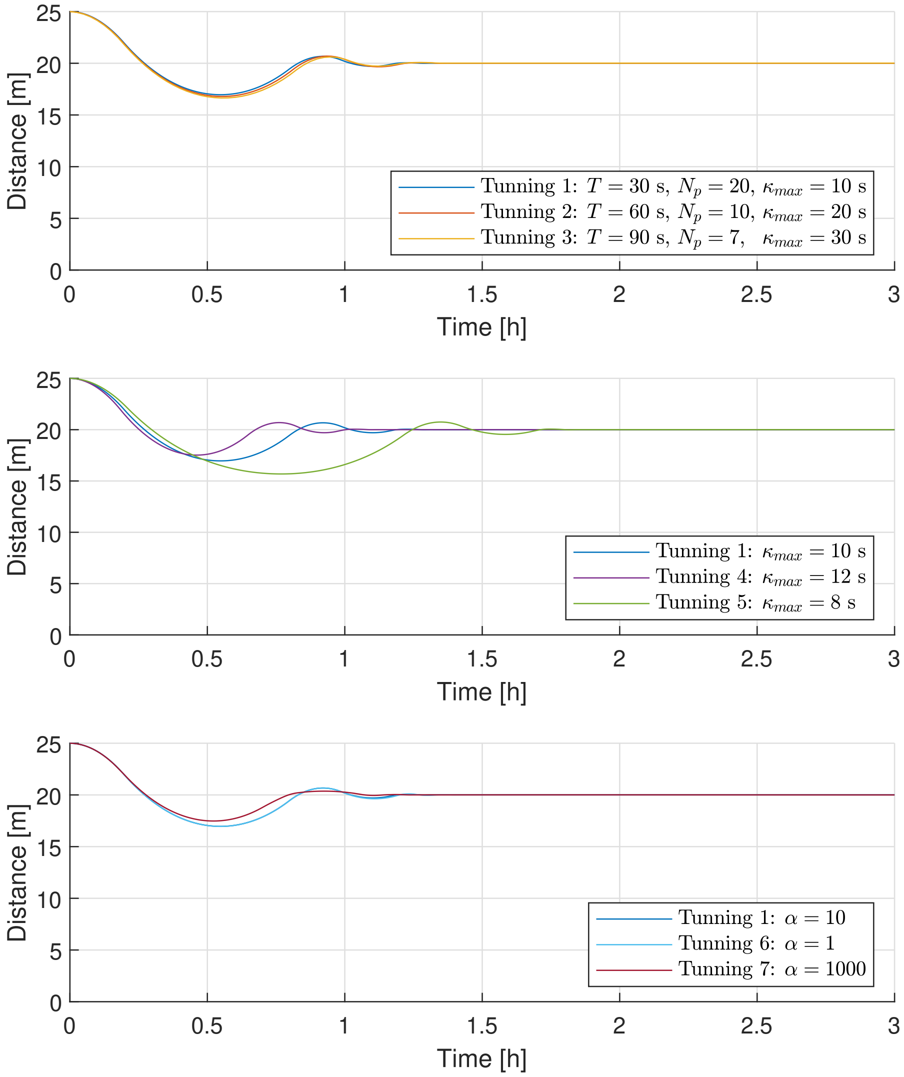


Figure 5.4: Evolution of separation distance tuning control parameters.

The top plot shows Tunings 1, 2 and 3. There is no major difference in the results of this three simulations. This is caused due to two factors. First, the planning horizon in units of time, which is calculated as $T \cdot N_p$, is the same for the three simulations: 10 minutes. And second, the maximum percentage of thrusting time, which is calculated as κ_{max}/T, is also conserved: 33.3%. If ratios were not conserved between Tunings 1 to 3, a lower value of T would result in a more precise solution, but a lower value of $T \cdot N_p$ would result in a less globally optimized solution.

The choice of κ_{max} depends of the amount of time allowed for thrusting according to specific mission requirements. However, the impulse $\kappa_{max} \cdot F_{thruster}$ must be enough to compensate the impulse of $(T - \kappa_{max}) \cdot F_c$ in the steady-state phase. The plot in the middle shows three simulations where only the κ_{max} parameter is changed. There are visible differences between the three lines of tug-debris separation distance plotted here. When this parameter is smaller, the system response makes larger oscillations and it takes more time to reach the steady-state phase. The opposite effect happens when κ_{max} gets greater.

The bottom plot shows Tunings 1, 6 and 7. These three simulations have different relative weight between input cost and state error (α). Even though the α parameters differ in order of magnitude, there are no major differences between the different responses of the system in the transition phase. This may be caused because for the given scenario ($L_0 = 25$ m, $L_r = 20$ m, $L_{min} = 15$ m) and the given parameters of both tug and debris spacecraft, an α of 1000 in value is not large enough to make the input cost rise next to the state error. Nevertheless, a small α is desired during the transition phase because it is critical that the tug approaches the debris object in the most secure way, as the separation distance is only a dozens of meters.

In the steady-state phase of the seven simulations, the results were the same as the red-dashed lines in Figure 5.2 and the lines for Scenario 1 in Figure 5.3.

5.3.3. Thrusters Configurations

The last analysis is carried out using the different configurations of micro-thrusters listed in Table 5.3. The four simulations consider Scenario 1, where $L_0 = 25$ m, $L_r = 20$ m, $L_{min} = 15$ m, and the controller parameters are set to Tuning 1: $T = 30$ s, $N_p = 20$, $\kappa_{max} = 10$ s and $\alpha = 10$. When using Configuration 1, the tug has 6 micro-thrusters of 15 mN each. The thrusters are oriented in the body-frame axes an each pair in opposite directions. Configuration 2 considers only the thrusters in the y axis, which is the axis defined by the orbital velocity vector of the debris spacecraft. Configuration 3 considers only one thruster: the thruster that generates a thrust in the $+y$ direction, so the tug and debris can separate. Finally, Configuration 4 lowers that maximum thrust in $+y$ direction to 1 milli-Newton.

Figure 5.5 plots the evolution of the separation distance between the tug and debris for the four simulations. The simulation with Configuration 1 has already been analyzed in the previous sections. When using Configuration 2, the evolution of the tug-debris distance is almost equal to the first simulation. There are two reasons for this. First, in this simulation, the tug starts very aligned with debris' orbital velocity. Therefore, there is almost no need to thrust in $\pm x$ and $\pm z$ directions (radial and out-of-plane, respectively). Second, the electron beam provides active charging control thanks to the generated Coulomb force. As this Coulomb force is always attractive, x and z coordinates of tug and debris

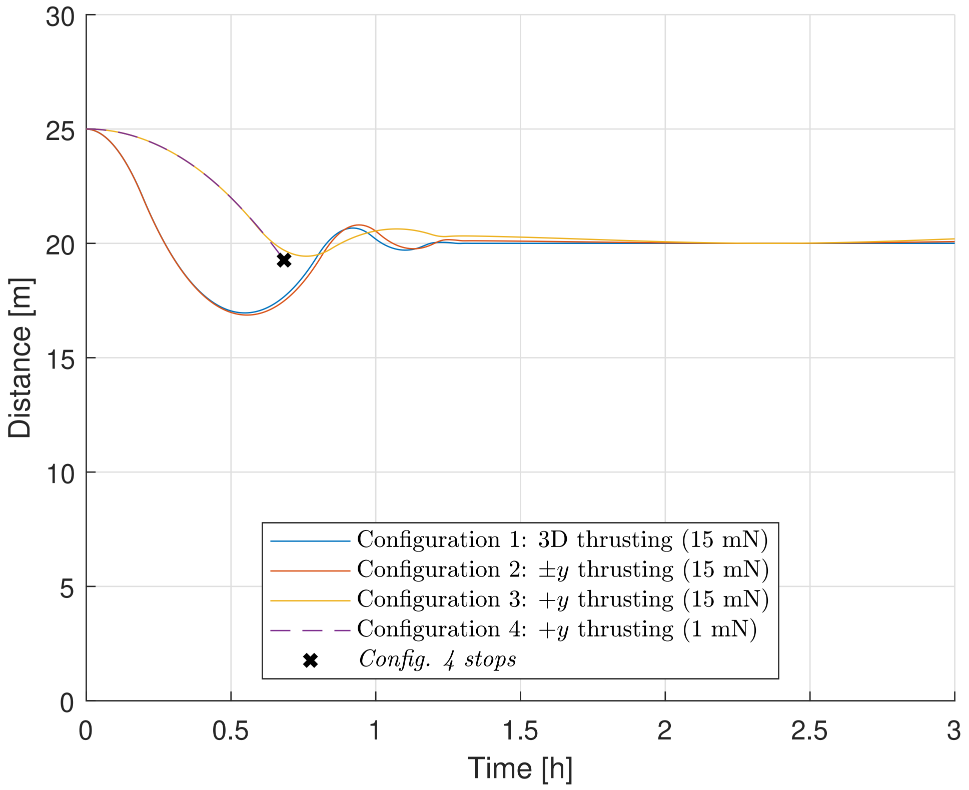


Figure 5.5: Evolution of separation distance with different micro-thrusters configurations.

spacecraft converge without thrusting in those directions.

The third simulation removes the micro-thruster that provides thrust in the $-y$ direction. Now, the tug has only one micro-thruster, able to provide thrust in the $+y$ direction in order to separate tug and debris. The evolution of the separation distance between tug and debris in this case differs considerably from the previous cases. Now, only the attractive Coulomb force brings both spacecraft together, so it takes more time to cross the $L = L_r$ line. Nevertheless, as the velocity is less than in the previous cases, the oscillation when using Configuration 3 is smaller in amplitude.

Finally, the fourth simulation takes the previous case but replaces the 15 mN micro-thruster by one of 1 mN. The dashed line for Configuration 4 plotted in Figure 5.5 follows the line for Configuration 3. However, in this simulation the control algorithm triggers a stop after 40 minutes. At this point, the algorithm finds out that the tug is not able to stay further than $d_{min} = 15$ m from the debris object at some moment in the following 10 minutes ($T \cdot N_p$). This happens because the impulse of 20 seconds of attractive Coulomb force cannot be compensated with 10 seconds of 1 milli-Newton thrust. A solution for this is briefly suggested in Section 6.3 as future work.

Figure 5.6 shows the delta-V generated by the thrusters of the Electrostatic Tractor (the tug space-

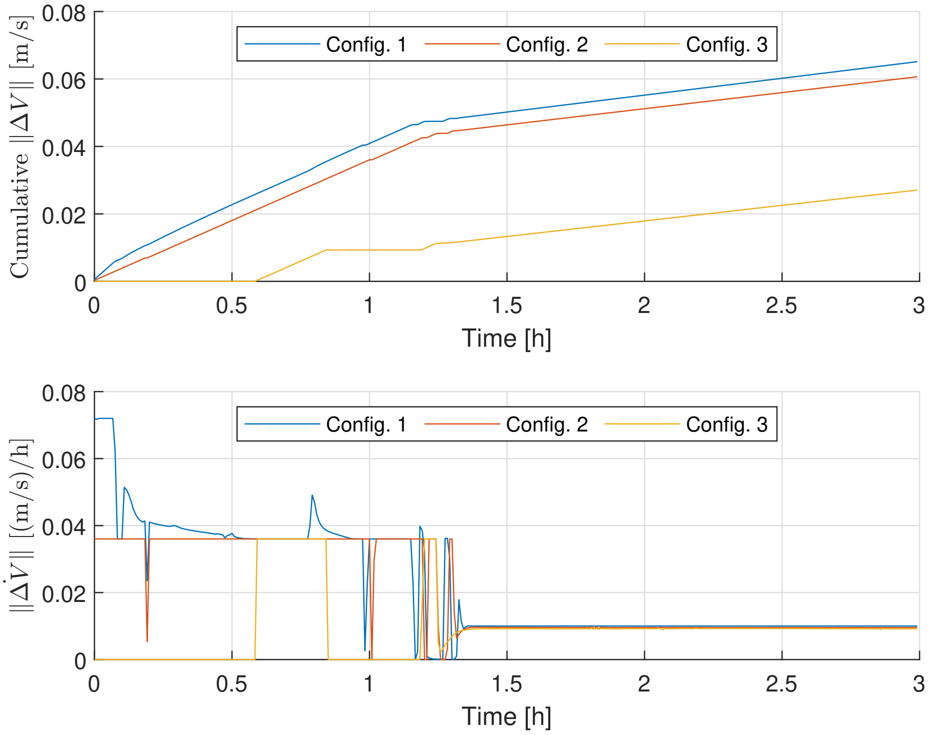


Figure 5.6: Delta-V generated by different tug thrusters configurations.

craft) when using Configurations 1, 2 and 3. The only remarkable data to point out here is that, in the transition phase of the tug-debris system, the delta-V generated by the thruster of Configuration 3 (orange line) is much less compared to the total delta-V of the previous two cases. This happens because, during the transition phase, the $+y$ and $-y$ thrusters are very used. However, Configuration 3 does not have the $-y$ thruster, so that delta-V is not generated and less fuel is consumed. Also, the duration of the transition phase using only one micro-thruster is very similar to the full 3D case. This puts the option of using only one micro-thruster in the Electrostatic Tractor as a good choice to be further studied.

6

Conclusions

This chapter summarizes the work conducted in this thesis, identifies its main findings and contributions, and highlights opportunities for future work.

6.1. Thesis Summary

There are tens of thousands of space debris objects orbiting Earth. Several active debris removal techniques have been proposed in the past, but the vast majority imply a physical contact with the debris object. Touching space debris becomes very dangerous when these are non-functioning big spacecraft such as the ones in the geostationary orbit. Also, this debris can be spinning at a high rate. The geosynchronous large debris reorbiter concept is a contactless active debris removal method proposed for defunct satellites in the geostationary orbit using the electrostatic tractor. However, research in the propulsion system of the electrostatic tractor was necessary in order to handle the challenging task of station keeping a few dozen meters away between the tug and the debris spacecraft must be done with thrusters that can only intermittently fire. In between the firing, the electron gun is used to control the tug and debris potential. In this thesis report, the author devises and simulates a reliable propulsion system for the geosynchronous large debris reorbiter concept by surveying different micro-propulsion thrusting options and developing a discrete thrust control algorithm using pulsed-width modulation thrust and active charging control.

The physics model of the Electrostatic Tractor for GEO space debris removal is based on the relative motion between two objects in orbit around the same central body. A basic PD feedback control for this relative motion could work if the thrusters could thrust continuously, simultaneous to the electron beam. A Coulomb force between two spherical bodies is sufficient for the controller to model the electrostatic force between tug and debris.

Cold gas and resistojets are better micro-thrusting options for the electrostatic tractor over chemical and electrical propulsion systems. These can operate safely in a high voltage environment. Additionally, having high thrust resolution and low thrust bias and noise is not necessary as these have very low effect for the control algorithm.

Approaching and maintaining debris and tug at a fixed separation distance, makes the Electrostatic Tractor problem equivalent to a spacecraft rendezvous problem. The latter consists of two nearby spacecraft where one approaches the other. The spacecraft rendezvous problem, and therefore the Electrostatic Tractor problem, is a trajectory planning problem. The relative orbital motion problem is expressed as a linear time-varying system. Then, a control algorithm that optimizes both present and future states of the trajectory planning problem is applicable to the Electrostatic Tractor. Moreover, this thesis shows how to make the control algorithm use pulsed-width modulation for the control inputs, so the thrusters fire intermittently.

When the discrete thrust control algorithm designed and the physics model are simulated, the results show the correctness and performance of the proposed propulsion system. Different scenarios, tuned controller parameters and multiple micro-thrusters configurations show different results. Some of the simulations have better performance in terms of reorbit time while others in terms of fuel consumption. Also, a lower separation distance between tug and spacecraft lowers the reorbit time but increases the chance of collision.

While this research proposes a suitable propulsion system for the geosynchronous large debris reorbiter concept, it also provides –at the end of this chapter– future work guidelines. Further research is necessary to precisely define and verify the most adequate propulsion system necessary. In some years, if the use of the electrostatic tractor to remove space debris becomes a reality, the author expects this thesis research project to have contributed to the actual mission.

6.2. Thesis Contributions

This thesis project the author devises and simulates a reliable propulsion system for the geosynchronous large debris reorbiter concept. To do so, a survey of different micro-propulsion thrusting options is provided in Chapter 3. Then, a discrete thrust control algorithm for this application of the Electrostatic Tractor is designed in Chapter 4. The designed control algorithm is based on a predictive control optimization problem with pulsed-width-modulated (PWM) inputs. The algorithm is particularized to the Electrostatic Tractor problem, which is a modification of the spacecraft rendezvous problem, a type of trajectory planning problem. The algorithm uses the physics model developed in Chapter 2 and is later simulated in Chapter 5. The propulsion system developed in this research allows the station keeping between the tug and the debris spacecraft with thrusters that can only intermittently fire. The electron gun is used to control the tug and debris potential in between the thruster firing. The research objective formulated in Section 1.3 has been achieved.

This research proposes cold gas systems and micro-resistojets as suitable candidates for the geosynchronous large debris reorbiter concept. Also, the discrete thrust control algorithm designed, based on the predictive control optimization problem with PWM control inputs, is suitable for the geosynchronous large debris reorbiter concept. Simulations in Chapter 5 show that the proposed propulsion system gives correct results and has a good performance according to the specifications defined in past research. Also, the thruster configuration that thrusts only in the direction to separate tug and debris spacecraft is demonstrated to use less fuel than the 3D thruster while maintaining the reorbit time –so performance improves.

6.3. Future Work

While different scenarios, controller parameters and thruster configurations are simulated for the propulsion system proposed in this research, more simulations are required. A Monte Carlo analysis is necessary to accurately test the system in terms of suitability and performance. Also, the propulsion system should be verified experimentally using the proposed micro-thruster technology and the designed control algorithm. A hardware-in-the-loop can be used for the experimental testing.

The proposed controller has areas of improvement. For example, the state constraint that avoids tug and debris to be at less than the minimum security separation distance introduces some practical problems. The current issue is that the controller may put the spacecraft at almost the minimum separation distance. While this works, with the current implementation there are cases where, on the next iteration of the controller, the actual separation distance is slightly less than the minimum safety distance (due to simulation propagation) and the optimization problem crashes as infeasible. One way to solve this issue is put the minimum distance constraint in the form of a soft constraint.

The Literature Study report proposed several hybrid control algorithms that can also be suitable for the Electrostatic Tractor removing space debris. While in this thesis project a combination of model predictive control and optimal control has been used, other control algorithms should be implemented and tested. For example, supervisory control and sliding mode control are recommended candidates in the literature study, which can be done in future work.

References

[1] ESA Space Debris Office. *ESA's Annual Space Environment Report*. European Space Operations Centre, May 2021, pp. 97–100.

[2] National Aeronautics and Space Administration. "Thirtieth Anniversary of the NASA Orbital Debris Program Office". In: *Orbital Debris Quaterly News* 13 (2009), pp. 1–2.

[3] Donald J. Kessler and Burton G. Cour-Palais. "Collision frequency of artificial satellites: The creation of a debris belt". In: *Journal of Geophysical Research: Space Physics* 83 (A6 June 1978), pp. 2637–2646.

[4] TD Bess - Work. "Mass distribution of orbiting man-made space debris". In: *ntrs.nasa.gov* (1975).

[5] CelesTrak. *SATCAT Boxscore*. 2021. URL: `http://www.celestrak.com/satcat/boxscore.php`.

[6] NASA Safety Standard. "Guidelines and assessment procedures for limiting orbital debris". In: *NASA NSS* 1740 (1995), p. 14.

[7] Rüdiger Jehn and Cristina Hernández. "International Practices to Protect the Geostationary Ring". In: *Space Debris* 1 (4 1999), pp. 221–233.

[8] Hanspeter Schaub and Daniel F. Moorer. "Geosynchronous Large Debris Reorbiter: Challenges and Prospects". In: *The Journal of the Astronautical Sciences 2013 59:1* 59 (1 Mar. 2014), pp. 161–176.

[9] Minghe Shan, Jian Guo, and Eberhard Gill. "Review and comparison of active space debris capturing and removal methods". In: *Progress in Aerospace Sciences* 80 (Jan. 2016), pp. 18–32.

[10] Steven G. Tragesser and Hakan San. "Orbital Maneuvering with Electrodynamic Tethers". In: *Journal of Guidance, Control, and Dynamics* 26 (5 2003), pp. 805–810.

[11] Shoji Kitamura, Yukio Hayakawa, and Satomi Kawamoto. "A reorbiter for large GEO debris objects using ion beam irradiation". In: *Acta Astronautica* 94 (2 Feb. 2014), pp. 725–735.

[12] Erik Hogan and Hanspeter Schaub. "Relative Motion Control for Two-Spacecraft Electrostatic Orbit Corrections". In: *AIAA Journal of Guidance, Control, and Dynamics* 36.1 (Jan. 2013), pp. 240–249.

[13] Guillem Rueda Oller. *Thrusting System for Electrostatic Space Debris Control. Literature Study*. Delft University of Technology, Jan. 2022.

[14] Joseph Hughes and Hanspeter Schaub. “Prospects of Using a Pulsed Electrostatic Tractor With Nominal Geosynchronous Conditions”. In: *IEEE Transactions on Plasma Science* 45 (8 Aug. 2017), pp. 1887–1897.

[15] Julian Hammerl and Hanspeter Schaub. “Effects of Electric Potential Uncertainty on Electrostatic Tractor Relative Motion Control Equilibria”. In: *Journal of Spacecraft and Rockets* (Sept. 2021), pp. 1–11.

[16] Lee E. Z. Jasper and Hanspeter Schaub. “Effective sphere modeling for electrostatic forces on a three-dimensional spacecraft shape”. In: 2011.

[17] Daan Stevenson and Hanspeter Schaub. “Multi-Sphere Method for modeling spacecraft electrostatic forces and torques”. In: *Advances in Space Research* 51 (1 Jan. 2013), pp. 10–20.

[18] Joseph A. Hughes and Hanspeter Schaub. “Heterogeneous Surface Multi-Sphere Models Using Method of Moments Foundations”. In: *Journal of Spacecraft and Rockets* 56 (4 July 2019), pp. 1259–1266.

[19] Jordan Maxwell et al. “Multi-Sphere Method for Flexible Conducting Space Objects: Modeling and Experiments”. In: *Journal of Spacecraft and Rockets* 57 (2 Mar. 2020), pp. 225–234.

[20] Kieran Wilson and Hanspeter Schaub. “Impact of Electrostatic Perturbations on Proximity Operations in High Earth Orbits”. In: *Journal of Spacecraft and Rockets* 58 (5 Sept. 2021), pp. 1293–1302.

[21] Angelo Cervone. *Micro-Propulsion. Course reader*. Delft University of Technology, Sept. 2018.

[22] Igor Levchenko et al. “Space micropropulsion systems for Cubesats and small satellites: From proximate targets to furthermost frontiers”. In: *Applied Physics Reviews* 5 (1 Feb. 2018), p. 011104.

[23] Akshay Reddy Tummala and Atri Dutta. “An Overview of Cube-Satellite Propulsion Technologies and Trends”. In: *Aerospace 2017, Vol. 4, Page 58* 4 (4 Dec. 2017), p. 58.

[24] David Krejci and Paulo Lozano. “Space Propulsion Technology for Small Spacecraft”. In: *Proceedings of the IEEE* 106 (3 Mar. 2018), pp. 362–378.

[25] Marsil A.C. Silva et al. “A review of MEMS micropropulsion technologies for CubeSats and PocketQubes”. In: *Acta Astronautica* 143 (Feb. 2018), pp. 234–243.

[26] Kristina Lemmer. “Propulsion for CubeSats”. In: *Acta Astronautica* 134 (May 2017), pp. 231–243.

[27] Khary I. Parker. “State-of-the-Art for Small Satellite Propulsion Systems”. In: (2016).

[28] Mirko Leomanni et al. “Propulsion options for very low Earth orbit microsatellites”. In: *Acta Astronautica* 133 (Apr. 2017), pp. 444–454.

[29] Schaub H. *Low Cost Orbital Debris Removal System: Geosynchronous Large Debris Reorbiter (GLiDeR)*. SBIR Phase I Final Report, Wacari Group, Jan. 2011.

[30] Hanspeter Schaub. *Module: etSphericalControl – Basilisk 2.1.3 documentation*. URL: https://hanspeterschaub.info/basilisk/Documentation/fswAlgorithms/formationFlying/etSphericalControl/etSphericalControl.html#etsphericalcontrol.

[31] Julian Hammerl and Hanspeter Schaub. *Scenario Debris Reorbit ET – Basilisk 2.1.3 documentation*. URL: https://hanspeterschaub.info/basilisk/examples/scenarioDebrisReorbitET.html.

[32] SatCatalog. *IFM Micro 100 Thruster - Electric Propulsion System*. URL: https://www.satcatalog.com/component/ifm-micro-100-thruster/.

[33] R. Vazquez, F. Gavilan, and E. F. Camacho. “Pulse-width predictive control for LTV systems with application to spacecraft rendezvous”. In: *Control Engineering Practice* 60 (Mar. 2017), pp. 199–210.

[34] WJ Rugh. *Linear system theory*. Prentice-Hall, Inc., 1996.

[35] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Sigma series in applied mathematics. Heldermann, 1988.

[36] Frédéric Delbos and Jean Charles Gilbert. “Global Linear Convergence of an Augmented Lagrangian Algorithm to Solve Convex Quadratic Optimization Problems”. In: *Journal of Convex Analysis* 12 (Jan. 2005), pp. 45–69.

[37] LLC Gurobi Optimization. *Method - Gurobi Optimization*. URL: https://www.gurobi.com/documentation/10.0/refman/method.html.

[38] Francisco Gavilan, Rafael Vazquez, and Eduardo F. Camacho. “Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation”. In: *Control Engineering Practice* 20 (2 Feb. 2012), pp. 111–122.

A

Simulation Code

This appendix contains the Matlab code implemented and executed to obtain the simulations presented in this thesis project. Additionally, the code can be downloaded from the following repository: `https://github.com/GuillemRueda/GLiDeRcontrolSim`

A.1. Matlab File: execute.m

```
function [distance, cum_delta_v, delta_vv, ...
    Delta_height, Delta_SMA, i] = execute(T, N_p, kappa_max, ...
        alpha, d_min, L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, ...
        t_step, max_it, beta, mu, n)

    a = nthroot(mu/(n^2), 3);

    x_debris = [a; 0; 0; 0; a*n; 0]; % circular orbit
    x_tug = x_debris + [0; L0; 0; 0; 0; 0];

    kc = 8.988e9;

    model = getModel(m1, m2, L, alpha, d_min, F_max, T, N_p, n, ...
        kappa_max, R1, R2, V1, V2);

    xx_tug = zeros(6, N_t + 1);
    xx_debris = zeros(6, N_t + 1);
    delta_vv = zeros(6, N_t);

    xx_tug(:, 1) = x_tug;
    xx_debris(:, 1) = x_debris;

    for i = 1:N_t
        try
            [F_now, C] = getThrust(model, x_tug, x_debris, kc, ...
                max_it, beta);
        catch
            break;
        end

        [new_x_tug, new_x_debris, delta_v] = simulate(F_now, C, ...
            x_tug, x_debris, mu, m1, m2, F_max, kappa_max, T, t_step, ...
            kc, R1, R2, V1, V2);

        xx_tug(:, i + 1) = new_x_tug;
        xx_debris(:, i + 1) = new_x_debris;
        delta_vv(:, i) = delta_v;

        x_tug = new_x_tug;
        x_debris = new_x_debris;
    end

    distance = sqrt(sum((xx_tug(1:3, :) - xx_debris(1:3, :)).^2));
    cum_delta_v = cumsum(sum(delta_vv));
    Delta_height = sqrt(sum(xx_debris(1:3, :).^2)) - a;
    Delta_SMA = 1./(2./sqrt(sum(xx_debris(1:3, :).^2)) - ...
        (sqrt(sum(xx_debris(4:6, :).^2)).^2)/mu) - a;
end
```

A.2. Matlab File: simulate.m

```
function [x_tug, x_debris, delta_v] = simulate(F_now, C, ...
        x_tug, x_debris, mu, m1, m2, F_max, kappa_max, T, t_step, ...
        kc, R1, R2, V1, V2)
    kappa_max = round(kappa_max, 3);
    kappa = round([max(F_now, 0); max(-F_now, 0)]./F_max, 3);
    kappa = min([kappa.'; repmat(kappa_max, 1, 6)]).';

    v = unique([0; kappa; kappa_max]);
    l = v(2:end) - v(1:(end-1));
    v = v(1:(end-1));
    n = length(v);
    delta_v = zeros(6, 1);

    U_max = F_max/m1; % all(U_max >= 0) is true

    r_tug = x_tug(1:3);
    v_tug = x_tug(4:6);
    r_debris = x_debris(1:3);
    v_debris = x_debris(4:6);

    for i = 1:n
        tt = l(i);
        u = diag(kappa > v(i))*U_max;
        delta_v = delta_v + u*tt;
        nn = floor(tt/t_step);
        t_vec = [repmat(t_step, nn, 1); tt - nn*t_step];

        for j = 1:(nn + 1)
            t = t_vec(j);
            a_tug = -(mu/(norm(r_tug)^3))*r_tug + [C, -C]*u;
            r_tug = r_tug + v_tug*t + a_tug*(t^2)/2;
            v_tug = v_tug + a_tug*t;

            a_debris = -(mu/(norm(r_debris)^3))*r_debris;
            r_debris = r_debris + v_debris*t + a_debris*(t^2)/2;
            v_debris = v_debris + a_debris*t;
        end
    end

    tt = T - kappa_max;
    if tt > 0
        nn = floor(tt/t_step);
        t_vec = [repmat(t_step, nn, 1); tt - nn*t_step];

        for j = 1:(nn + 1)
            t = t_vec(j);
            L = norm(r_tug - r_debris);
            S = kc*[1/R1, 1/L;
                    1/L,  1/R2];
            Q = S\[V1; V2];

            F_c = kc*Q(1)*Q(2)/(L^2);
            F_c = (F_c/L)*(r_tug - r_debris);

            a_tug = -(mu/(norm(r_tug)^3))*r_tug + F_c/m1;
            r_tug = r_tug + v_tug*t + a_tug*(t^2)/2;
            v_tug = v_tug + a_tug*t;

            a_debris = -(mu/(norm(r_debris)^3))*r_debris - F_c/m2;
            r_debris = r_debris + v_debris*t + a_debris*(t^2)/2;
            v_debris = v_debris + a_debris*t;
        end
    end

    x_tug = [r_tug; v_tug];
    x_debris = [r_debris; v_debris];
end
```

A.3. Matlab File: getModel.m

```
function model = getModel(m1, m2, L, alpha, d_min, F_max, T, N_p, n, ...
        kappa_max, R1, R2, V1, V2)
    U_max = kappa_max*F_max/m1; % all(U_max >= 0) is true

    Phi = getStateTransitionMatrices(T, N_p, n);

    F = sparse(blkdiag(Phi{:}))*repmat(sparse(eye(6)), N_p, 1);

    G = spalloc(6*N_p, 6*N_p, 36*(N_p+1)*N_p/2);
    for i = 0:(N_p - 1)
        G((1 + 6*i):end, (1 + 6*i):(6 + 6*i)) = ...
            F((1 + 6*i):end, :); %#ok<SPRIX>
    end

    B = repmat({sparse([zeros(3); eye(3)])}, 1, 1, N_p);
    G = G*blkdiag(B{:});

    P = sparse([eye(3*N_p), -eye(3*N_p), eye(3*N_p)]);
    Q = sparse(eye(6*N_p));

    model.GP = G*P;

    C_s = repmat({sparse([0, -1, 0, 0, 0, 0])}, 1, 1, N_p);
    C_s = blkdiag(C_s{:});
    model.A = C_s*model.GP;
    model.d_s = repmat(-(d_min - L), N_p, 1);
    model.M = C_s*F;

    model.Q = model.GP.'*Q*model.GP;

    model.H = 2*(F.')*Q*model.GP;
    model.Alpha = repmat(alpha, 1, 9*N_p);

    U_max_bar = repmat(U_max(1:3), N_p, 1); % all(U_max_bar >= 0) is true
    U_min_bar = repmat(-U_max(4:6), N_p, 1); % all(U_min_bar <= 0) is true

```

```matlab
    model.lb = zeros(9*N_p, 1);
    model.ub = [U_max_bar; -U_min_bar; zeros(3*N_p, 1)];

    model.L = L;
    model.mass_tug = m1;
    model.mass_debris = m2;
    model.kappa_max = kappa_max;
    model.T = T;
    model.N_p = N_p;
    model.F = F;
    model.R1 = R1;
    model.R2 = R2;
    model.V1 = V1;
    model.V2 = V2;
end
```

A.4. Matlab File: getStateTransitionMatrices.m

```matlab
function Phi = getStateTransitionMatrices(T, N_p, n)
    t = T*(1:N_p);
    delta_anom = n*t;
    s = sin(delta_anom);
    c = cos(delta_anom);

    Phi = zeros(6, 6, N_p);
    Phi(1,1,:) = 4 - 3*c;
    Phi(2,1,:) = 6*(s - delta_anom);
    Phi(4,1,:) = 3*n*s;
    Phi(5,1,:) = 6*n*(c - 1);
    Phi(2,2,:) = 1;
    Phi(3,3,:) = c;
    Phi(6,3,:) = -n*s;
    Phi(1,4,:) = s/n;
    Phi(2,4,:) = 2*(c - 1)/n;
    Phi(4,4,:) = c;
    Phi(5,4,:) = -2*s;
    Phi(1,5,:) = 2*(1 - c)/n;
    Phi(2,5,:) = 4*s/n - 3*t;
    Phi(4,5,:) = 2*s;
    Phi(5,5,:) = 4*c - 3;
    Phi(3,6,:) = s/n;
    Phi(6,6,:) = c;

    Phi = mat2cell(Phi, 6, 6, ones(1, N_p));
end
```

A.5. Matlab File: getThrust.m

```matlab
function [F_now, C] = getThrust(model, x_tug, x_debris, kc, max_it, beta)
    r = x_tug(1:3);
    v = x_tug(4:6);
    omega = cross(r, v);

    y_coord = v/norm(v);
    z_coord = omega/norm(omega);
    x_coord = cross(y_coord, z_coord);
    C = [x_coord, y_coord, z_coord];

    x_rel = [C\(x_tug(1:3) - x_debris(1:3));
        C\(x_tug(4:6) - x_debris(4:6))];

    F_now = getForce(x_rel, model, kc, max_it, beta);
end
```

A.6. Matlab File: getForce.m

```matlab
function F_now = getForce(x_0, model, kc, max_it, beta)
    x_now = x_0 - [0; model.L; 0; 0; 0; 0];

    model.rhs = model.d_s - model.M*x_now;
    model.obj = (x_now.')*model.H + model.Alpha;

    V = repmat([model.V1; model.V2], model.N_p, 1);
    S0 = diag(sparse(repmat([1/model.R1; 1/model.R2], model.N_p, 1)));

    curr_results = zeros(9*model.N_p, 1);

    for i = 1:max_it
        model.results = curr_results;

        X = model.F*x_now + model.GP*model.results;
        X = [x_now; X(1:(end - 6))];
        X = reshape(X, 6, []);
        r = X(1:3, :);
        r(2, :) = r(2, :) + model.L;
        d = sqrt(sum(r.^2));

        dd = reshape([1./d; sparse(1, model.N_p)], 1, []);
        dd = dd(1:(end - 1));

        S = kc*(S0 + diag(dd, 1) + diag(dd, -1));

        Q = S\V;
        qq = Q(1:2:(end - 1)).*Q(2:2:end);

        F_C = kc*reshape(r.*repmat(qq.'./(d.^3), 3, 1), [], 1);
        u_C = (model.T - model.kappa_max) * F_C * ...
            (1/model.mass_tug + 1/model.mass_debris);

        model.lb((6*model.N_p + 1):end) = u_C;
        model.ub((6*model.N_p + 1):end) = u_C;

        curr_results = gurobi(model, struct('OutputFlag', 0)).x;
        if sum((curr_results - model.results).^2) < beta
            break;
        end
    end

    model.results = curr_results;

    u_now = model.results(1:3) - model.results(3*model.N_p + (1:3));
    F_now = model.mass_tug*u_now;
end
```

A.7. Matlab File: script441.m

```
% Copyright © Guillem Rueda Oller, 2022
% Author: Guillem Rueda Oller (student number 5006538)
% September 2022. Multiple Scenarios
close all;
clear all; %#ok<CLALL>
clc;

Tday = 86164.0905;
mu = 3.986044418e14;
n = 2*pi/Tday;

R1 = 3; % [m]
R2 = 3; % [m]
V1 = 20e3; % [V]
V2 = -20e3; % [V]
m1 = 500; % [kg]
m2 = 1000; % [kg]

t_step = 0.001; % [s]
max_it = 20;
beta = 1e-20;

T = 30; % [s]
N_p = 20;
kappa_max = 10; % [s]
alpha = 10;

F_max_single = 15e-3; % [N]
F_max = [F_max_single; F_max_single; F_max_single; ...
         F_max_single; F_max_single; F_max_single];

n_t = 24; % [h]
N_t = n_t*3600/T;

%% Scenario 1
L0 = 25; % [m]
```

```
L = 20; % [m]
d_min = 15; % [m]
tic;
[d1, v1, dv1, h1, hh1] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 2
L0 = 20; % [m]
L = 20; % [m]
d_min = 15; % [m]
tic;
[d2, v2, dv2, h2, hh2] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 3
L0 = 15; % [m]
L = 20; % [m]
d_min = 15; % [m]
tic;
[d3, v3, dv3, h3, hh3] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 4
L0 = 40; % [m]
L = 35; % [m]
d_min = 15; % [m]
tic;
[d4, v4, dv4, h4, hh4] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 5
L0 = 35; % [m]
L = 35; % [m]
```

```
d_min = 15; % [m]
tic;
[d5, v5, dv5, h5, hh5] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 6
L0 = 30; % [m]
L = 35; % [m]
d_min = 15; % [m]
tic;
[d6, v6, dv6, h6, hh6] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;

%% Plots

t = 0:(T/3600):n_t;

%% Figure 4.1
figure;
hold on;
plot(t, d1);
plot(t, d2, '--');
plot(t, d3);
plot(t, d4);
plot(t, d5, '--');
plot(t, d6);
grid;
ylim0 = ylim();
ylim([0 ylim0(2)]);
xlim([0 3]);
xlabel('Time [h]')
ylabel('Distance [m]');
lgd = legend('Scenario 1', 'Scenario 2', 'Scenario 3', ...
    'Scenario 4', 'Scenario 5', 'Scenario 6', 'Location', 'south', ...
    'Orientation', 'horizontal');
lgd.NumColumns = 3;

%% Figure 4.2
figure;
subplot(2,2,1);
hold on;
plot(t, h2, '--', 'color', [0.850, 0.325, 0.098]);
plot(t, h5, '--', 'color', [0.466, 0.674, 0.188]);
grid;
xlim([0 n_t]);
ylim1 = ylim();
ylim([0 ylim1(2)]);
xticks(0:6:n_t);
xlabel('Time [h]')
ylabel('$\Delta{r}$ [m]', 'Interpreter', 'latex');

subplot(2,2,3);
hold on;
plot(t(1:(end-1)), (h2(2:end)-h2(1:(end-1)))/(T/3600), '--', ...
    'color', [0.850, 0.325, 0.098]);
plot(t(1:(end-1)), (h5(2:end)-h5(1:(end-1)))/(T/3600), '--', ...
    'color', [0.466, 0.674, 0.188]);
grid;
xlim([0 n_t]);
ylim2 = ylim();
ylim([0 ylim2(2)]);
xticks(0:6:n_t);
xlabel('Time [h]')
ylabel('$\dot{r}$ [m/h]', 'Interpreter', 'latex');

subplot(2,2,2);
hold on;
plot(t, hh2, '--', 'color', [0.850, 0.325, 0.098]);
plot(t, hh5, '--', 'color', [0.466, 0.674, 0.188]);
grid;
xlim([0 n_t]);
```

```matlab
ylim([0 ylim1(2)]);
xticks(0:6:n_t);
ylim2 = ylim();
ylim([0 ylim2(2)]);
xlabel('Time [h]')
ylabel('$\Delta{a}$ [m]', 'Interpreter', 'latex');

subplot(2,2,4);
hold on;
plot(t(1:(end-1)), (hh2(2:end)-hh2(1:(end-1)))/(T/3600), '--', ...
    'color', [0.850, 0.325, 0.098]);
plot(t(1:(end-1)), (hh5(2:end)-hh5(1:(end-1)))/(T/3600), '--', ...
    'color', [0.466, 0.674, 0.188]);
grid;
xlim([0 n_t]);
ylim([0 ylim2(2)]);
xticks(0:6:n_t);
ylim2 = ylim();
ylim([0 ylim2(2)]);
xlabel('Time [h]')
ylabel('$\dot{a}$ [m/h]', 'Interpreter', 'latex');

%% Figure 4.3
figure;
subplot(2, 1, 1);
hold on;
plot(t(1:(end-1)), v1);
plot(t(1:(end-1)), v2, '--');
plot(t(1:(end-1)), v3);
plot(t(1:(end-1)), v4);
plot(t(1:(end-1)), v5, '--');
plot(t(1:(end-1)), v6);
grid;
xlim([0 n_t]);
xticks(0:2:n_t);
xlabel('Time [h]')
ylabel('Cumulative $\left\|\Delta{V}\right\|$ [m/s]', ...
    'Interpreter', 'latex');
lgd = legend('Scenario 1', 'Scenario 2', 'Scenario 3', ...
    'Scenario 4', 'Scenario 5', 'Scenario 6', 'Location', 'north', ...
    'Orientation', 'horizontal');
lgd.NumColumns = 3;

subplot(2, 1, 2);
hold on;
plot(t(1:(end-1)), sum(dv1)/(T/3600));
plot(t(1:(end-1)), sum(dv2)/(T/3600), '--');
plot(t(1:(end-1)), sum(dv3)/(T/3600));
plot(t(1:(end-1)), sum(dv4)/(T/3600));
plot(t(1:(end-1)), sum(dv5)/(T/3600), '--');
plot(t(1:(end-1)), sum(dv6)/(T/3600));
grid;
xlim([0 n_t]);
xticks(0:2:n_t);
xlabel('Time [h]')
ylabel('$\dot{\left\|\Delta{V}\right\|}$ [(m/s)/h]', ...
    'Interpreter', 'latex');
lgd = legend('Scenario 1', 'Scenario 2', 'Scenario 3', ...
    'Scenario 4', 'Scenario 5', 'Scenario 6', 'Location', 'north', ...
    'Orientation', 'horizontal');
lgd.NumColumns = 3;
```

A.8. Matlab File: script442.m

```
% Copyright © Guillem Rueda Oller, 2022
% Author: Guillem Rueda Oller (student number 5006538)
% September 2022. Tuning Controller Parameters
close all;
clear all; %#ok<CLALL>
clc;

Tday = 86164.0905;
mu = 3.986044418e14;
n = 2*pi/Tday;

R1 = 3; % [m]
R2 = 3; % [m]
V1 = 20e3; % [V]
V2 = -20e3; % [V]
m1 = 500; % [kg]
m2 = 1000; % [kg]

t_step = 0.1; % [s]
max_it = 20;
beta = 1e-20;

L0 = 25; % [m]
L = 20; % [m]
d_min = 15; % [m]

F_max_single = 15e-3; % [N]
F_max = [F_max_single; F_max_single; F_max_single; ...
         F_max_single; F_max_single; F_max_single];

n_t = 3; % [h]

%% Scenario 1, Tunning 1
T = 30; % [s]
N_p = 20;
kappa_max = 10; % [s]
```

```
alpha = 10;
N_t = n_t*3600/T;
tic;
[d1, v1, dv1, h1, hh1] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 2
T = 60; % [s]
N_p = 10;
kappa_max = 20; % [s]
alpha = 10;
N_t = n_t*3600/T;
tic;
[d2, v2, dv2, h2, hh2] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 3
T = 90; % [s]
N_p = 4;
kappa_max = 30; % [s]
alpha = 10;
N_t = n_t*3600/T;
tic;
[d3, v3, dv3, h3, hh3] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 4
T = 30; % [s]
N_p = 20;
kappa_max = 12; % [s]
alpha = 10;
N_t = n_t*3600/T;
tic;
[d4, v4, dv4, h4, hh4] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 5
T = 30; % [s]
N_p = 20;
kappa_max = 8; % [s]
alpha = 10;
N_t = n_t*3600/T;
tic;
[d5, v5, dv5, h5, hh5] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 6
T = 30; % [s]
N_p = 20;
kappa_max = 10; % [s]
alpha = 1;
N_t = n_t*3600/T;
tic;
[d6, v6, dv6, h6, hh6] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Tunning 7
T = 30; % [s]
N_p = 20;
kappa_max = 10; % [s]
alpha = 1000;
N_t = n_t*3600/T;
tic;
[d7, v7, dv7, h7, hh7] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;

```

```
%% Plot. Figure 4.4

t = 0:(T/3600):n_t;
t2 = 0:(2*T/3600):n_t;
t3 = 0:(3*T/3600):n_t;
figure;

subplot(3, 1, 1);
hold on;
plot(t, d1);
plot(t2, d2);
plot(t3, d3);
grid;
ylim0 = ylim();
ylim([0 ylim0(2)]);
xlim([0 3]);
xlabel('Time [h]')
ylabel('Distance [m]');
legend('Tunning 1: $T=30$ s, $N_p=20$, $\kappa_{max}=10$ s', ...
    'Tunning 2: $T=60$ s, $N_p=10$, $\kappa_{max}=20$ s', ...
    'Tunning 3: $T=90$ s, $N_p=7$, $\phantom{0}\kappa_{max}=30$ s', ...
    'Location', 'southeast', 'Interpreter', 'latex');

subplot(3, 1, 2);
hold on;
plot(t, d1);
plot(t, d4, 'color', [0.4940 0.1840 0.5560]);
plot(t, d5, 'color', [0.4660 0.6740 0.1880]);
grid;
ylim0 = ylim();
ylim([0 ylim0(2)]);
xlim([0 3]);
xlabel('Time [h]')
ylabel('Distance [m]');
legend('Tunning 1: $\kappa_{max}=10$ s', ...
    'Tunning 4: $\kappa_{max}=12$ s', ...
    'Tunning 5: $\kappa_{max}=8$ s', ...
    'Location', 'southeast', 'Interpreter', 'latex');

subplot(3, 1, 3);
hold on;
plot(t, d1);
plot(t, d6, 'color', [0.3010 0.7450 0.9330]);
plot(t, d7, 'color', [0.6350 0.0780 0.1840]);
grid;
ylim0 = ylim();
ylim([0 ylim0(2)]);
xlim([0 3]);
xlabel('Time [h]')
ylabel('Distance [m]');
legend('Tunning 1: $\alpha=10$', ...
    'Tunning 6: $\alpha=1$', ...
    'Tunning 7: $\alpha=1000$', ...
    'Location', 'southeast', 'Interpreter', 'latex');
```

A.9. Matlab File: script443.m

```
% Copyright © Guillem Rueda Oller, 2022
% Author: Guillem Rueda Oller (student number 5006538)
% September 2022. Thrusters Configurations
close all;
clear all; %#ok<CLALL>
clc;

Tday = 86164.0905;
mu = 3.986044418e14;
n = 2*pi/Tday;

R1 = 3; % [m]
R2 = 3; % [m]
V1 = 20e3; % [V]
V2 = -20e3; % [V]
m1 = 500; % [kg]
m2 = 1000; % [kg]

t_step = 0.1; % [s]
max_it = 20;
beta = 1e-20;

L0 = 25; % [m]
L = 20; % [m]
d_min = 15; % [m]

T = 30; % [s]
N_p = 20;
kappa_max = 10; % [s]
alpha = 10;

F_max_single = 15e-3; % [N]

n_t = 3; % [h]
N_t = n_t*3600/T;

```

```
%% Scenario 1, Configuration 1
F_max = [F_max_single; F_max_single; F_max_single; ...
         F_max_single; F_max_single; F_max_single];
tic;
[d1, v1, dv1, h1, hh1] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Configuration 2
F_max = [0; F_max_single; 0; ...
         0; F_max_single; 0];
tic;
[d2, v2, dv2, h2, hh2] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Configuration 3
F_max = [0; F_max_single; 0; ...
         0;            0; 0];
tic;
[d3, v3, dv3, h3, hh3] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;
%% Scenario 1, Configuration 4
F_max = [0; 1e-3; 0; ...
         0;    0; 0]; % [N]
tic;
[d4, v4, dv4, h4, hh4, i] = execute(T, N_p, kappa_max, alpha, d_min, ...
    L, L0, F_max, N_t, R1, R2, V1, V2, m1, m2, t_step, max_it, ...
    beta, mu, n);
toc;

%% Plots

t = 0:(T/3600):n_t;

```

```
%% Figure 4.5

figure;
hold on;
plot(t, d1);
plot(t, d2);
plot(t, d3);
plot(t(1:(i-1)), d4(1:(i-1)), '--');
plot(t(i), d4(i), 'x', 'color', 'black','LineWidth', 2);
grid;
ylim0 = ylim();
ylim([0 30]);
xlim([0 3]);
xlabel('Time [h]')
ylabel('Distance [m]');
legend('Configuration 1: 3D thrusting (15 mN)', ...
    'Configuration 2: $\pm y$ thrusting (15 mN)', ...
    'Configuration 3: $+y$ thrusting (15 mN)', ...
    'Configuration 4: $+y$ thrusting (1 mN)', ...
    '\it{Config. 4 stops}', 'Location', 'south', ...
    'Interpreter', 'latex');

%% Figure 4.6
figure;
subplot(2, 1, 1);
hold on;
plot(t(1:(end-1)), v1);
plot(t(1:(end-1)), v2);
plot(t(1:(end-1)), v3);
grid;
xlim([0 n_t]);
xlabel('Time [h]')
ylabel('Cumulative $\left\|\Delta{V}\right\|$ [m/s]', ...
    'Interpreter', 'latex');
lgd = legend('Config. 1', 'Config. 2', 'Config. 3', ...
    'Location', 'north', 'Orientation', 'horizontal');
lgd.NumColumns = 3;
```

```

subplot(2, 1, 2);
hold on;
plot(t(1:(end-1)), sum(dv1)/(T/3600));
plot(t(1:(end-1)), sum(dv2)/(T/3600));
plot(t(1:(end-1)), sum(dv3)/(T/3600));
grid;
xlim([0 n_t]);
xlabel('Time [h]')
ylabel('$\dot{\left\|\Delta{V}\right\|}$ [(m/s)/h]', ...
    'Interpreter', 'latex');
lgd = legend('Config. 1', 'Config. 2', 'Config. 3', ...
    'Location', 'north', 'Orientation', 'horizontal');
lgd.NumColumns = 3;
```