POLITECNICO
MILANO 1863

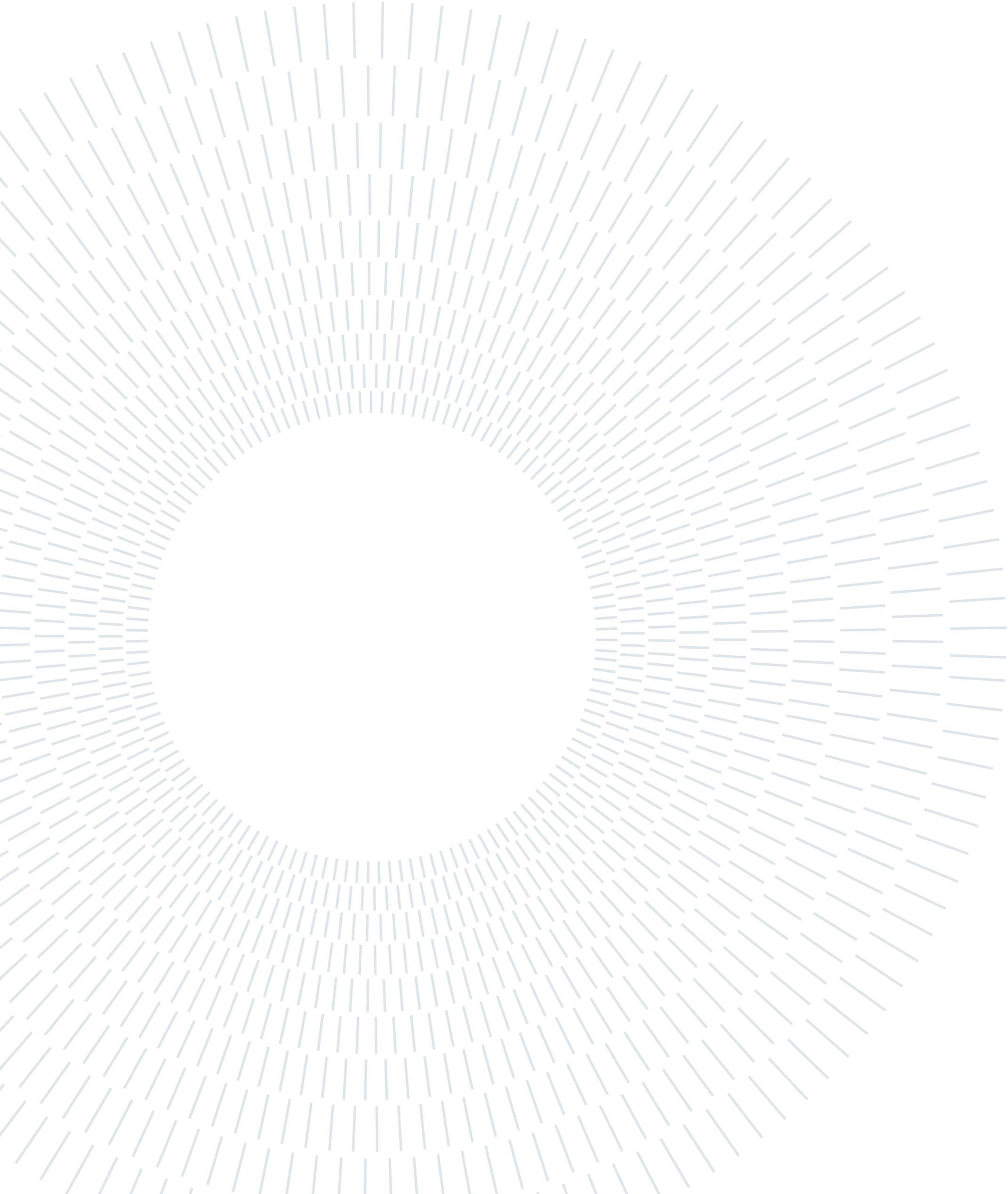
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Model Discovery Predictive Control for Satellite Relative Range Bounding

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

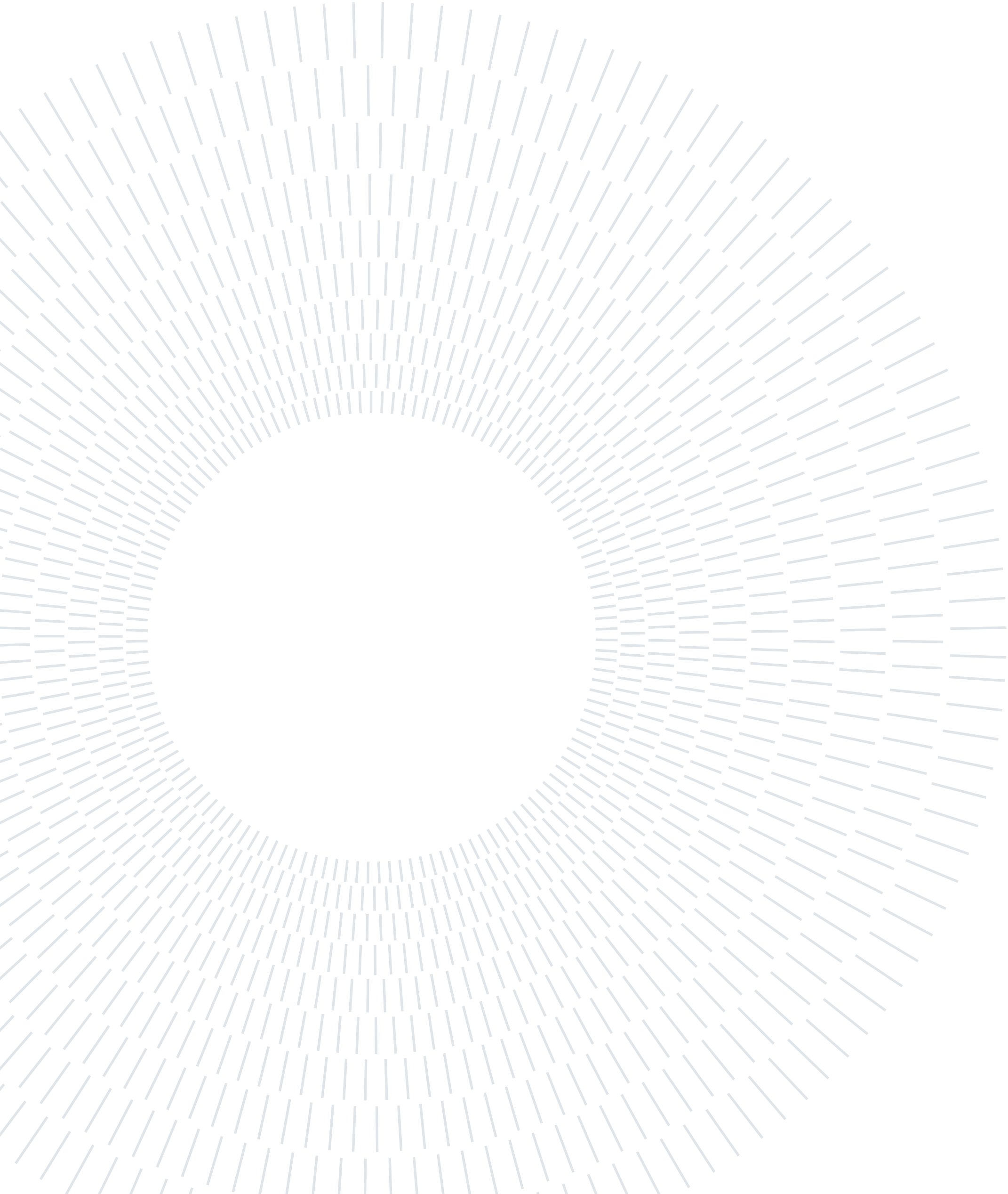
Author: **Hariharan Venkatesh Vitaladevuni**

Student ID: 940223
Advisor: Prof. Mauro Massari
Academic Year: 2021-22



*All our knowledge - past,
present, and future - is
nothing compared to what
we will never know.*

KONSTANTIN E. TSIOLKOVSKY



Abstract

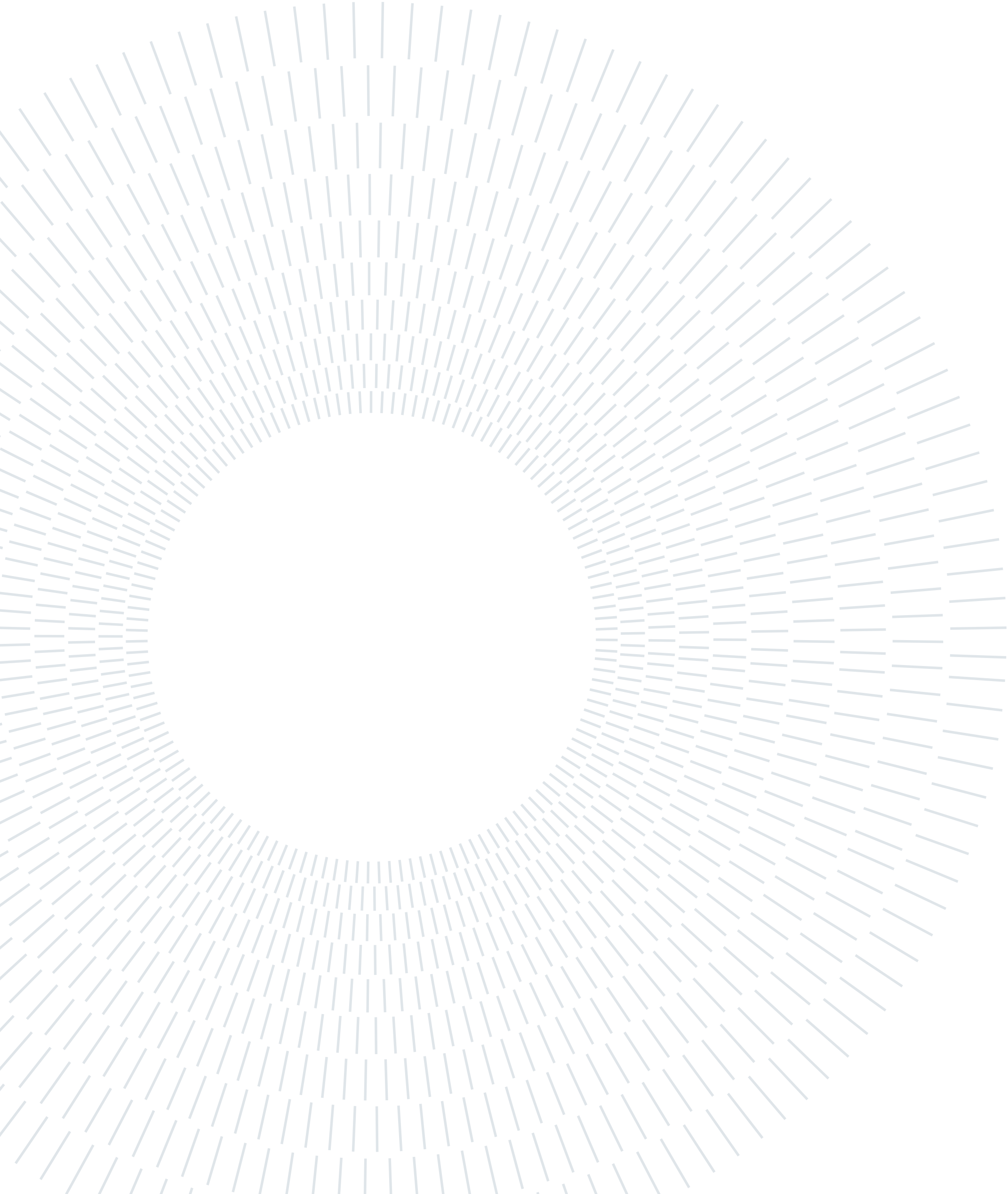
The majority of control systems for orbital relative motion are directed toward circular orbits and larger spacecraft. The direction of the growth in space exploration suggests that solutions must be developed for relative motion control, and applied to small spacecraft in eccentric orbits. The solution addresses the problem of relative orbital motion control, uncoupled from attitude control, utilising model predictive controllers and data-based system identification methods.

The implemented Model Discovery and Predictive Control (MDPC) solution is designed to work on smaller spacecraft with limited actuation, low telemetry bandwidth and high latency. The proposed solution considers the problem of bounding the separation of the chaser spacecraft relative to the target, with a constant pointing requirement of the chaser spacecraft towards the target. The onboard actuation system on the chaser spacecraft is assumed to be a single pair of opposite facing thrusters pointed at the target spacecraft.

The novelty of the solution is the regression based model discovery framework which utilises the sensor measurements of relative position and velocity to determine the local relative motion dynamics. The model discovery is made more stable and robust through stochastic methods. This is utilised by the prediction algorithm in the MPC to optimise the control sequence to maintain the chaser spacecraft within the specified range bounds.

The simulations of the MDPC performance suggest that the proposed system effectively satisfies the objectives under a wide range of orbital environments with perturbations and varying eccentricities. The resulting MDPC framework is also expected to discover the local dynamics in a faster and more insightful manner compared to neural network based learning algorithms. The MDPC system has a high degree of autonomy due to its minimal reliance on data from the ground-segment for control and guidance.

Keywords: relative, orbit, control, model, MPC, identification, discovery, predict



Sommario

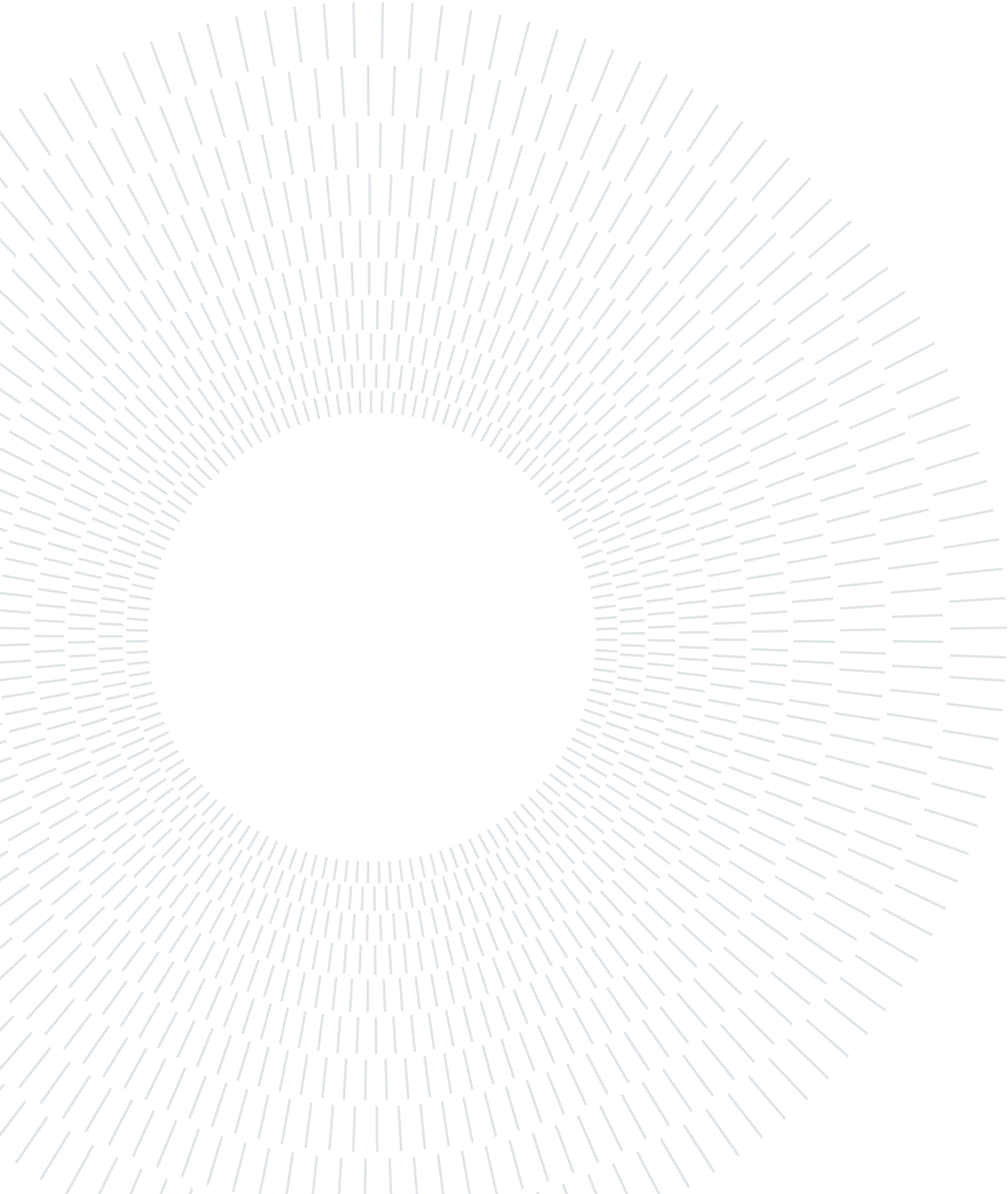
La maggior parte dei sistemi di controllo per il moto relativo orbitale sono orientati verso orbite circolari e veicoli spaziali di grandi dimensioni. I recenti sviluppi dell'esplorazione spaziale suggeriscono che debbano essere sviluppate soluzioni per il controllo del moto relativo applicate a piccoli veicoli spaziali in orbite eccentriche. La soluzione proposta affronta il problema del controllo del moto orbitale relativo, senza la necessità del controllo dell'assetto, utilizzando sistemi di controllo predittivo e metodi di identificazione del sistema basati sui dati.

La soluzione MDPC (Model Discovery and Predictive Control) implementata è progettata per funzionare su veicoli spaziali di piccole dimensioni sottoattuati, con larghezza di banda per la telemetria limtata e con latenza elevata. La soluzione proposta considera il problema di delimitare la separazione del veicolo spaziale chaser rispetto ad un target, con una richiesta costante di puntamento del chaser rispetto al target. Si presume che il sistema di attuazione a bordo dello spacecraft inseguitore sia una singola coppia di thrusters opposti puntati verso il target.

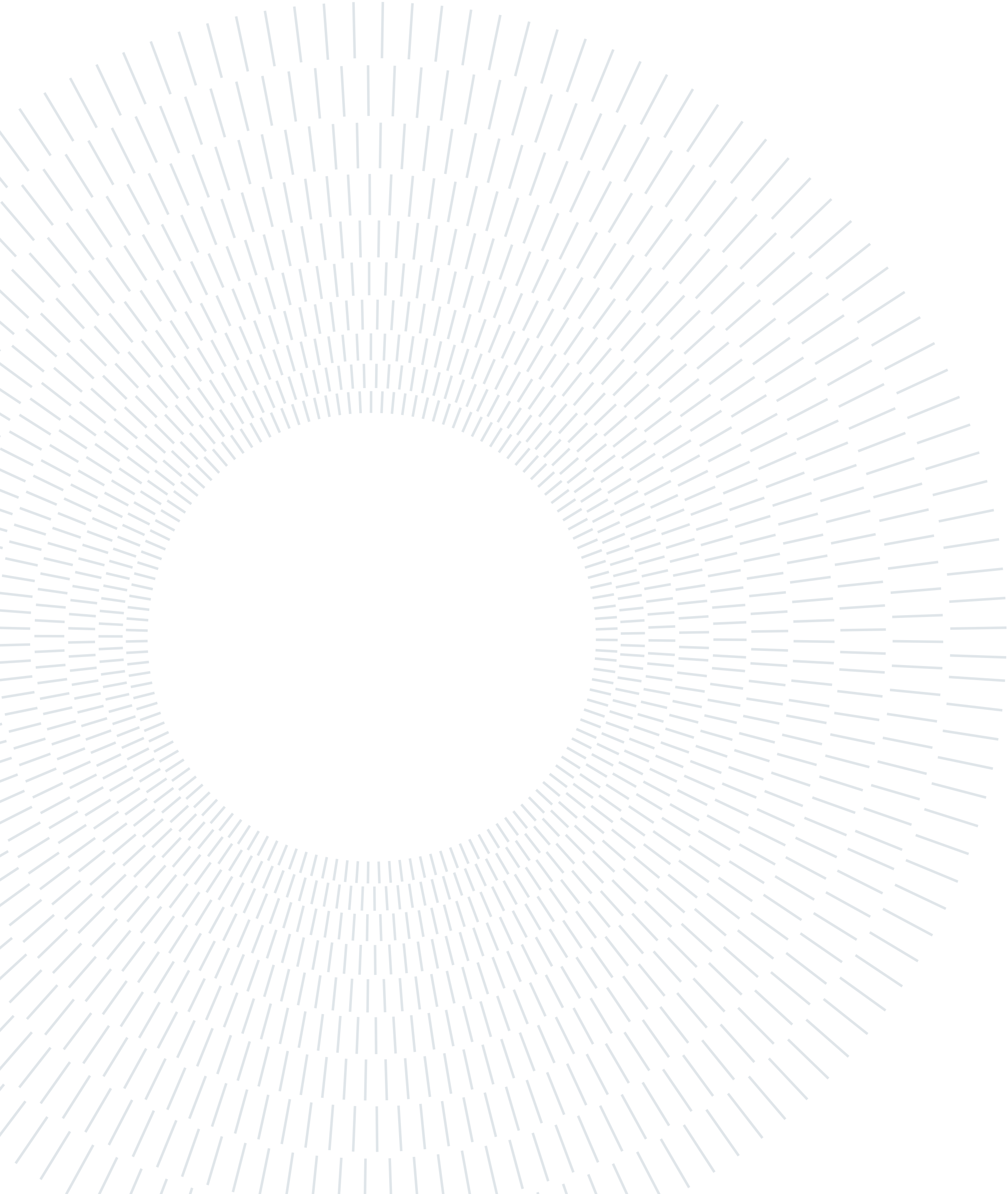
La novità della soluzione è il framework di identificazione del modello basato su una regressione che utilizza le misure del sensore di posizione relativa e della velocità per determinare la dinamica del moto relativo locale. L'identificazione del modello è resa più stabile e robusta attraverso metodi stocastici. Questo modello viene utilizzato dall'algoritmo di previsione nell'MPC per ottimizzare la sequenza di controllo necessaria a mantenere il veicolo spaziale chaser entro i limiti specificati.

Le simulazioni delle prestazioni dell'MDPC suggeriscono che il sistema proposto soddisfa efficacemente gli obiettivi in un'ampia gamma di condizioni orbitali con perturbazioni ed eccentricità variabili. Ci si aspetta inoltre che il framework MDPC proposta si in gradio di identificare le dinamiche locali in modo più rapido e approfondito rispetto agli algoritmi di apprendimento basati sulle reti neurali. Il sistema MDPC ha un alto grado di autonomia grazie alla sua dipendenza minima dai dati ottenibili del segmento di terra utilizzati per il controllo e la guida.

Parole chiave: moto relativo, controllo, modello, MPC, identificazione, predizione



Contents



1 | Introduction

The motion of spacecraft in formation with the goal of approaching a common final position and velocity with the intention to meet is called orbital rendezvous. This forms an important field of study for contemporary research and spaceflight operations. The wide range of applications stemming from such rendezvous operations include orbital docking for resupply, spacecraft repair as in the case of the Hubble space telescope. Most of the orbital rendezvous are performed in larger spacecraft due to the lack of complex hardware in smaller satellites.

The advent of cheaper and more reliable small satellites have enabled complex missions undertaken by the small satellites. However, there is not much advantage to docking two small spacecraft as there is in formation flying. Formation flying is the process of maintaining a coordinated relative orbital motion at relatively close proximity. An orbital proven example of such technology is the Tandem-X SAR satellite missions.

Historically, the first rendezvous attempts were successfully made by the crew of the Gemini mission on 16 March 1966. Neil Armstrong and Dave Scott successfully docked with the Agena module performing the historic feat. Several unmanned docking attempts were made until the Cosmos 186 spacecraft docked with Cosmos 188, becoming the first unmanned spacecraft to perform docking. Since then, most of our human space exploration in the context of the International Space Station has employed docking technologies. Some deep space rendezvous and proximity manoeuvres include the DART and Hayabusa missions to distant minor bodies. Most of the contemporary space-faring nations already have proven rendezvous technologies or are currently under development stages.

The primary branch of cutting-edge research in this field is enabling autonomous control system development for orbital rendezvous and proximity operations. A coherent global effort is visible towards development of autonomous control systems for small satellites for formation flying, proximity operations and rendezvous. Due to the abundance of opportunities and the relatively high launch cadence, small satellites have become key-enablers in the field today. The primary research sub-domain for the near-future seems to be in the direction of complete control autonomy and enabling self-contained control architectures independent from ground support.

The typical phases of spacecraft rendezvous are diagrammatically represented in Fig. 1.1. A common notation is adopted to refer to the two spacecraft undertaking this manoeuvre and differentiate between them. The spacecraft in the process of actively approaching another spacecraft is called the "chaser" and the other spacecraft is called the "target". Understanding the reasoning behind the naming scheme is very trivial.

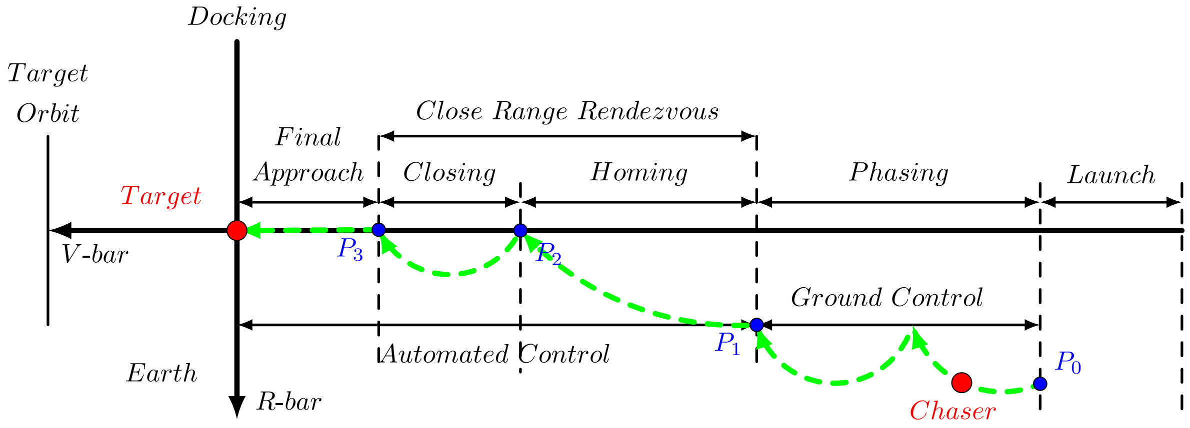


Figure 1.1: Typical spacecraft rendezvous and docking process (reproduced from [27]).

The usual orbital rendezvous process begins with a ground controlled orbital phasing, which brings the chaser vessel orbit closer to the target orbit. Upon completion, the automated control begins with the homing phase, where the chaser localises the target vessel and proceeds to move towards it. The automated control then initiates the closing phase to fine-tune the approach of the chaser vessel within close range. The last phase is the final approach phase undertaken by the automated control system, where the chaser vessel is already within close proximity of the target with the same relative velocity and orbital shape. The final approach phase is responsible for bringing the chaser vessel to the docking position by gradually reducing the relative range between the two vessels.

The primary mode of guidance in the automated phases of the rendezvous process is the relative navigation. There is no ground-segment support and guidance based on an inertial reference system towards the control of the chaser vessel. The final phase utilises very strict constraints and tolerances to maintain a rectilinear relative trajectory with a 6-**DoF** controller.

1.1. Background

Advancing frontiers in space exploration necessitate cutting-edge technological progress in autonomous space exploration. There is no lack of problems to be solved in this domain from an engineering standpoint. However, the current work will focus on the spacecraft control problem under an autonomous framework for maintaining a bounded relative range with respect to its target. While this problem is not novel and might seem trivial due to its simplicity and existence of time-tested solutions, the scope of the work presents a novel method and several applications that might be ideal for the proposed method. To understand how the current work integrates into the arena of available solutions, it is prudent to discuss the historicity and available alternative solutions.

The primary objective of the dissertation is to investigate a coupled system implemented through mathematical models of the spaceflight mechanics, data-based numerical methods and control theory. The mathematical formulations are discussed through the derivation of the relative orbital dynamics governing the chaser spacecraft motion relative to the target. The investigations concerning the control theory are focused through the implementation of a model predictive controller capable of controlling the aforementioned relative motion. Data-based methods are used as a tool to “discover” the system dynamics - model - that will be implemented in the model predictive controller.

This dissertation attempts to implement a control system capable of solving the autonomous relative motion control problem in orbits of arbitrary eccentricities. The novelty of the approach presented in this dissertation is the implementation of a model discovery algorithm to identify the concurrent local relative dynamics governing the relative motion, through sensor measurements. This investigation aims to find a fast, stable and robust model discovery framework which can be implemented with relatively low computational loads in model predictive control architectures for relative orbital position control applications in an elliptic orbit.

The classical solution to the relative orbital motion is provided by the Hill-Clohessy-Wiltshire model, through linearised time-invariant ODEs. The validity of this model is limited to the circular orbit of the target spacecraft and small initial separations between the target and the chaser. The solutions for arbitrarily eccentric target orbit are manifested through a time-varying nonlinear ODEs. The solution proposed by Yamanaka-Ankerssen is one such model, but also applicable to small initial separations. The proposed MPC in this dissertation utilises a constantly linearised dynamics to approximate a full nonlinear time-variant dynamics.

Historically, the relative motion of the chaser spacecraft is controlled through classical control theory implemented through **PID** controllers or linear-quadratic-Gaussian controllers. The cutting edge however, explores the utility and implementation of a novel model predictive controllers. This investigation utilises model predictive control where the model plays a critical role in its performance. The given algorithm, however, distinguishes itself from typical approaches by assuming no knowledge of the current dynamics governing the relative orbital motion. The proposed framework "discovers" the model using data-based methods, employing the state measurements or estimations available.

The term "guidance" in the orbital control jargon, refers to long-term manoeuvres or planning in the scale of the orbit. While smaller manoeuvres, such as maintaining requested attitude or relative position, constitute "control" of the spacecraft. In the context of this dissertation, the term "control" also refers to maintaining the chaser spacecraft within the set limits of relative range. The typical ranges where the **MPC** is envisioned to work ranges from a few hundred metres to hundreds of kilometres. Hence, the classical meaning of guidance can also be applied to the functions performed by the **MPC** within some specific cases.

1.2. Dissertation Scope

The dissertation proposes a control system based on the mathematical formulations of the orbital relative motion dynamics, principles of the model predictive control using "learned" plant models. The detailed derivation of the principles are discussed in their respective chapters, concluding with the integration and testing of the closed-loop control system. The mathematical formulations are derived with assumptions of Newtonian two-body problem and Keplerian orbits. The only deviations from these assumptions are introduced in the form of J2 perturbations.

The model predictive controller is designed without detailed derivations of the state estimator and the coupled attitude determination and control systems. This is done explicitly to discuss and highlight the novelty of the proposed solution. Derivations and discussions about the attitude control system and the state estimators would not contribute to the demonstration of the aforementioned novelty. Stochastic disturbances are neglected in the state estimation algorithm, which is justified by the principle of separation of estimation and control. Further resources regarding this principle can be found in Ref [9].

1.3. Motivation

The case of circular “target” orbit has been thoroughly investigated through linear models and it reflects in the vast majority of applications of such control schemes for orbital rendezvous in circular orbits. The relative scarcity of linearised models addressing the control schemes for elliptical “target” orbits cannot be denied. Upcoming challenges accompanied by the miniaturisation of satellites necessitate a closer examination of relative dynamics-based control schemes and their applications for the elliptical “target” orbit case.

The literature survey undertaken indicates an overwhelming volume of study in the domain of linear **MPC**s. The primary disadvantage of the linear models employed in model predictive controllers is the model divergence. Due to the linearisation, the plant dynamics used within the **MPC** are not accurate and quickly lose accuracy due to model dynamics diverging from the actual evolving dynamics i.e. “ground truth”. These limitations constrict the applications to smaller separations between the chaser and the target spacecraft. The solution proposed in this thesis aims to address this problem and propose a robust solution.

The linearised dynamic models also lead to inefficiencies in the control loop behaviours. This is evident due to inaccuracies in the model contributing to worse performance through propellant over-expenditure, discontinuous control sequence, etc. The final motivation arises from the need to increase the degree of autonomy of the control system. This is done through minimising the role of ground-segment telemetry in the control system. Considerations of a low-bandwidth telemetry and higher latency resistant control system are implemented.

Hence, this study explores the development of a model “discovery” algorithm that offers a fast and simple method to model the local dynamics and use it in the prediction algorithm for the model predictive controller. To improve the robustness and stability of such a model “discovery” framework, the possibility of including the perturbations into the “discovered” local dynamics is explored.

1.4. Objectives

The derivation of the local plant-model for relative orbital motion via data-based methods in an autonomous framework is a challenging objective. A necessary attribute of the proposed framework for model “discovery” is the reduced reliance on ground-link, simplicity and speed.

The primary objective is the design and implementation of a model predictive controller which aims to prove the thesis of this dissertation. The following features are formulated under the scope of the presented work:

1. To create a tool for identifying the local orbital relative motion dynamics and utilising it for optimising the control input sequence of the controller.
2. To provide a framework for control system design to tackle constantly evolving local dynamics under limited ground-segment support.
3. To enable relative range bounding control for arbitrary orbital eccentricities.
4. The proposed method of model "discovery" reduces the complexity and computational load when compared to alternative methods implemented through machine learning, deep learning, support vector machines, etc.

The next step is the realisation of a simulation and analysis environment and conducting numerical experimentation campaigns.

1.5. Literature Survey

This section provides a summary of the important aspects, relevant history and evolution of research in the concerned topics. This helps in understanding where the state of art exists and how does the proposed work fit into the arena of contemporary research.

1.5.1. Relative Orbital Mechanics

The basis of modern tools and formulations utilised to model the relative orbital motion were laid by the works of Hill-Clohessy-Wiltshire. Who developed a set of equations describing the motion of a chaser spacecraft in the target centred rotating reference frame. These equations are classically called the **HCW** equations of motion [13, 19]. The primary drawback was the scope of these equations were limited to circular orbits with very small initial separations, on the order of less than a kilometre.

These drawbacks were addressed by several future works by Lowden and Tschauner and Hempel. The former developed a linearised solution for elliptical orbits while the latter formulated simultaneous differential equations for the same [43]. The problem of application of these equations for arbitrarily eccentric orbits were still left unsolved.

The first steps in this direction were taken by Carter [12], with his proposed **STM** solution as a function of the true-anomaly of the target spacecraft. The work was advanced by the groundbreaking generalisation of theses solutions into a explicit time-dependant **STM** solution provided by Yamanaka-Ankerson [45]. They provided a simple exact analytical solution of relative motion for orbits of arbitrary eccentricity.

The Yamanaka-Ankerson solutions are one of the most widely used methods to model the relative motion. Several improvements and particular solutions for minimum-time rendezvous[1] and introduction of higher order terms for accuracy[26] and adoption of the Yamanaka-Ankerson solution into a spherical coordinate frame[24] were made.

Attempts at considering the various perturbations into the generalised solutions were made by Ross[33] through inclusion of spherical harmonics to account for J_2 perturbations. Due to the complex nature of these solutions, Schweighart and Sedwick proposed an approximation to utilise time-invariant coefficients and make them computationally efficient[39].

While most of the solutions were presented in a Cartesian reference frame, efforts were made to utilise the **GVE** to aid a higher order accuracy due to the nature of the time-derivatives of orbital elements being smaller and hence offering robust numerical calculus performance.A state transition matrix capable of propagating spacecraft with large separations in elliptical orbits and incorporating the effects of J2 is presented in Ref. [8]. The dynamics of the relative motion problem in a perturbed orbital environment is exploited based on Gauss' variational equations by Okasha and Newman in Ref. [29]. Another approach, utilising difference in argument of latitude, the difference in orbital radii, and their first-order derivatives to describe relative trajectories is presented by Baranov in Ref. [5].

For a more detailed overview on the historicity and various concurrent models of relative motion, refer to the work presented in Ref. [3, 27].

1.5.2. Control System

The introduction of a model based predictive controller algorithm is a relatively novel development in the field of control theory. The optimal control theory formed the basis of the development of model predictive control. The works of Kalman [22, 23] in 1960s, laid the foundations for optimal control. His work focused on infinite-horizon optimal control solutions. This led to the formulation of the **LQR**. These powerful control strategies focused on minimising a quadratic cost function to find a instantaneous control input subject to optimised infinite-horizon.

However, they were plagued by the lack of constraints and not considering the actual plant nonlinearities. The works in this field in the next decade focused on using an approximate formulation of the plant dynamics to predict the effect of control sequence on the future states of the system. This prediction was optimised subject to satisfaction of constraints based on a dynamic plant model propagation. Such methods were called **DMC**, proposed by Cutler and Ramaker [2, 15].

Initial predictive controllers were not stability guaranteeing algorithms due to finite optimisation horizons and no information about the stability of the underlying plant dynamics. This was solved by the introduction of an impulse based model excitation and finite horizon optimisation of a quadratic cost function. This was implemented in the first computer application called **IDCOM** [32].

Further improvements included using a linear step input excitation and using least squares optimisation algorithms was presented in the work by Cutler and Ramaker in 1980 [15].

The modern generalised formulation of **MPC** was driven by the introduction of two cost functions concerning the outputs and the control sequence. This algorithm allowed a better control over the optimisation space to augment the stability margins. The work proposed by Marquis and Broustail in 1988 [28], introduced the utilisation of modern state space models into the predictive control framework.

A wide range of modern applications of **MPC** utilise the state-space formulation with two cost functions. The work proposed in this dissertation utilises the same framework of the **MPC**.

To understand the specific application of orbital control strategies, the following references are recommended. For general digital control for relative orbital control, a good base can be formed through Ref. [17]. For a deeper investigation on applications on control of formations and docking, refer to Ref. [11].

1.6. Primary Contributions

The primary objectives and contributions of the presented work is in the field of model predictive control for relative motion control of spacecraft. However, a detailed derivation, justifications and results of the work will be presented further. The main contributions are listed below:

1. A mathematical, numerical and computational framework for local relative motion dynamics discovery.
2. A novel method to integrate data-based model discovery methods into model pre-

dictive controllers for in-space applications with higher autonomy.

3. A distinctive and more accurate prediction model is generated compared to alternative methods for nonlinear problems (elliptical orbits with higher initial separation).

4. The layout of a control/guidance algorithm with novel formulation of prediction model is provided accounting for model predictive controller parameter variation.

1.7. Prospective Applications

The motivation for developing a framework for MDPC (Model Discovery and Predictive Control) controller for relative orbit motion control is to broaden the scope of application and propose a general and flexible method for varied use cases. There can be several potential applications for the proposed framework, however, a few of them are enlisted below to create a general idea.

The application of the framework is foreseen in the following scenarios:

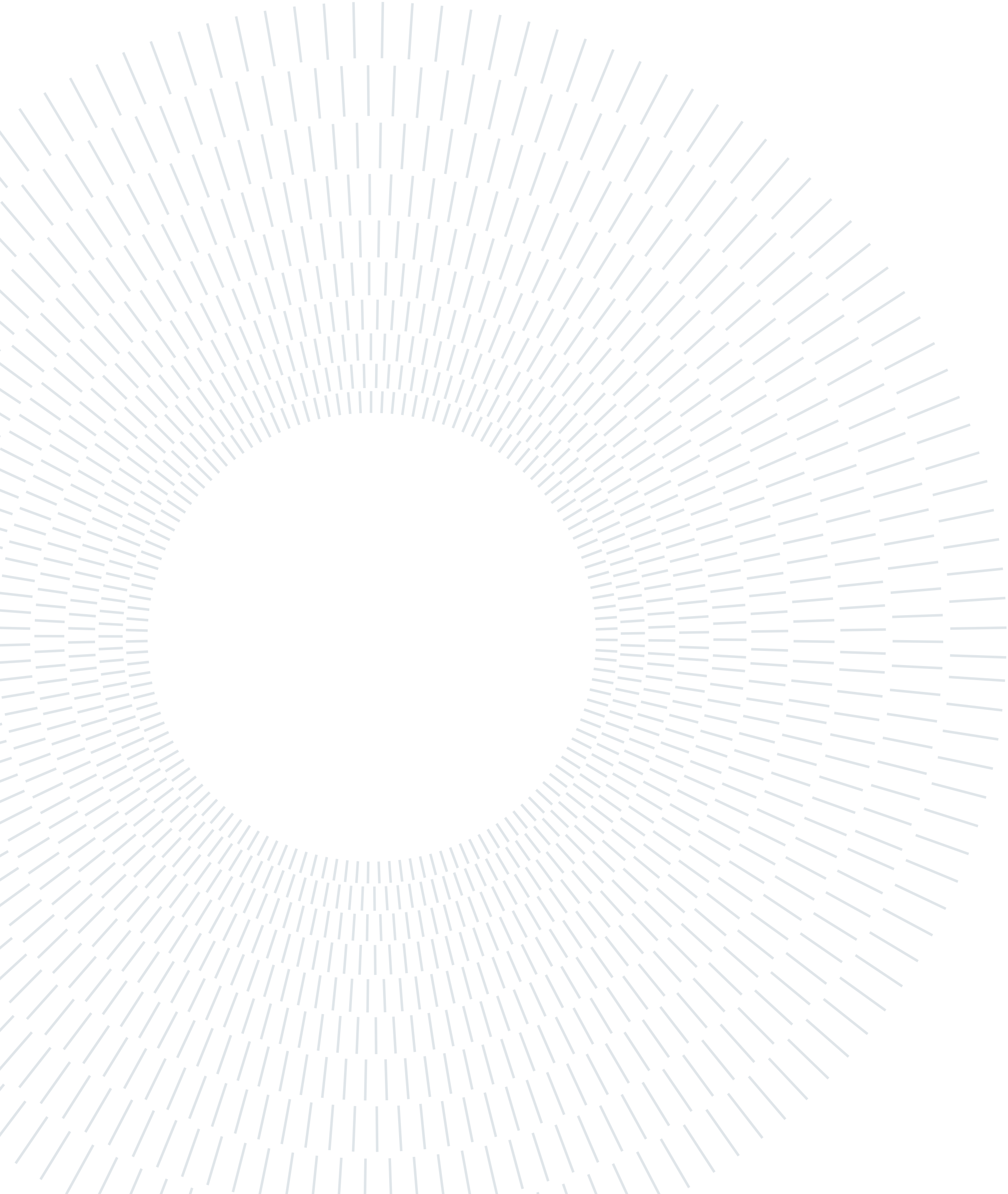
1. Satellite inspections or optical observations under dominant nonlinear relative motion dynamics.

2. Companion satellite relative range bounding for communications or science applications.

3. Deep space applications that reduce the possibility of on-ground support for control and guidance of relative motion.

4. Relative motion control around celestial bodies that deviate significantly from the Keplerian formulation of orbits, such as asteroids, the presence of mascons leading to gravitational anomalies, etc.

5. Fast-changing relative motion dynamics laws necessitate prediction methods to adapt quickly to the local dynamics, such as low orbits around the primary attractor.



2 | Mathematical Formulations

This chapter introduces the mathematical models that will form the basis for further development of the control system and the model discovery framework. The models formulate the relative orbital motion dynamics, used in the prediction algorithm of the model predictive controller. These models are derived from the basic laws of Newtonian mechanics and Keplerian orbital motion. Finally, the mathematical model describing the mass of the spacecraft is formulated as a conclusion of this chapter.

The derivation of the mathematical formulations in this chapter have been reproduced from multiple sources, Ref. [3, 18], and assembled into a desirable model. The frame of reference used for relative motion analysis is the **LVLH** frame, diagrammatically shown in Fig. A.1

2.1. Relative Orbital Motion in LVLH Frame

The basic control problem of the dissertation is focused on the control of the relative range of a spacecraft relative to another spacecraft in proximity. The spacecrafts forming this configuration are referred to as the target and the chaser spacecraft. The target spacecraft forms the centre of the frames of reference in which the chaser spacecraft motion is described. The objective of the chaser spacecraft is to maintain a relative trajectory or positional configuration in this target spacecraft centred reference frame. The only spacecraft under control is assumed to be the chaser spacecraft, which needs to perform a rendezvous manoeuvre.

2.1.1. State-Space Representation

The primary aim of this section is to develop mathematical models to describe the relative motion of the chaser spacecraft that can be utilised for control system design. This is achieved by deriving a state-space representation of the mathematical model of **NERM** developed in Section A.

The derivation of the state-space representation/model starts with the assumption of the spacecraft state vector:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{2.1}$$

The eq. 2.1 defines a state vector which will be employed in the control system. The scalar components of the vector are defined as follows:

- x, y and z are the relative position components defined in the orthogonal unit basis vectors of the **LVLH** frame described in Fig. A.1.
- $\dot{x}$, $\dot{y}$ and $\dot{z}$ are the relative velocity components defined in the orthogonal unit basis vectors of the same **LVLH** frame.

Now, we can combine our knowledge of the **NERM** developed in Section A through eq. A.19, eq. A.20 and eq. A.21, to formulate the relationship between the scalar components of the state vector and their respective first temporal derivative:

$$\frac{dx}{dt} = \dot{x} \tag{2.2}$$

$$\frac{dy}{dt} = \dot{y} \tag{2.3}$$

$$\frac{dz}{dt} = \dot{z} \tag{2.4}$$

$$\frac{d\dot{x}}{dt} = 2\dot{f}\dot{y} - 2\dot{f}y\frac{\dot{r}_t}{r_t} + x\dot{f}^2 + \frac{\mu}{r_t^2} - \frac{\mu}{r_c^3}r_t - \frac{\mu}{r_c^3}x + \frac{u_{dx}}{m_c} \tag{2.5}$$

$$\frac{d\dot{y}}{dt} = -2\dot{f}\dot{x} + 2\dot{f}x\frac{\dot{r}_t}{r_t} + y\dot{f}^2 - \frac{\mu}{r_c^3}y + \frac{u_{dy}}{m_c} \tag{2.6}$$

$$\frac{d\dot{z}}{dt} = -\frac{\mu}{r_c^3}z + \frac{u_{dz}}{m_c} \tag{2.7}$$

By incorporating the disturbance forces described in eq. A.15 into the state-space model based on state vector $\boldsymbol{x}$ defined in eq. 2.1, we can generalise the set of equations described above (eq. 2.2 ... eq. 2.7) to form a matrix equation. This would result in the derivation of the state-space form that forms the basis of the model for control system:

$$\dot{\mathbf{x}} = \mathbf{A}_{NL}\mathbf{x} + \mathbf{B}\left(\mathbf{u} + \mathbf{u}_p\right) + \mathbf{V} \tag{2.8}$$

where, matrix $\boldsymbol{A}_{NL}$ is called the state matrix for non-linear system and defined as:

$$\mathbf{A}_{NL} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \dot{f}^2 - \frac{\mu}{r_c^3} & -2\dot{f}\frac{\dot{r}_t}{r_t} & 0 & 0 & 2\dot{f} & 0 \\ 2\dot{f}\frac{\dot{r}_t}{r_t} & \dot{f}^2 - \frac{\mu}{r_c^3} & 0 & -2\dot{f} & 0 & 0 \\ 0 & 0 & -\frac{\mu}{r_c^3} & 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

Input matrix namely, matrix $\boldsymbol{B}$ is defined as:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_c} & 0 & 0 \\ 0 & \frac{1}{m_c} & 0 \\ 0 & 0 & \frac{1}{m_c} \end{bmatrix} \tag{2.10}$$

Finally, the additional vector $\boldsymbol{V}$ incorporates the non-linear terms:

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & 0 & \mu\left(\frac{1}{r_t^2} - \frac{r_t}{r_c^3}\right) & 0 & 0 \end{bmatrix}^T \tag{2.11}$$

There was no assumptions in the above derivations which enables us to use the equations for arbitrary eccentricities. We can linearise the eq. A.18 by assuming $\rho \ll r_t$:

$$\begin{aligned} r_c &= \sqrt{(r_t + x)^2 + y^2 + z^2} \\ &= r_t\sqrt{1 + \frac{2x}{r_t} + \frac{x^2 + y^+ z^2}{r_t^2}} \\ &\approx r_t\sqrt{1 + \frac{2x}{r_t}} \end{aligned} \tag{2.12}$$

The expression for $\frac{\mu}{r_c^3}$ can now be written using binomial theorem as:

$$\begin{aligned} \frac{\mu}{r_c^3} &\approx \frac{\mu}{\left(r\sqrt{1 + \frac{2x}{r_t}}\right)^3} \\ &\approx \frac{\mu}{r_t^3}\left(1 + \frac{2x}{r_t}\right)^{-\frac{3}{2}} \\ &\approx \frac{\mu}{r_t^3}\left(1 - \frac{3x}{r_t}\right) \end{aligned} \tag{2.13}$$

Now, eq. A.17 can be approximated by substituting eq. 2.13 in it. This results in:

$$\ddot{\mathbf{r}}_c \approx -\frac{\mu}{r_t^3}\left(1-\frac{3x}{r_t}\right)\begin{bmatrix} r_t + x \\ y \\ z \end{bmatrix} + \frac{1}{m_d}\begin{bmatrix} u_{dx} \\ u_{dy} \\ u_{dz} \end{bmatrix} \approx -\frac{\mu}{r_t^3}\begin{bmatrix} r_t - 2x \\ y \\ z \end{bmatrix} + \frac{1}{m_d}\begin{bmatrix} u_{dx} \\ u_{dy} \\ u_{dz} \end{bmatrix} \tag{2.14}$$

Hence, we can write the linearised equations of motion linearised around small separations between chaser and target spacecrafts. By substituting eq. 2.14 into eq. 2.5, eq. 2.6 and eq. 2.7 and also assuming $\dot{f} = \omega$, we can get:

$$\frac{d\dot{x}}{dt} = \left(\omega^2 + 2\frac{\mu}{r_t^3}\right)x + \dot{\omega}y + 2\omega\dot{y} + \frac{u_{dx}}{m_c} \tag{2.15}$$

$$\frac{d\dot{y}}{dt} = \dot{\omega}x + \left(\omega^2 - \frac{\mu}{r_t^3}\right)y - 2\omega\dot{x} + \frac{u_{dy}}{m_c} \tag{2.16}$$

$$\frac{d\dot{z}}{dt} = -\frac{\mu}{r_c^3}z + \frac{u_{dz}}{m_c} \tag{2.17}$$

These equations can be called the linearised equations of relative motion (**LERM**). We can derive the state-space representation using the same procedure as before by applied to the **LERM**, while also assuming $k = \frac{\mu}{h^{\frac{3}{2}}} = const.$ and $\frac{\mu}{r_t^3} = k\omega^{\frac{3}{2}}$. It is apparent that only the state matrix $\mathbf{A}_{NL}$ is changing:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) & \dot{\omega} & 0 & 0 & 2\omega & 0 \\ -\dot{\omega} & \left(\omega^2 - k\omega^{\frac{3}{2}}\right) & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -k\omega^{\frac{3}{2}} & 0 & 0 & 0 \end{bmatrix} \tag{2.18}$$

Finally, a linearised version of eq. 2.8 can be written as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\left(\mathbf{u} + \mathbf{u}_p\right) \tag{2.19}$$

This concludes the derivation of the state-space model responsible to describe the governing equations of relative motion. However, we also need to account for the state-space model describing the behaviour of the chaser spacecraft with regards to its propellant expulsion and the corresponding dynamics associated with its mass. This is addressed in the Section 2.2.

2.2. Mass Model of Chaser Spacecraft

While it is a complicated process to model the mass of the chaser spacecraft, a choice is made to derive a simplistic state-space representation for the same. This is deemed adequate for primary investigations and evaluating the performance of the control system, which is the primary objective of the thesis. The model is derived based on the thrust equation of a rocket motor with reference to the specific impulse of the rocket motor shown in eq. 2.20. The specific impulse employed in this model is the net equivalent specific impulse of the propulsion system used in the chaser spacecraft, whose derivation is referred from Ref. [41].

$$\dot{m} = \frac{F_{thrust}}{I_{sp} g_0} \tag{2.20}$$

In the above equation, $\dot{m}$ denotes the propellant mass expulsion rate due to the thrust actuation in the chaser spacecraft. F_{thrust} represents the actuation force of the thrusters and the I_{sp} is their net equivalent specific impulse. Finally, g_0 is the reference value of the gravitational acceleration on the surface of the Earth, assumed to be $g_0 = -9.81\frac{m}{s^2}$. An important point to note while employing this equation is to utilise the scalar magnitude of the actuation force which will always be positive irrespective of the direction of the actuation.

Let us assume a state vector which represents the mass of the chaser spacecraft. The choice is very trivial, as mass is not a vector. This results in the obvious choice of $\boldsymbol{x}_p$ equivalent to the actual mass of the propellant expelled by chaser spacecraft, resulting in a state scalar. Now the state-space model of the chaser spacecraft's expelled mass is:

$$\dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p \mathbf{u} \tag{2.21}$$

where the state matrix $\boldsymbol{A}_p$= [0] because the mass expelled has no effect on the rate of mass expulsion. The control vector $\mathbf{u}$ is defined previously in eq. A.14. The input vector $\boldsymbol{B}_p$ is defined as a row matrix with size $[1 \times 3]$.

To define the actual elements of the input matrix $\mathbf{B}_p$, we need to impose certain logical conditions in the modelling framework. The need for this arises due to physical limitations of the limited propellant and the necessity of the mass expulsion rate to be negative. It is trivial to model the input matrix $\mathbf{B}_p$ when the chaser spacecraft completely depletes its propellant:

$$\forall x_p \geq m_{p0} \implies \left\{ \mathbf{B}_p = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \right\} \tag{2.22}$$

where m_{p0} represents the initial propellant mass. Finally, to comply with the necessity of negative expulsion rates, we can define a switching condition:

$$\forall x_p < m_{p0} \implies \left\{ \mathbf{B}_p = sgn\left(\mathbf{u}\right)^T \cdot \begin{bmatrix} \frac{1}{I_{sp} g_0} & \frac{1}{I_{sp} g_0} & \frac{1}{I_{sp} g_0} \end{bmatrix} \right\} \tag{2.23}$$

where $sgn\left(\right)$ function denotes the *signum* function which returns the sign of each component of a vector as a vector. Now we can write an equation to model the current mass of the chaser spacecraft:

$$m_d = m_{dry} + m_{p0} - x_p \tag{2.24}$$

where m_{dry} is the mass of the chaser spacecraft devoid completely of all stored propellant. The current value of x_p will be determined for the current time during the simulation of the model through integration of eq. 2.21.

3 | Model Discovery

This chapter presents the methodology and primary reasoning behind the adopted model discovery methodology. The internal prediction model in the controller is derived from the results of the data-driven model discovery problem. Starting from the outline of the contended model discovery methods, the chapter provides justifications for shortlisting the chosen method before moving on to the adaptation of the procedure to the relative orbital motion prediction problem before finally concluding with the final model discovery framework.

3.1. General Concept

Governing equations are of primary importance in the field of engineering. Determining or knowing the governing equations allows understanding the physical phenomenon and provides insights into development of methods to control it. Classically, these equations are derived from the basic universal principles and conservation laws. However, contemporary research [7, 10, 30, 34] is conducted in fields where either the governing equations are unknown, partially known or influenced by a superposition of multiple factors, often too much to analytically consider or evaluate their respective independent degree of influences on the system. This necessitates a method to discover the governing equations behind such physical processes through measurement data.

Application of data-driven model discovery has been employed for a long time in the fields of vibrations, finance, fundamental physics, aerodynamics, chaos theory, etc. Oftentimes, data-driven model discovery methods are attributed to a low degree of physical insight into the physical process and higher computational loads. This is primarily due to the numerical nature of the problem solving algorithms. Such methods are essentially “blind” to the actual physical laws, they only ensure that the measurements are correlated via numerically determined coefficients to a high degree of accuracy without any insights on if such systems could actually exist.

The primary concern for such methods is to strike a balance between the model efficiency and descriptive capacity of the derived models. Often, models are either efficient and accurate numerically or analytically descriptive, but not both. It is this conundrum that drives the forefront of the cutting edge in this field. While many methods are available to solve this problem, every method prerequisites time-rich sensor measurements to ensure accurate model derivation.

If broken down to the lowest level of complexity, data-driven model discovery is akin to the process of curve-fitting through numerical regression. It computes the coefficients of the contribution of each input variable to the output variables via measurements of the same. To improve the accuracy of such methods, several approaches and considerations are available such as neural networks, model dynamic decomposition, principal component analysis, numerical regression, etc. Each of these methods offers a unique set of possibilities and features but depends heavily on the correct choice of candidate coordinates to describe the dynamics.

The optimal choice of the coordinates is usually at one's own discretion, driven by the intuition and cognisance of oneself. This introduces another avenue for our problem, namely the identification and employing the correct candidate coordinates that will be used to discover the system. This is addressed in the following sections in detail.

Ultimately, the evaluation of the performance of the model discovery methodology is primarily performed by its model prediction capabilities. The benchmarking is performed with reference sensor measurements independent of the measurement set used for model discovery. This is also described in this chapter along with the choice of performance parameters for benchmarking.

3.2. Model Discovery Methods

While many different methods exist for deriving the model of the system from measurements, the efficacy of some methods is better for the purpose of this dissertation. Primarily, model discovery is performed in either the time domain or the frequency domain. For the proposed solution, it is computationally and analytically preferred to employ time-domain methods. The justifications for such assumptions are driven by the formulation of relative orbital motion equations being formulated in the time domain.

One of the possible methods for approaching a model discovery problem is the classical machine learning or neural networks [4, 6, 31, 44, 46]. The idea behind the usage of classical neural networks in model discovery is the relative ease of modelling such model discovery architectures. Sometimes trivially referred to as "black-box" systems, neural networks are infamously less intuitive in their insights into the dynamics.

The major drawbacks of the neural networks based model discovery can be listed as follows:

- Neural networks are good at understanding systems through interpolation of training data-set. For purposes of **MPC**, prediction of the system states is essentially a forecast/extrapolation.
- Low level interpretability of the discovered models.
- Lack of conditioning methods to "inform" the neural network of physical constraints.
- Discovered model is usually not sparse i.e. the coordinates used for model identification is more than what is needed to describe it parsimoniously with fewest possible coordinates Ref. [10, 34, 35, 38].

When generalised and interpret-able models are required to be generated through model discovery, it is highly preferred to have parsimonious models that are described by the fewest possible coordinates with the maximum contribution to the system dynamics. Such methods are referred to as sparse model identification methods. Such systems are most difficult to implement via neural networks due to the conflicting nature of neural networks being anti-parsimonious. Nevertheless, methods such as symbolic regressions, simultaneous linear regressions and sparse regressions are available to be used for our applications.

Another feature of critical importance is the ability to inform the model discovery framework of any preexisting knowledge of the dynamics to aid the speed or the accuracy of the discovered system dynamics. A relatively novel method proposed in Ref. [10] by Brunton et al, presents a framework for sparse model discovery through numerical regression methods employing sensor measurements. While this method has been successfully utilised in several domains, it is extremely rare to find its applications in relative orbital dynamics and control. The proposed work of this thesis primarily focuses on the implementation of this strategy in the model prediction framework of the model predictive controller.

3.3. SINDy Algorithm

The **SINDy** algorithm was developed to offer a model identification framework which offers interpretability and also flexibility for generalisation. It offers a method to identify the dynamical governing equations based on sensor measurements. There have been several applications which show the interoperability and flexibility of this framework, from structures [25], chemical kinetics [20], ray optics [42] and fundamental physics [16].

These are just representative of the potential applications of this novel method. There exist provisions to introduce pre-existing knowledge of physical systems into this framework to aid the regression process to identify the model. Fundamentally, **SINDy** algorithm focuses on sparsity inducing regression methods and algorithms. It is also proven that the **SINDy** algorithm is faster, more robust and computationally light compared to neural networks [21].

While **SINDy** algorithm is a great method for the purposes of **MPC**, it is heavily dependant on the quality of the measurements, volume of measurements, sparsification function definition and also on the choice of the basis coordinates used to describe the dynamics in the regression problem.

For our application, we have already developed the mathematical models of our relative motion in Chapter. 2. We know the formulation, what we need to find is the coefficients of the governing equations that are applicable in the spatial and temporal locality of the chaser spacecraft. This would allow us to dynamically tune our **MPC** performance based on the local dynamics without a need for ground-segment support.

3.3.1. Discovery Framework

This section will lay down the mathematical and schematic descriptions of the discovery framework proposed by the **SINDy** algorithm. The fundamental basis of the algorithm is rooted in a regular multi-dimensional linear regression problem. Let us now familiarise ourselves with the jargon used in the context of the **SINDy** algorithm. The mathematical equation that describes such a problem in the context of **SINDy** algorithm is given by:

$$\dot{\mathbf{X}} = \mathbf{\Theta}\left(\mathbf{X}\right)\mathbf{\Xi} \tag{3.1}$$

where, $\dot{\mathbf{X}}$ is called the target function or target matrix, $\mathbf{\Theta}$ denotes the matrix of coordinates used in the regression also known as feature library and finally, $\mathbf{\Xi}$ is called the regression coefficient matrix. It is often the case in dynamics that the targets are the derivatives of a state vector $\mathbf{x}$ or a combination of functions whose domain is the state vector. This is represented by the $\mathbf{\Theta}$ being a function of the state vector, denoted as $\mathbf{\Theta}\left(\mathbf{X}\right)$. Most often, in the context of control systems or state space representations, the derivative of the state vector is only dependent on the state vector or a linear/nonlinear combination of states only. In our case, according to the mathematical models of the relative motion, the derivative of the state vector is only a function of the state vector as denoted in (eq. 2.2 ... eq. 2.7).

The diagrammatic representation of eq. 3.1 is given in the Fig. 3.1:

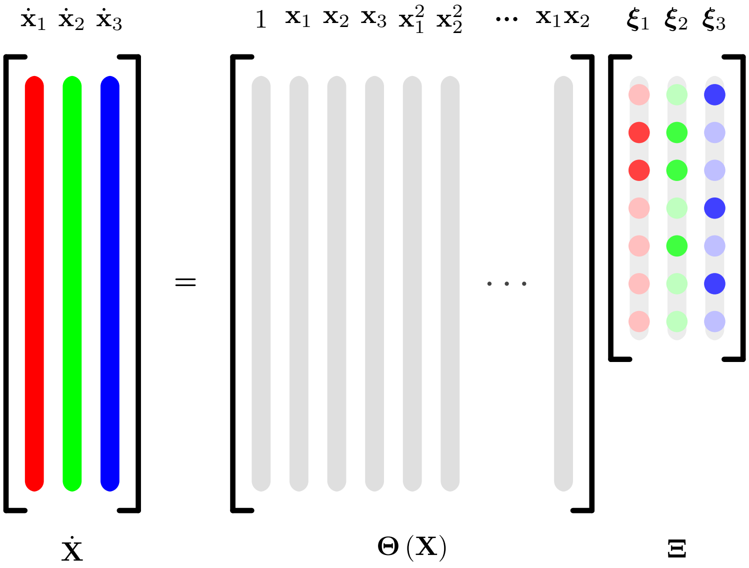


Figure 3.1: SINDy framework.

The coefficients matrix $\mathbf{\Xi}$ is characterised by several coefficients denoted by coloured circles. The intensity of the colour represents the magnitude of the coefficients. In the context of regression, the magnitude of the coefficients represents the degree of correlation of the corresponding feature in $\mathbf{\Theta}$ to the target in $\dot{\mathbf{X}}$. The correspondence of each feature to its target is represented by a unique colour.

The target and features matrices are populated by the measurement data acquired by the sensors. Due to the presence of noise and the nature of numerical methods, there will exist at least a weak correlation between the targets and features which are physically unrelated. This results in inaccurate coefficients in the regression process. This is represented in the Fig. 3.1, matrix $\mathbf{\Xi}$, through very mild correlation coefficients in lighter colours.

The process of eliminating the mild correlation coefficients through numerical methods or physical knowledge of the system results in a well-conditioned coefficient matrix which has just enough coefficients to describe the dynamics of the system. Such a coefficient matrix is called a sparse coefficient matrix. Sparse coefficient matrices offer good extrapolation/prediction performance because they capture the system dynamics accurately.

There exist methods to sparsify the coefficients, usually implemented through regression regularisers or optimisers. Such a process is figuratively represented below:

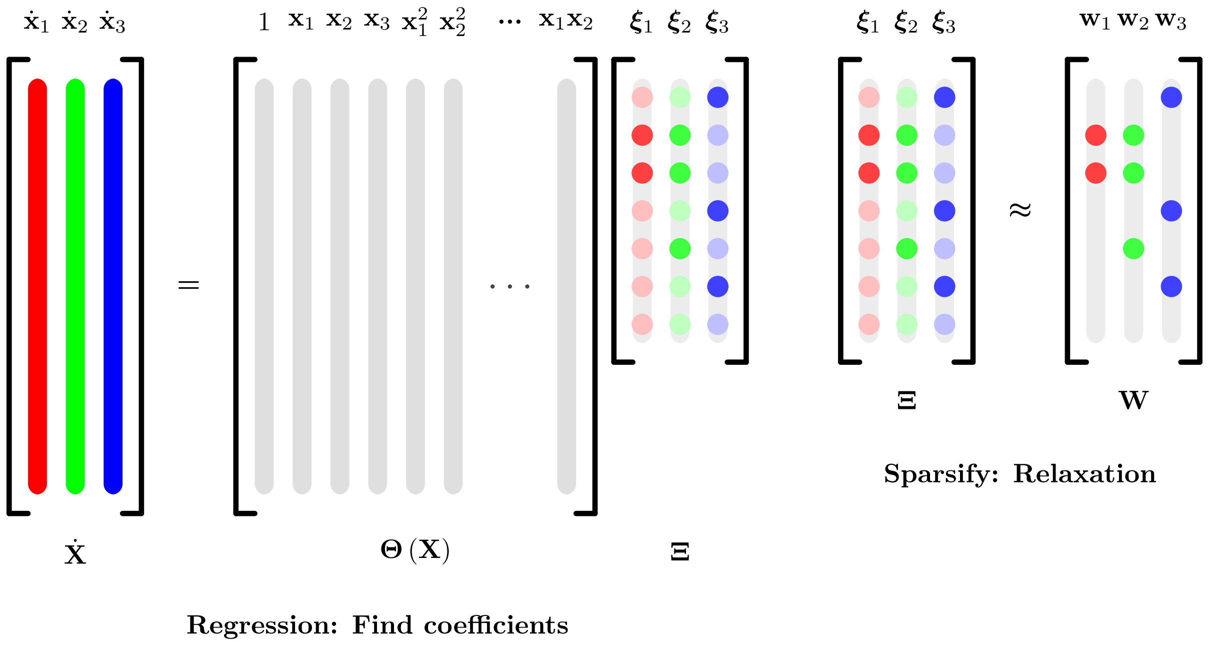


Figure 3.2: SINDy framework: Coefficient sparsification [48].

It is evident from the Fig. 3.2 above that the coefficients depicting mild correlation are eliminated and form the sparse/relaxed coefficients matrix $\boldsymbol{W}$.

An important point to note before proceeding further is the dependence of **SINDy** algorithm accuracy on the quality of the measurements. Any physical sensor measurement is characterised by the presence of noise and biases. Some workaround strategies are discussed to avoid performance degradation in the sections following this.

3.3.2. Candidate Library

The term "library" refers to the collection of measurements or functions that contribute possibly to the targets. Wherein the term "targets" refers to the collection of measurements or functions that form the **L.H.S** of the regression formulation. The choice of an optimum library aids in the accuracy and speed of model discovery. The prior knowledge of the underlying mathematical models helps us in identifying candidates for the library selection. Within the **SINDy** framework, the terminology used to describe the collection of terms that are prospective candidates for the regression coordinates is “features”. Hence, in this section, the terminology will be used interchangeably, as is in modern model identification methods.

The **SINDy** framework assumes that the model can be described as a linear combination of features. If the system under investigation does not obey such governing equations,

SINDy method fails to obtain good results.

From the discussions in the Chapter 2, concluding with eq. 2.18, we can proceed to formulate our regression feature library. To convert our understanding of the mathematical models into a regression formulation, we must write the state equation presented in eq. 2.19 in the formulation used in the **SINDy** framework, as shown in Fig. 3.1. Such a formulation can be expressed as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \Leftrightarrow \dot{\mathbf{X}} = \mathbf{\Theta}\left(\mathbf{X}\right)\mathbf{\Xi} \tag{3.2}$$

Rearranging by forming an augmented matrix by transposing the individual matrices:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \Rightarrow \dot{\mathbf{x}}^T = \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} \tag{3.3}$$

It is important to note that $\mathbf{X}$ represents the feature vector in the context of regression, while $\mathbf{x}$ represents the state vector in the context of relative orbital motion. While eq. 3.3 is completely valid, there are some considerations to be made before finalising the formulation.

To account for practical implementation of **SINDy** method, or any regression based methods, it is generally advised to keep the formulation very simple and try to minimise the unknown parameters. This strategy is key to reducing computational loads for regression, improving robustness of solution and also augment model identification accuracy.

Since measurements of external acceleration $\mathbf{a}_d$ can be obtained easily compared to obtaining accurate measurements of $\mathbf{u}$. Measuring the force imparted on the spacecraft by the actuators and the perturbations is impractical. Under such considerations, we can try to simplify our formulation by assuming a good mass state estimator. This allows us to have an accurate estimation of the spacecraft mass which could be measured and sampled to be used further in model discovery. As per eq. 2.10, $\mathbf{B}$ matrix can be completely determined if we have an accurate estimation of the spacecraft mass. By dividing matrix u by the mass of the chaser spacecraft m_c, we need to multiply $\mathbf{B}$ with m_c to balance the equation. Which results in:

$$\dot{\mathbf{x}}^T = \begin{bmatrix} \mathbf{x}^T & \frac{\mathbf{u}^T}{m_c} \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T m_c \end{bmatrix} = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_d^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \tag{3.4}$$

where $\mathbf{a}_d$ represents the disturbance acceleration discussed in eq. A.15 and:

$$\mathbf{B}_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

The final library used in the regression problem is determined to be the augmented matrix containing the relative state measurements $\mathbf{x}_p$ and the control acceleration measurements $\mathbf{a}_d$. The **SINDy** regression equation can be expanded using eq. 2.1, eq. 2.18 and eq. 3.5 and substituting into eq. 3.4 to derive an alternative formulation of the equation as:

$$\begin{gathered}\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_d^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^T = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ a_{dx} \\ a_{dy} \\ a_{dz} \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) & -\dot{\omega} & 0 \\ 0 & 0 & 0 & \dot{\omega} & \left(\omega^2 - k\omega^{\frac{3}{2}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & -k\omega^{\frac{3}{2}} \\ 1 & 0 & 0 & 0 & -2\omega & 0 \\ 0 & 1 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}\end{gathered} \tag{3.6}$$

We can alternatively expand the notation $\mathbf{a}_d$ into its individual contributions. This leads to the following equation:

$$\mathbf{a}_d = \mathbf{a}_f + \mathbf{a}_p \tag{3.7}$$

Where $\mathbf{a}_f$ corresponds to the acceleration imparted by the spacecraft actuation systems and $\mathbf{a}_p$ corresponds to the acceleration imparted by the orbital perturbations. Hence $\mathbf{a}_d$ represents the total external acceleration on the body. We can now rewrite the eq. 3.6 as:

$$
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^T = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ (a_{fx} + a_{px}) \\ (a_{fy} + a_{py}) \\ (a_{fz} + a_{pz}) \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) & -\dot{\omega} & 0 \\ 0 & 0 & 0 & \dot{\omega} & \left(\omega^2 - k\omega^{\frac{3}{2}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & -k\omega^{\frac{3}{2}} \\ 1 & 0 & 0 & 0 & -2\omega & 0 \\ 0 & 1 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}
$$

By splitting the external accelerations $\mathbf{a}_f$ and $\mathbf{a}_p$ terms:

$$
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}^T = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ a_{fx} \\ a_{fy} \\ a_{fz} \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) & -\dot{\omega} & 0 \\ 0 & 0 & 0 & \dot{\omega} & \left(\omega^2 - k\omega^{\frac{3}{2}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & -k\omega^{\frac{3}{2}} \\ 1 & 0 & 0 & 0 & -2\omega & 0 \\ 0 & 1 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ a_{px} \\ a_{py} \\ a_{pz} \end{bmatrix}^T \tag{3.9}
$$

By borrowing the same equation notation from eq. 3.3, we can formulate eq. 3.9 as follows:

$$
\dot{\mathbf{x}}^T = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} + \boldsymbol{\Upsilon}^T \tag{3.10}
$$

where $\boldsymbol{\Upsilon}$ denotes the matrix including the contribution of perturbation acceleration on the **R.H.S** in eq. 3.9.

The eq. 3.10 still does not conform to the **SINDy** equation, eq. 3.1. We can now introduce another assumption to make it compatible. According to Fig. B.1, the acceleration of the various perturbations are on the order of $\mathbf{a}_p < 10^{-4}\ \frac{km}{s}$. If the chaser spacecraft is actuated with higher forced accelerations $\mathbf{a}_f$ compared to $\mathbf{a}_p$, we can assume a minor role of the perturbations in the relative motion dynamics.

We can also note that most of the perturbations faced by the chaser spacecraft are also simultaneously faced by the target spacecraft. By assuming $\rho \ll r$, both spacecraft experience almost similar perturbation forces. Hence, the relative perturbation acceleration vector in the **LVLH** frame, $\mathbf{a}_p$, is negligible in magnitude even compared to the perturbation acceleration vector in the inertial frame of reference.

With these considerations, we can assume $\boldsymbol{\Upsilon}$ to be negligible and reformulate eq. 3.10 into:

$$\dot{\mathbf{x}}^T \approx \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \tag{3.11}$$

By comparing eq. 3.11 and **SINDy** equation, eq. 3.1:

$$\dot{\mathbf{x}}^T \approx \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \Leftrightarrow \dot{\mathbf{X}} = \boldsymbol{\Theta}\left(\mathbf{X}\right) \boldsymbol{\Xi} \tag{3.12}$$

and assuming that $\mathbf{X} = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix}$, we can conclude that:

$$\boldsymbol{\Theta}\left(\mathbf{X}\right) \equiv \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \Rightarrow \boldsymbol{\Theta}\left(\mathbf{X}\right) = \mathbf{X} \tag{3.13}$$

$$\boldsymbol{\Xi} \equiv \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \tag{3.14}$$

To successfully implement this equation into the **SINDy** framework, we must ensure that the measurements that constitute the library $\Theta\left(\mathbf{X}\right)$ and targets $\dot{\mathbf{X}}$ must be obtainable accurately and with a high measurement sampling rate. By observing eq. 3.9, we can see that all the measurements pertaining to $\Theta\left(\mathbf{X}\right)$ and $\dot{\mathbf{X}}$ are physically measurable directly or through a good implementation of navigational filters and/or state estimators.

3.3.3. Regulariser

The regression equation is determined and an optimum selection of the regression strategy has to be formulated to discover the system. **SINDy** algorithm employs an iterative regression solver that utilises an optimiser and a regulariser to iteratively improve the coefficient matrix. The default implementation utilises a **STLSQ** optimiser. The optimiser minimises a cost function that evaluates the regression fitness or accuracy. The objective function of the **STLSQ** optimiser based regression is denoted below:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \parallel \dot{\mathbf{X}} - \boldsymbol{\Theta}\left(\mathbf{X}\right)\boldsymbol{\Xi} \parallel_2 \tag{3.15}$$

For more control over the selection of coefficients in $\boldsymbol{\Xi}$, a thresholder is introduced. The thresholder imposes a constraint on the minimum absolute value any coefficient can take in $\boldsymbol{\Xi}$ to avoid elimination in the iterative optimisation.

For each iteration, the smallest correlation coefficients which are smaller than a threshold value are eliminated before proceeding to the next iteration. The mathematical equation for the thresholded **STLSQ** optimiser could be formulated as:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \parallel \dot{\mathbf{X}} - \boldsymbol{\Theta}\left(\mathbf{X}\right)\boldsymbol{\Xi} \parallel_2 + \lambda R\left(\boldsymbol{\Xi}\right) \tag{3.16}$$

where λ is a hyper-parameter associated with tuning the thresholder and $R\left(\cdot\right)$ denotes arbitrary regulariser function which promotes sparsity.

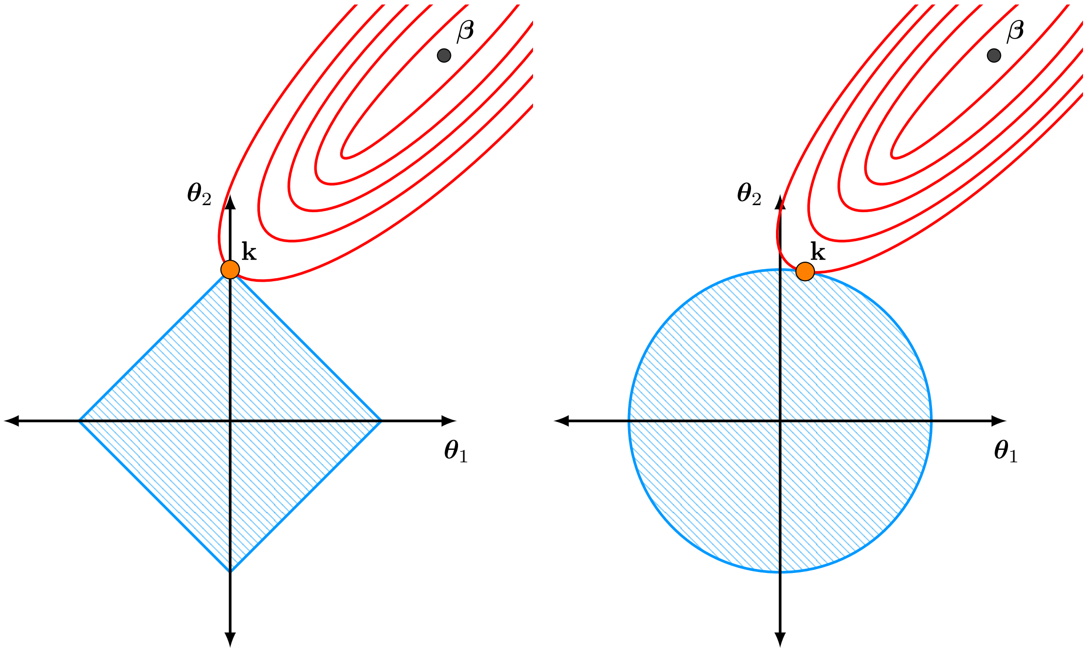
To maintain the convex nature of the objective function, we must avoid the $L0$ norm. Hence, typically, $L1$ and $L2$ norms are used as regularisers. The equations representing them are listed below:

$$||v||_1 = \sum_i |v_i| \tag{3.17}$$

$$||v||_2 = \sqrt{\sum_i v_i^2} \tag{3.18}$$

where v represents an arbitrary vector with components v_i. The key difference between these regularisers, are their sparsity promoting behaviour. Mathematically, these norms can be represented as hyper-surfaces in multi-dimensional function-spaces.

The shape of these surfaces are representative of how the regulariser promotes the iteration step along each dimension. To analyse the effects we can assume an arbitrary 2-dimensional function-space, which is graphically represented in Fig. 3.3.



(a) $L1$ Norm. (b) $L2$ Norm.

Figure 3.3: Regularisation comparison.

where θ_1 and θ_2 represent the coordinates that span the multi-dimensional function space of the coefficients in matrix Ξ. The point $\boldsymbol{\beta}$ represents the global minima of the objective function formulated in eq. 3.16. The optimisation of the objective function starts at a particular initial coordinate in the function space of θ_1 and θ_2 which is represented as the origin of the arbitrary coordinate axes. While only two coordinates are shown in the Fig. 3.3 for representational purposes, the principle remains the same even for higher dimensional function-spaces.

The $L2$ norm promotes the solution in iteration steps equally in all the coordinates, while the $L1$ norm promotes the solution along the basis vectors corresponding to the function space. Each iteration expands the volume spanned by the n-dimensional rhomboid and sphere, in the case of $L1$ and $L2$ norms respectively, until the volume spanned touches the minima $\boldsymbol{\beta}$.

The Fig. 3.3 represents one iteration step, which results in the solution moving from the origin of the axes i.e. initial value to the point denoted by $\mathbf{k}$. It is evident that the $L2$ norm solution does not promote the solution selectively in any direction, which results in non-zero coefficients along each coordinate axes. Whereas, $L1$ norm promotes the solution along a particular axis θ_2 which has a non-zero coefficient contribution while θ_1 remains zero. This is the sparsity promoting behaviour of the $L1$ norm, which ensures that the solution moves in a direction to keep the contributions of all but one axis non-zero. Hence, for the purposes of this thesis, a weighted $L1$ norm is used as a regulariser. The eq. 3.16 can be alternatively written as:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \parallel \dot{\mathbf{X}} - \boldsymbol{\Theta}\left(\mathbf{X}\right)\boldsymbol{\Xi} \parallel_2 + \lambda ||\boldsymbol{\Xi}||_1 \tag{3.19}$$

3.3.4. Thresholding

Referring to eq. 3.16, a higher value of λ would result in the elimination of more coefficients during the optimisation. While this would result in a sparse matrix, it might eliminate coefficients that are important but have a lesser value than λ. It is also important to note that **STLSQ** optimiser offers a single threshold value for all the coefficients in $\boldsymbol{\Xi}$.

Let us substitute eq. 3.13 and eq. 3.14 into eq. 3.19 to derive the objective function for our regression problem previously formulated. This results in the following expression:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \left\| \dot{\mathbf{x}}^T - \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_2 + \lambda \left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_1 \tag{3.20}$$

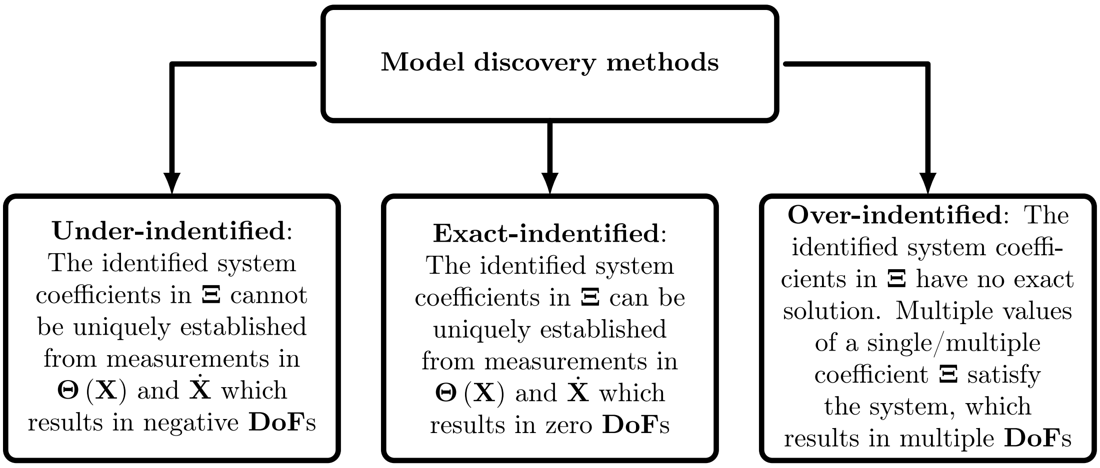
On closer inspection of eq. 3.9, we can see that the coefficients corresponding to $\boldsymbol{\Xi}$ vary a lot in their magnitudes. This means that a common thresholder cannot be used to drive regression optimisation. A variable thresholder uniquely specific to every element/coefficient in $\boldsymbol{\Xi}$ is required. Such a thresholder can be modelled as follows:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \left\| \dot{\mathbf{x}}^T - \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_2 + [\boldsymbol{\lambda}] \left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_1 \tag{3.21}$$

where $[\boldsymbol{\lambda}]$ denotes a threshold matrix with the same size as $\boldsymbol{\Xi}^T$. Each element in $[\boldsymbol{\lambda}]$ corresponds to a threshold on a unique and corresponding coefficient in $\boldsymbol{\Xi}$. We can utilise this to impose unique thresholds on each coefficient to control the regression iterations better.

3.3.5. Constraints

Although our regression problem is formulated, we need to use pre-existing knowledge of the system dynamics to "inform" the numerical regression method. Any system/model discovery method can be classified based on the resulting "discovered" model behaviour compared to the baseline system. The classifications are shown in the Fig. 3.4 below:



Under-indentified: The identified system coefficients in Ξ cannot be uniquely established from measurements in $\boldsymbol{\Theta}(\mathbf{X})$ and $\dot{\mathbf{X}}$ which results in negative **DoFs**

Exact-indentified: The identified system coefficients in Ξ can be uniquely established from measurements in $\boldsymbol{\Theta}(\mathbf{X})$ and $\dot{\mathbf{X}}$ which results in zero **DoFs**

Over-indentified: The identified system coefficients in Ξ have no exact solution. Multiple values of a single/multiple coefficient Ξ satisfy the system, which results in multiple **DoFs**

Figure 3.4: Model discovery methods classification

Our mathematical models are well defined and provide exact solutions to the relative orbital motion. The measurements chosen are physically possible to acquire with good temporal and spatial resolutions. This ensures that our model discovery methods are not under-identified. However, we need to ensure that the model discovery framework does not produce arbitrary non-unique coefficients that fit the measured targets but fail in model prediction due to lack of physical insight of the system. The coefficients discovered for such a system would result in an over-identified system. This is ensured by "informing" the model discovery framework about the physical dynamics through constraints.

The constraints on the coefficients in Ξ could establish relationships between the various individual coefficients or even include linear combinations of functions of the coefficients.

A method has been developed under the investigation of this thesis to impose physical constraints on Ξ. Recalling the formulation of Ξ as per eq. 3.14 and using eq. 3.9, Ξ can be expanded as:

$$\Xi = \begin{bmatrix} 0 & 0 & 0 & \left(\omega^2 + 2k\omega^{\frac{3}{2}}\right) & -\dot{\omega} & 0 \\ 0 & 0 & 0 & \dot{\omega} & \left(\omega^2 - k\omega^{\frac{3}{2}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & -k\omega^{\frac{3}{2}} \\ 1 & 0 & 0 & 0 & -2\omega & 0 \\ 0 & 1 & 0 & 2\omega & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

Using the notation of Fig. 3.1 for coefficients in $\boldsymbol{\Xi}$ and comparing it with eq. 3.22, we can write a family of equations as:

$$\begin{aligned} \boldsymbol{\xi}_{1,4} &= \omega^2 + 2k\omega^{\frac{3}{2}} \\ \boldsymbol{\xi}_{1,5} &= -\dot{\omega} \\ \boldsymbol{\xi}_{2,4} &= \dot{\omega} \\ \boldsymbol{\xi}_{2,5} &= \omega^2 - k\omega^{\frac{3}{2}} \\ \boldsymbol{\xi}_{3,6} &= -k\omega^{\frac{3}{2}} \\ \boldsymbol{\xi}_{4,5} &= -2\omega \\ \boldsymbol{\xi}_{5,4} &= 2\omega \\ \boldsymbol{\xi}_{4,1} = \boldsymbol{\xi}_{5,2} = \boldsymbol{\xi}_{6,3} = \boldsymbol{\xi}_{7,4} = \boldsymbol{\xi}_{8,5} = \boldsymbol{\xi}_{9,6} &= 1 \end{aligned} \tag{3.23}$$

We can notice a few relationships among the coefficients ξ which can be converted into constraints for the iterative regression. The relationships can be formulated mathematically as:

$$\begin{aligned} \boldsymbol{\xi}_{1,4} - \boldsymbol{\xi}_{2,5} + 3\boldsymbol{\xi}_{3,6} &= 0 \\ \boldsymbol{\xi}_{1,5} + \boldsymbol{\xi}_{2,4} &= 0 \\ \boldsymbol{\xi}_{4,5} + \boldsymbol{\xi}_{5,4} &= 0 \\ \boldsymbol{\xi}_{4,1} = \boldsymbol{\xi}_{5,2} = \boldsymbol{\xi}_{6,3} = \boldsymbol{\xi}_{7,4} = \boldsymbol{\xi}_{8,5} = \boldsymbol{\xi}_{9,6} &= 1 \end{aligned} \tag{3.24}$$

Typically, affine equality constraints are imposed as a simultaneous linear equation to be solved along with the objective function. It is usually represented as:

$$\mathbf{C}_i \boldsymbol{\Xi} = \mathbf{d}_i \tag{3.25}$$

where $\boldsymbol{C}_i$ denotes a matrix of coefficients that are formulated from a single relationship equation shown in eq. 3.24 and $\boldsymbol{d}_i$ denote the scalar term on the right hand side of those equations. Which converts the regression problem shown in eq. 3.21, into simultaneous equations shown below:

$$\left.\begin{aligned} \min_{\boldsymbol{\Xi}} \frac{1}{2} \left\| \dot{\mathbf{x}}^T - \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_2 + [\boldsymbol{\lambda}] \left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_1 \\ \mathbf{C}_i \boldsymbol{\Xi} = \mathbf{d}_i \end{aligned}\right\} \tag{3.26}$$

Mathematically, using matrix algebra, we can rewrite the relationships shown in eq. 3.24 into a matrix notation borrowed from eq. 3.25. For each relationship, we can write a single equation in the denoted form. Since we have four relationships, re-formulation would result in four matrix equations which mathematically denote our model discovery constraints. The eq. 3.26 can be rewritten as:

$$\left.\begin{aligned} \min_{\boldsymbol{\Xi}} \frac{1}{2}\left\|\dot{\mathbf{x}}^T - \begin{bmatrix}\mathbf{x}^T & \mathbf{a}_f^T\end{bmatrix}\begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}_I^T\end{bmatrix}\right\|_2 + [\boldsymbol{\lambda}]\left\|\begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}_I^T\end{bmatrix}\right\|_1 \\ \mathbf{C}_1\Xi = \mathbf{d}_1 \\ \mathbf{C}_2\Xi = \mathbf{d}_2 \\ \mathbf{C}_3\Xi = \mathbf{d}_3 \\ \mathbf{C}_4\Xi = \mathbf{d}_4 \end{aligned}\right\} \tag{3.27}$$

By vectorising the matrix equation denoting the relationships between the coefficients in Ξ, also known as constraint matrix, we can write eq. 3.27 as:

$$\left.\begin{aligned} \min_{\boldsymbol{\Xi}} \frac{1}{2}\left\|\dot{\mathbf{x}}^T - \begin{bmatrix}\mathbf{x}^T & \mathbf{a}_f^T\end{bmatrix}\begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}_I^T\end{bmatrix}\right\|_2 + [\boldsymbol{\lambda}]\left\|\begin{bmatrix}\mathbf{A}^T\\ \mathbf{B}_I^T\end{bmatrix}\right\|_1 \\ \mathbf{C}_{eq}\Xi_{eq} = \mathbf{d}_{eq} \end{aligned}\right\} \tag{3.28}$$

where $\boldsymbol{C}_{eq}$ and $\boldsymbol{d}_{eq}$ are vectorised equality constraint matrices comprising of $\mathbf{C}_i$ and $\mathbf{d}_i$ respectively. The sizes of these matrices are as follows:

- $\mathbf{C}_{eq}$: $[N_c, N_f \times N_t]$
- $\boldsymbol{\Xi}_{eq}$: $[N_f \times N_t, N_t]$
- $\mathbf{d}_{eq}$: $[N_c, N_t]$

where N_t, N_f and N_c represent the number of targets in $\dot{\mathbf{X}}$, number of features in $\mathbf{X}$ and number of constraints in eq. 3.24 respectively. The $\boldsymbol{\Xi}_{eq}$ is generated by vertically appending $\boldsymbol{\Xi}$ to itself N_t times.

From the notation, we know that $\xi_{i,j}$ corresponds to the coefficient of the contribution of i^{th} feature in $\mathbf{X}$ towards the j^{th} target in $\dot{\mathbf{X}}$. The general equation for populating the elements of the constraint matrix $\mathbf{C}_{eq}$ can be written as:

Suppose we need to impose the m^{th} constraint $h \times \xi_{i,j} = k$:

$$\mathbf{C}_{eq\ [j,\ (i+m\times N_f)]} = h;\ \mathbf{d}_{eq\ [m,1]} = k \tag{3.29}$$

For example, imposing the first constraint according eq. 3.24 would result in such formulation:

We know the first constraint as:

$$\boldsymbol{\xi}_{1,4} - \boldsymbol{\xi}_{2,5} + 3\boldsymbol{\xi}_{3,6} = 0 \tag{3.30}$$

This constraint corresponds to three unique coefficients in $\boldsymbol{\Xi}$, hence we must populate three elements in $\mathbf{C}_{eq}$ and since it is only one constraint, one element in $\mathbf{d}_{eq}$. Using the general formula from eq. 3.29, we can populate constraint matrices as:

$$\begin{cases} \mathbf{C}_{eq\ [4,\ (1+1\times 6)]} = 1 \\ \mathbf{C}_{eq\ [5,\ (2+1\times 6)]} = -1 \\ \mathbf{C}_{eq\ [6,\ (3+1\times 6)]} = 3 \\ \mathbf{d}_{eq\ [1,1]} = 0 \end{cases} \tag{3.31}$$

The regression problem which is developed until now and formulated in eq. 3.28, is complete and can be solved using an iterative regression solver.

However, it should be noted that while we have only imposed four constraints, this is not sufficient to reduce the **DoF**s of the model discovery regression to zero. This would mean that the model discovery would result in a coefficient matrix $\boldsymbol{\Xi}$ which is over-identified according to Fig. 3.4. To compensate for this, a strategy to implement an additional fifth constraint is developed in the next chapter.

3.3.6. Optimiser

The standard **STLSQ** regression optimiser penalises the model fitting error $\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})$ and the sparsity of the discovered coefficients matrix elements in $\boldsymbol{\Xi}$. The hyper-parameter λ controlling the sparsity of $\boldsymbol{\Xi}$ can be tweaked to obtain desired levels of sparsity in the discovered models. Higher values of λ penalise the regularisation of $\boldsymbol{\Xi}$, which results in a highly sparse $\boldsymbol{\Xi}$ matrix after the regression problem shown in eq. 3.28 converges and vice-versa.

Since the fundamental philosophy is sparsity promoting regularised model discovery regression, we can utilise a novel objective function called the **SR3** proposed in Ref. [48]. The **SR3** algorithm proposes an objective function which generates more robust discovered models of the system. This is done by introducing an additional term in the regression objective function which penalises the variation in the coefficient matrix over each iteration. While the **STLSQ** objective function promotes sparsity, **SR3** promotes both sparsity, correlation and robustness of the discovered model.

The **SR3** objective function can be mathematically formulated similarly to the **STLSQ** objective function in eq. 3.19 as:

$$\min_{\boldsymbol{\Xi}} \frac{1}{2} \parallel \dot{\mathbf{X}} - \boldsymbol{\Theta}\left(\mathbf{X}\right) \boldsymbol{\Xi} \parallel_2 + \lambda ||\boldsymbol{\Xi}||_1 + \frac{1}{2\nu} \parallel \boldsymbol{\Xi} - \mathbf{W} \parallel_2 \tag{3.32}$$

where **W** represents a coefficient matrix similar to $\boldsymbol{\Xi}$, but it is generated through relaxation of the coefficients in $\boldsymbol{\Xi}$. This process is shown graphically in Fig. 3.2. The process of relaxation is done by elimination of coefficients which correspond to a weak correlation, during every iteration. The threshold for elimination can be tweaked pragmatically in the algorithm.

We can now assemble the combined regression equation for model discovery using the **SR3** optimiser with $L1$-norm sparsity promoting regulariser, coefficient dependent thresholding and coefficient constraints. By using eq. 3.28 and substituting it into eq. 3.32, we obtain the complete system of objective equations for the regression, which will be referred henceforth as **CSR3** algorithm:

$$\left.\begin{aligned} \min_{\boldsymbol{\Xi}} \frac{1}{2} \left\| \dot{\mathbf{x}}^T - \begin{bmatrix} \mathbf{x}^T & \mathbf{a}_f^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_2 + [\boldsymbol{\lambda}] \left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \right\|_1 + \frac{1}{2\nu} \left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} - \mathbf{W} \right\|_2 \\ \mathbf{W} = \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix} \odot \left\{ \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}_I^T \end{bmatrix}_{i,j} > \boldsymbol{\eta} \right\} \\ \mathbf{C}_{eq} \boldsymbol{\Xi}_{eq} = \mathbf{d}_{eq} \end{aligned}\right\} \tag{3.33}$$

where $\boldsymbol{\eta}$ represents the coefficient specific matrix of relaxation thresholds, $\odot$ represents an element-wise multiplication operation, the expression enclosed in curly braces is an element-wise conditional binary operator on the matrix.

Finally, the only undefined variable is $\boldsymbol{\eta}$. Since the actual values of the thresholds would depend on the coefficient values in $\boldsymbol{\Xi}$, which in-turn depend on the spatial and temporal locality of the chaser spacecraft. Hence, we need to dynamically generate these thresholds depending on the locality. The procedure to dynamically generate thresholds will be discussed in the following chapter.

3.4. Model Discovery Robustness

The regression based model discovery method detailed in the previous section is characterised by two main drawbacks. Since it is essentially a data based method, it relies on accuracy and quality of data measurements. According to the framework laid out for model discovery, the measurements that are used are essentially the classical state vectors for relative motion and mass. These measurements are sourced from on-board sensors which adds measurement noise and bias to them. This would hamper the model discovery method's efficacy in its trajectory prediction performance.

Another drawback is the stability of the discovered model. It is a well investigated problem that numerical regression methods can produce an accurate coefficient matrix which describes the dynamics. However, for implementing the discovered model as the predictor in the associated **MPC**, we need models that are not prone to stochastic instability. Slight variations in discovered coefficients can produce largely varying trajectory predictions when integrated over a long interval, even under the same constraints. Another outcome of this is degraded model discovery repeatability. With the same measurements and constraints the probability of obtaining the same discovered model while running the algorithm multiple times is very low.

These will form the basis of investigation for this section as well as methods to make the model discovery framework robust and resistant to degraded sensor measurement quality.

3.4.1. Data Processing

The performance of **CSR3** algorithm deteriorates in the presence of inaccurate measurement data. The primary foreseen application of autonomous **MDPC** controllers are foreseen in cubesats. Most cubesats performing tandem missions or proximity operations rely on **GNSS**, onboard **IMUs** and radio ranging to measure or estimate relative position and velocities. Within the scope of this thesis, it is assumed that the available measurements for model discovery are obtained from radio ranging.

The choice for radio ranging only is driven by the satisfactory performance and frugality of the systems, which is often the reason for their prevalence in cubesats. This ensures that the developed **MDPC** can be potentially used in rudimentary cubesats. The typical accuracy of radio ranging methods in cubesats are on the order of 10^{-2} $m, \frac{m}{s}$ [47] and measurement frequency in the order of $[10^4 - 10^3]$ Hz as demonstrated in 2010 in the **PRISMA** mission by Swedish National Space Board.

There is a lack of accurate relative acceleration measurement which is required for model discovery regression as targets in $\dot{\mathbf{X}}$ matrix. Hence, a numerical differentiation based estimation is implemented to pre-process the relative position and velocity measurements to estimate the relative acceleration. It is well known that numerical differentiation causes dynamics artefacts in the measurements and can be characterised by large errors based on the quality of the original measurements. To avoid this issue, smoothed high-order numerical finite differentiation methods are utilised.

The differentiation method used is the fourth-order moving-mean windowed finite difference scheme. The choice of the optimum window size depends on the measurement frequency and the bandwidth of the system dynamics. The chosen window size is 5. This ensures that there is no loss of dynamics information during differentiation of measurements. This is essentially a low-pass filter connected to a numerical differentiator.

3.4.2. Simultaneous Model Discovery

While the problem of measurement accuracy and quality was trivial to solve, the problem concerning the discovered coefficient stability and robustness is more complicated to solve. The term stochastic hardening refers to the process of making the model discovery framework more resistant to stochastic performance deterioration. The diagrammatic representation of the stochastic hardening is depicted in Fig. 3.5.

The model discovery process produces, as an output, the coefficient matrix Ξ which represents the system matrix $\boldsymbol{A}$. To understand the inherent variance in the model discovery coefficient matrix, we can apply the model discovery framework multiple times to generate multiple models representing the same underlying dynamics.

This is performed by sub-sampling the sensor measurement data-set into multiple subsets denoted in the Fig. 3.5 by $\dot{\mathbf{x}}_n$ and $\mathbf{x}_n$, where n denotes the number of sub-sampled data-sets. Then model discovery is performed on each data-set to generate n independent coefficient matrices Ξ_n. This process is called simultaneous model discovery.

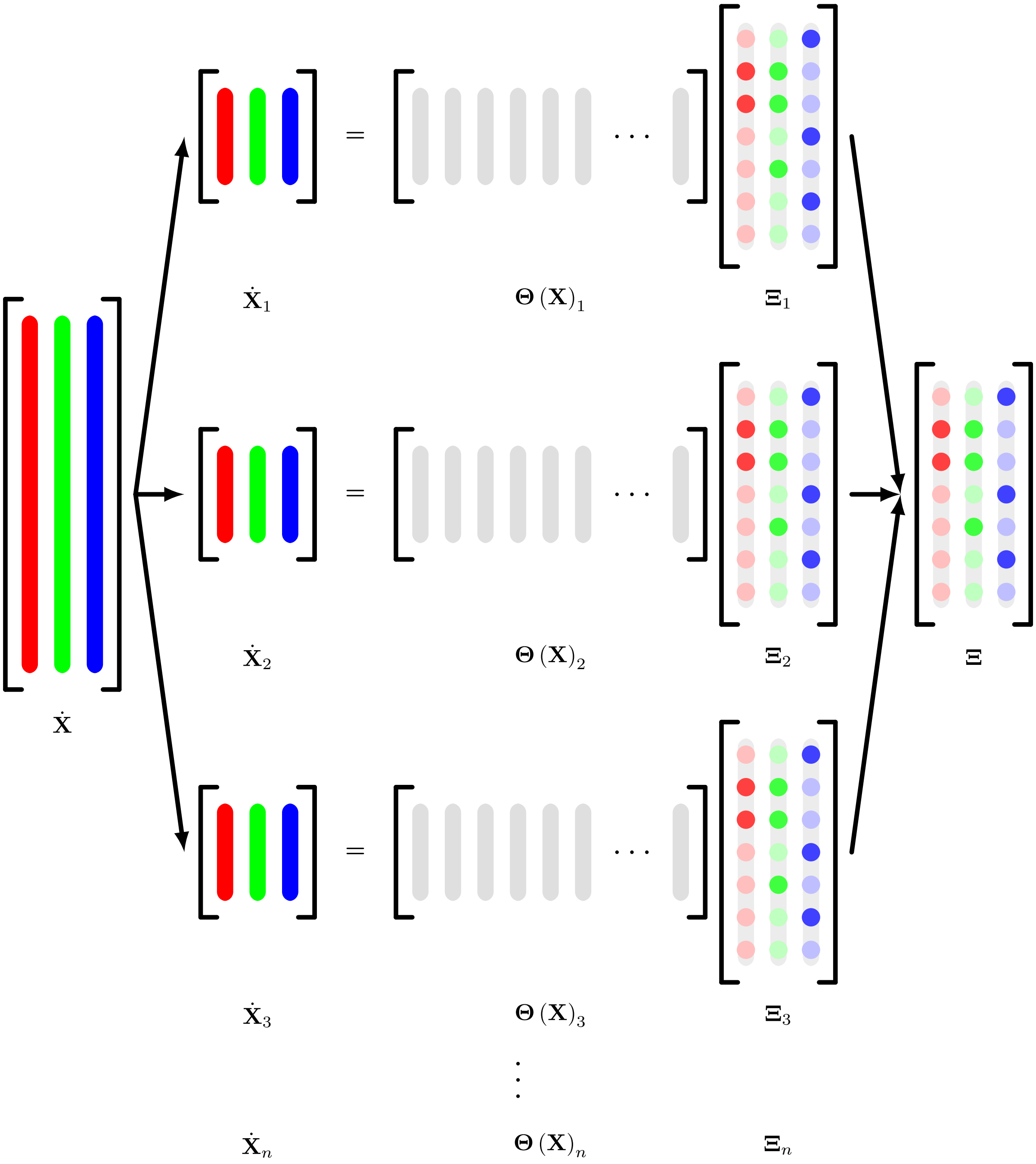


Figure 3.5: Simultaneous Model discovery framework.

The strategy used for sub-sampling the data-set is crucial in determining if the obtained coefficient matrices are genuinely correlated to the underlying physical dynamics. Since the actual coefficients of the system matrix are time and state dependent, the system model evolves with time. Hence, if sub-sampling of data is performed without care, the resulting coefficient matrices would not represent the same linearised models. They would correspond to slightly evolved system models, each linearised in a different interval.

Two different and simple sub-sampling strategies are analysed and one is chosen among it for the final implementation. The algorithms can be represented as follows:

- **Stacked sub-sampling**: This refers to sub-sampling of measurements into n different data-sets based on subdividing the original data-set into n consecutive equal chunks of measurements bagged into the corresponding data-set.
- **Distributed sub-sampling**: This refers to sub-sampling of measurements into n different data-sets based on distributing every n^{th} consecutive measurement being bagged into the corresponding data-set.

The stacked sub-sampling method results in n different models which are linearised in n consecutive intervals, whereas, distributed sub-sampling results in n different models linearised in the same interval but with each subset having reduced temporal resolution of measurements by a factor of n compared to stacked sub-sampling.

The choice of sub-sampling strategy is driven by the nature of sensor measurements. Since, the **SINDy** algorithm based model discovery method requires measurements with high temporal resolution, if the distributed sub-sampling results in subsets with adequate temporal resolution, it should be preferred over stacked sub-sampling. If this is not the case and not enough temporal resolution can be maintained by distributed sub-sampling, stacked sub-sampling can be utilised.

For the purpose of this thesis, a distributed sub-sapling strategy is utilised. This choice is driven by the assumption of adequately high sampling frequency of the sensors, as substantiated by Ref. [47].

3.4.3. Monte-Carlo Based Multiple Shooting

To understand the uncertainty propagation in the context of the simultaneously discovered models, multiple shooting schemes are utilised. In this scheme, all the n different models are propagated using an **IVP** formulation for the same interval as the original sensor measurements. The divergence of the models can be analysed by looking at the solutions of the **IVP** and comparing it with the "ground truth" of the sensor measurements.

A simulation of this method is performed assuming relevant initial conditions typical to an orbital relative motion problem and the results are shown in the Fig. 3.6 below:

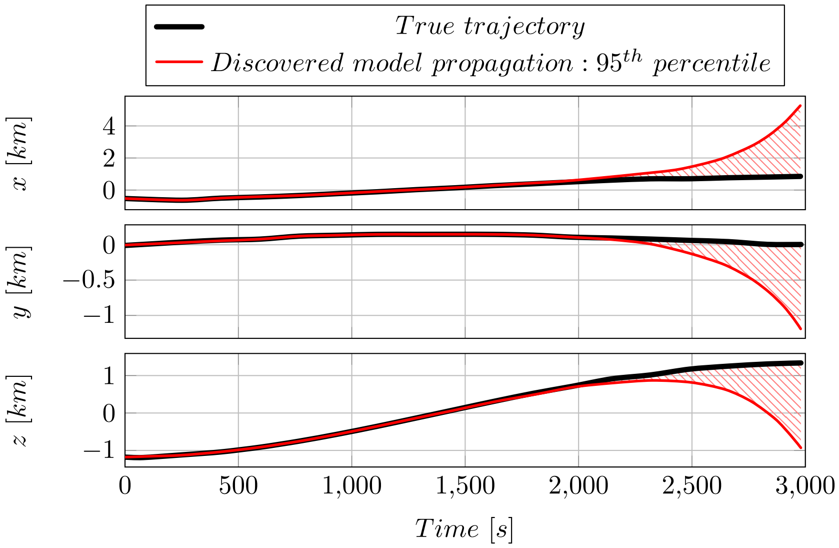


Figure 3.6: Monte-Carlo multiple shooting trajectories.

It can be observed from the Fig. 3.6, that the different models generated in the simultaneous model discovery scheme result in diverging trajectories when propagated in an **IVP** problem. By "shooting" several models forward in time, we now have enough data to analyse their accuracy and stability parameters, which can be exploited to formulate an effective model which is more robust.

3.4.4. Robust Model Assembly

The objective of this section is to formulate mathematical methods followed to improve the model discovery stability and robustness based on the Monte-Carlo analysis. While many statistical tools exist to perform this task, we require a simple method that provides adequately stable models with least complexity and computational loads. This is necessitated by the lack of on-board computational power in cubesats.

The proposed method is called the weighted averaging of coefficients based on the root of normalised mean-squared error or **RNMSE** weights. The mathematical formulation utilised to compute the trajectory propagation **RNMSE** is shown below:

$$E_{RNMSE,\ n} = \sqrt{\sum_{i=1}^{M} \frac{(\hat{\rho}_i - \rho_i)^2}{\rho_i^2}} \tag{3.34}$$

where $E_{RNMSE,\ n}$ denotes the propagation error, ρ and $\hat{\rho}$ denote the relative position "ground truth" and relative position from propagated models respectively. M denotes the total samples in the propagated trajectory and n denotes the total number of trajectories or essentially the total number of simultaneously discovered models.

With each simultaneously discovered model associated with its corresponding $E_{RNMSE,\ n}$, we can formulate the effective model which is assembled from all the n models. This is performed by averaging the coefficients of the Ξ_n matrices based on weights associated to their corresponding $E_{RNMSE,\ n}$. This can be mathematically denoted as:

$$\Xi_{eff} = \frac{\sum_{i=1}^{n} \left(\frac{1}{E_{RNMSE,\ i}}\right) \Xi_i}{\sum_{i=1}^{n} \left(\frac{1}{E_{RNMSE,\ i}}\right)} \tag{3.35}$$

where n denotes the total number of simultaneously discovered models. The inverse of **RNMSE** is used as weights to promote models which correspond to least **RNMSE**. The final output of the model discovery framework is the coefficient matrix Ξ_{eff}, which is a stable and robust model coefficient matrix discovered from sensor measurements. The resulting model performance is shown in the Fig. 3.7 below:

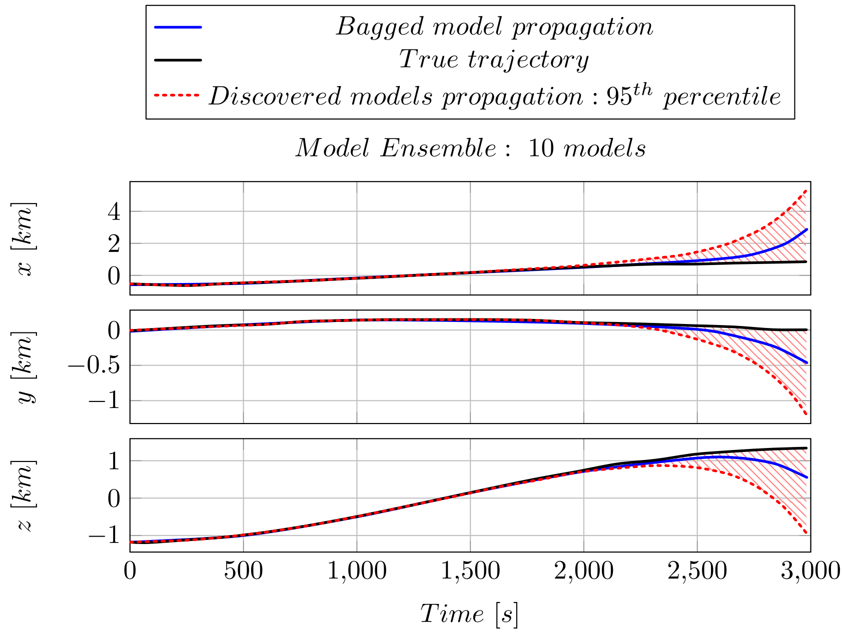
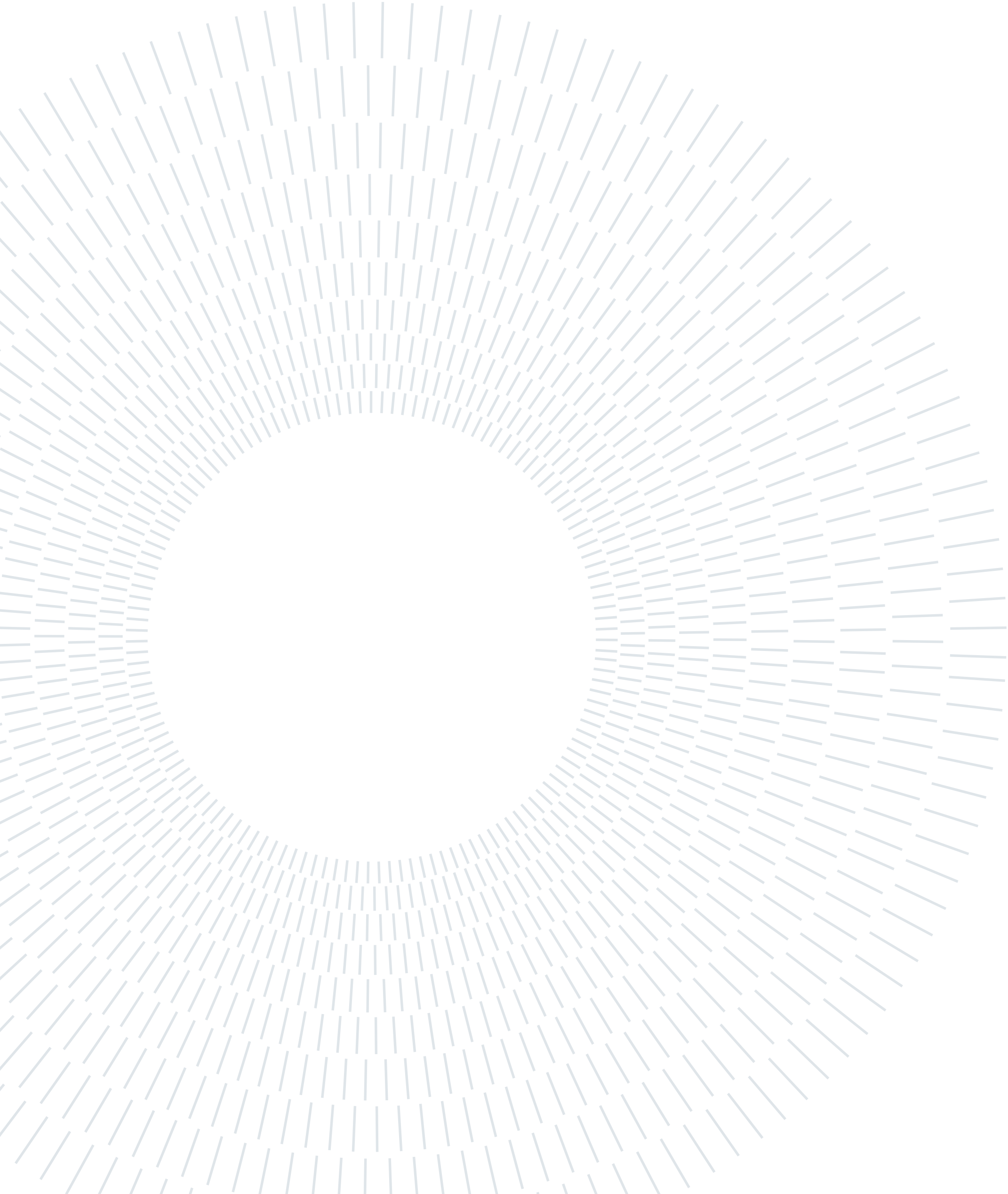


Figure 3.7: Assembled model performance.

It is evident that the robust model assembly method improves the model stability and robustness, as denoted by the corresponding trajectory. Although the assembled model still diverges, this is expected due to the effective linearisation performed on model discovery in the interval of measurement data-set. But this can be tackled by reducing the linearisation interval which would lead to a less diverging model. This will be elaborated in the next chapter.

As a final note, the yet untouched topic regarding coefficient specific thresholding matrix $\boldsymbol{\eta}$. This will be elaborated further in the next chapter.



4 | Prediction Performance Analysis

This chapter details the procedure undertaken to analyse the prediction performance of the model discovery framework laid-out in Chapter 3. The chosen methodology for analysis is comparison of "ground truth" relative trajectory of the chaser spacecraft and the predicted trajectory obtained from propagating the discovered model.

The following sections describe the testing campaign algorithmic layout, the simulation parameters and the initial conditions used for the purpose of this testing campaign along with the obtained results and the relevant inferences drawn.

4.1. Testing Campaign Layout

The testing procedure is initiated with generation of ground truth trajectory data and simulated sensor measurements of the features and targets relevant for model discovery. Then the data-set is partitioned into two consecutive chunks namely, discovery data-set and prediction data-set. The intention behind such splitting is to conduct linearised model discovery based on the first data-set and use the discovered model to generate future trajectory prediction and compare its performance with the "ground truth" prediction data-set.

The performance of the model discovery framework is measured in both the discovery and prediction data-set to draw insights into the interpolatory fitness performance in the discovery data-set and extrapolatory accuracy in the prediction data-set. It is crucial to understand both the performances because the interpolatory fitness represents how well the system has been understood based on the limited data-set. The extrapolatory accuracy indicates if the discovered model remains stable enough to maintain accuracy in future predictions outside the linearisation interval T_{lin}. The extrapolatory performance is of utmost importance in the **MPC** predictor module.

The implementation schematic is shown in Fig. 4.1 including the software and interfaces.

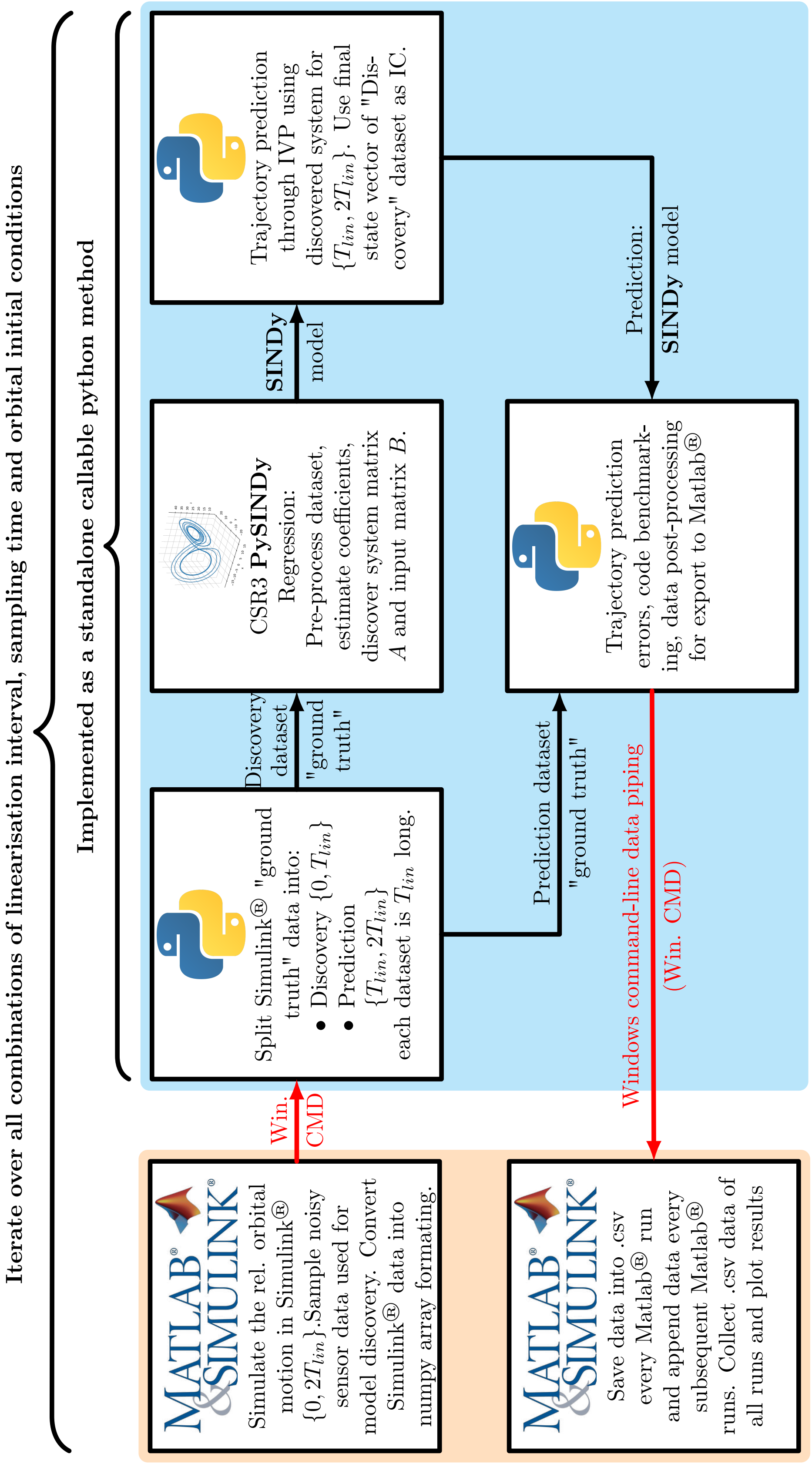


Figure 4.1: Testing campaign flowchart.

As represented in Fig. 4.1, the testing process begins with generation of "ground truth" data. The corresponding high-level block representation of "ground truth" simulation is shown in Fig. 4.2. It is performed by propagating the chaser and target spacecraft using Newtonian **2BP ODE** with J_2 perturbation contributions in the **ECEI** reference frame, see eq. B.1. Then the orbital states are rotated into the target spacecraft centred **LVLH** frame using a rotation matrix computed at each simulation step based on the target spacecraft orbital position in **ECEI** frame. Refer to Chapter 2 for detailed description of the **LVLH** frame and the **2BP ODE**.

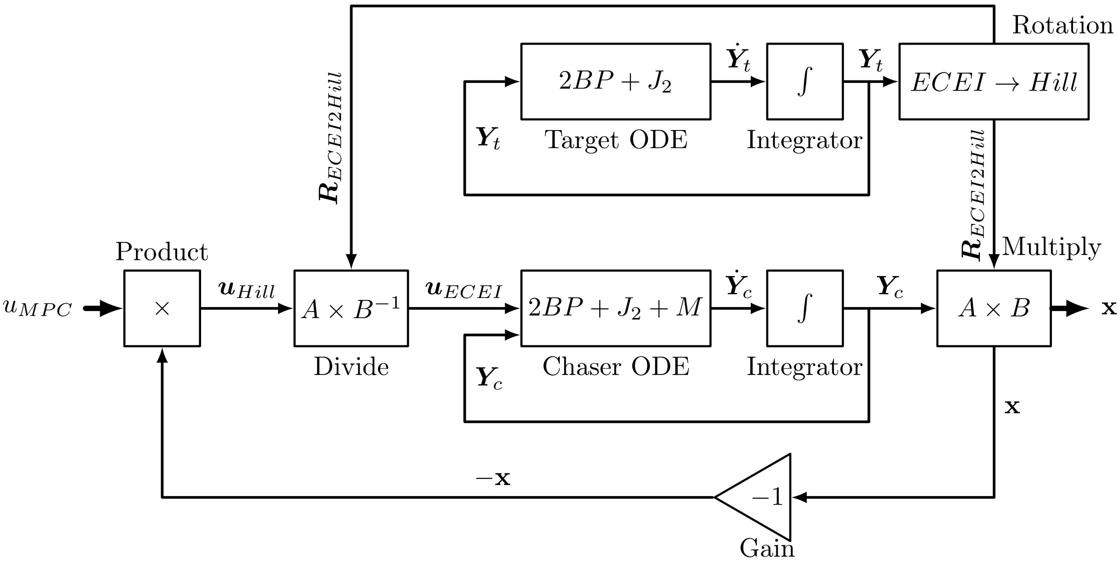


Figure 4.2: Ground truth simulation: Block representation.

Since the position vector of the target spacecraft in the targeted centred **LVLH** frame is always zero, the relative position of the chaser spacecraft in the same frame is obtained directly from the **ECEI-LVLH** rotation operation. The relative velocity of the chaser spacecraft is obtained by taking the vectorial difference of both the spacecrafts' velocities represented in the target centred **LVLH** frame. The mathematical equations utilised for the same are already introduced in Chapter 2 and omitted here for the sake of conciseness.

4.2. Spacecraft Actuator Configuration

Since, the primary focus of the thesis is to control the relative range of the chaser spacecraft with persistent target pointing, the attitude determination and control systems required to satisfy such pointing requirements is beyond the scope of this thesis.

The chaser spacecraft is assumed to be fitted with a pair of opposite facing thrusters/actuators which provide thrust along the longitudinal axis of the chaser spacecraft, aligned such that they always point towards and away from the target spacecraft.

Hence, the thrust unit vector is assumed as follows:

$$\hat{\mathbf{Q}} = -\frac{\boldsymbol{\rho}}{\rho} \tag{4.1}$$

where $\boldsymbol{\rho}$ represents the relative position vector of the chaser spacecraft in the target centred **LVLH** frame.

For the purposes of simplicity, the thrust applied during the "ground truth" simulation by the chaser spacecraft was modelled as a thrust vector pointing at the target spacecraft with a sinusoidal magnitude. The maximum limit of the thrust magnitude was set to be $1N$. This can be represented as:

$$\mathbf{Q} = -Q_{max} \sin\left(\omega_Q t\right) \left[\frac{\boldsymbol{\rho}}{\rho}\right] \implies \{Q_{max} = 1N;\ \omega_Q = 4\omega_{orb}\} \tag{4.2}$$

where ω_Q represents the frequency of the sinusoidal thrust input and ω_{orb} denotes the frequency of the orbit based on the **SMA** of the target spacecraft. The t denotes the simulation time of the "ground truth" simulation.

Finally, the target spacecraft is assumed to have no thrusters and hence is governed by the homogeneous solution to the **2BP** and J_2 perturbations.

4.3. Sensor Modelling

The chosen baseline to model the sensor was chosen to be the radio ranging sensors validated in the **PRISMA** mission. The sensor model is characterised by sensor noise and bias. The simplistic sensor model is chosen to aid in the ease of modelling and it is sufficient for the needs of a preliminary investigation. The main objective is to study the performance of the model discovery framework without a state estimator, directly fed by the sensor measurements. The schematic diagram of the same is represented in Fig. 4.3, where $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the "ground truth" and sensor measured values.

Based on the typical specifications of the **PRISMA** sensor suite, the values of sensor bias, noise and sampling resolution and frequency have been chosen. The chosen values are reported in Table 4.1.

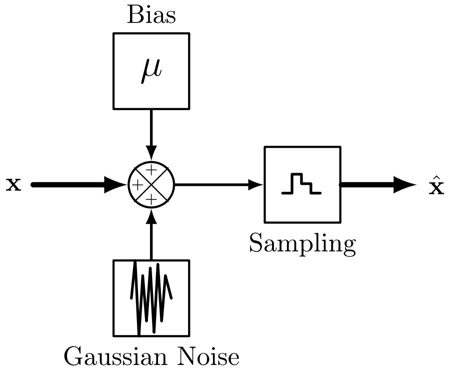


Figure 4.3: Sensor modelling schematic

Parameters	**Position**	**Velocity**	**Mass**
Bias: μ	$1.0\ e^{-2}\ m$	$1.0\ e^{-2}\ \frac{m}{s}$	$0\ kg$
Noise: σ	$1.0\ e^{-2}\ m$	$1.0\ e^{-2}\ \frac{m}{s}$	$1.0\ e^{-1}\ kg$
Sampling freq.	$1.0\ e^{3}$ - $10.0\ e^{3}\ Hz$	$1.0\ e^{3}$ - $10.0\ e^{3}\ Hz$	$1.0\ e^{3}$ - $10.0\ e^{3}\ Hz$
Discretisation	$1.0\ e^{-3}\ m$	$1.0\ e^{-3}\ \frac{m}{s}$	$1.0\ e^{-3}\ kg$

Table 4.1: Sensor modelling parameters

The parameter values chosen do not represent an actual sensor suite, however, are typical and adequate for simulating our problem. Modern sensor suites offer higher accuracy, resolution and sampling frequency, but a diminished performance case approach has been assumed for model discovery.

4.4. Prediction Performance Augmentation

In Chapter 3, the framework for model discovery was established and a method was proposed to increase the stability of the numerical regression solution. Different methods were discussed about imposing physical knowledge to drive the model discovery through constraining and thresholding. However, two things were left out to be discussed later namely, coefficient specific thresholding discussed in the context of eq. 3.33 and introduction of an additional constraint to make the model discovery framework have a null **DoF**. These topics will be addressed in this section along with development of a method to estimate a first preliminary guess of the system model coefficient matrix to initiate the model discovery iterative framework and to help augment its computational speed.

4.4.1. Model Initial Guess

Let us begin by recalling the expanded analytical matrix formulation of the Ξ matrix, from eq. 3.22. After imposing the constraints discussed in eq. 3.24, the only free **DoF**s of the Ξ matrix can be attributed to the unconstrained zero elements of Ξ matrix. Hence, if a method can be developed to provide a good first guess estimate for the coefficients corresponding to the constraints in eq. 3.24, we can formulate the complete first guess formulation of the Ξ matrix.

By observing eq. 3.24 and correlating each coefficient with their corresponding analytical expression from eq. 3.22, we can determine that there exists only three unique variables that formulate all these coefficients. These variables are namely, ω, $\dot{\omega}$ and k. We can express the analytical forms of these variables, borrowing from discussion preceding eq. 2.18 and utilising the notations of orbital mechanics models from Chapter 2:

$$k = \frac{\mu}{h^{\frac{3}{2}}};\; k\omega^{\frac{3}{2}} = \frac{\mu}{r_t^3};\; \dot{\omega} = \dot{\omega} \tag{4.3}$$

To obtain the estimates of these variables, the control system requires a complicated suite of sensors and estimation hardware or maintain a frequent contact with ground based control stations for updating these variables into the system. For the purpose of this thesis, these were deemed unacceptable because the former case is not typical of a cubesat while the later case reduces the degree of autonomy of the satellite control. Hence, to maintain a high degree of autonomy while also restricting the on-board hardware requirements, the following method was formulated.

Upon observing the various quantities that could be utilised to drive the estimation of these three variables, the following characteristics were necessary:

- The quantities used must be easily estimated with on-board sensors
- The reliance on ground-segment support must be minimal
- Frequency and latency of ground-segment support, if required, must be very infrequent and high-latency tolerant. This is typical for cubesats in deep-space missions where signal delays are considerably high. While also frequent ground-segment support drives the mission costs and increases the chances of failure in the case of severance of telemetry.

After narrowing down the possible solutions, the most acceptable method is presented henceforth.

Since our control system design already has good estimates and measurements of the relative motion, we can exploit the measurements to estimate some variables. Recalling eq. 3.9, we can isolate the equation corresponding to the local $z - axis$ as:

$$\ddot{z} = -k\omega^{\frac{3}{2}} z + a_{fz} + a_{pz} \tag{4.4}$$

which can be rearranged as:

$$k\omega^{\frac{3}{2}} = \frac{a_{fz} + a_{pz} - \ddot{z}}{z} \tag{4.5}$$

In this formulation, among the terms on the **R.H.S**, we already have estimations or measurements of a_{fz}, z and $\ddot{z}$. And the only unmeasured term a_{pz}, which is the relative perturbation acceleration in the **LVLH** frame in the direction perpendicular to the orbit, has already been established as a minimal value in prospective applications. Hence, by eliminating a_{pz} from the eq. 4.5 by assuming minimal contribution, we can rearrange it as:

$$k\omega^{\frac{3}{2}} \approx \frac{a_{fz} - \ddot{z}}{z} \tag{4.6}$$

Substituting the analytical variables with their estimated counterparts lead to:

$$\widehat{k\omega^{\frac{3}{2}}} \approx \frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}} \tag{4.7}$$

We have obtained the first variable estimate, we can utilise this to estimate the other two variables now. However, to estimate the other two variables, we cannot exploit the available measurement data. The choice was made to utilise a constant of orbit with very low variance to estimate the other two variables. The next estimation is driven by the assumption of knowledge of the magnitude of the specific orbital momentum of the spacecraft. However, to keep autonomy of the **MDPC** system high, the update frequency of the estimate of specific orbital momentum magnitude is limited to once per orbit, unless the original orbit is changed considerably during the interval.

Hence, we can proceed now with the second estimation with the added assumption of availability of $\hat{h}$ once per orbit. We can write:

$$\omega = \left(\frac{k\omega^{\frac{3}{2}}}{k} \right)^{\frac{2}{3}} \tag{4.8}$$

which can be combined with the substitution of k from eq. 4.3, resulting in:

$$\omega = \left(\frac{k\omega^{\frac{3}{2}}}{\frac{\mu}{h^{\frac{3}{2}}}} \right)^{\frac{2}{3}} = h \left(\frac{k\omega^{\frac{3}{2}}}{\mu} \right)^{\frac{2}{3}} \tag{4.9}$$

which can be written with their estimated counterparts leading to:

$$\hat{\omega} \approx \hat{h} \left(\frac{\widehat{k\omega^{\frac{3}{2}}}}{\mu} \right)^{\frac{2}{3}} \tag{4.10}$$

substituting eq. 4.7 and reformulating:

$$\hat{\omega} \approx \hat{h} \left(\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\mu \hat{z}} \right)^{\frac{2}{3}} \tag{4.11}$$

This gives us the estimated values for our second variable. The estimation of the third variable is not that straightforward due to the lack of measurements regarding the orbital state in the **ECEI** frame. However, since this estimation only functions as a first estimate for the regression algorithm, it is not required to be accurate. Rather, a higher accuracy of estimation would only help with the convergence speed of the model discovery algorithm.

We can estimate the third variable using the following method:

Using basic orbital mechanics, we can write:

$$\omega_c = \frac{h_c}{r_c^2} \tag{4.12}$$

where subscript $_c$ denotes parameters pertaining to the chaser spacecraft. By taking the derivative in time, we obtain:

$$\dot{\omega}_c = -2\frac{h_c}{r_c^3}\frac{d}{dt}r_c \tag{4.13}$$

Since, $\frac{d}{dt}(r_c)$ actually denotes the orbital velocity of the chaser spacecraft, we can use the general equations of orbital motion from Ref. [14] to write:

$$v_c = \sqrt{2\left(\frac{\mu}{r_c} - \frac{\mu}{2a_c}\right)} \tag{4.14}$$

where a_c represents the **SMA** of the chaser spacecraft orbit. By using basic substitutions, we can alternatively write the same equation using classical orbital mechanics notations as:

$$v_c = \frac{\mu}{h_c}\sqrt{1 + 2e_c \cos f_c + e_c^2} \tag{4.15}$$

we can now substitute this into eq. 4.13 and obtain:

$$\dot{\omega}_c = -2\frac{h_c}{r_c^3}\frac{\mu}{h_c}\sqrt{1 + 2e_c \cos f_c + e_c^2} = -2\frac{\mu}{r_c^3}\sqrt{1 + 2e_c \cos f_c + e_c^2} \tag{4.16}$$

Let us assume $\sqrt{1 + 2e_c \cos f_c + e_c^2} = \kappa_c$. Due to the lack of estimations of the eccentricity of orbit, we are forced to rely on the expected value of κ_c. This can be found using statistical methods to find expected values applied to the complete domain to the function.

The expected value of κ_c computed in the entire domain of κ_c, $f_c : \{0, 2\pi\};\ e_c : \{0, 1\}$, is given below:

$$\overline{\kappa}_c = \frac{3}{2} \tag{4.17}$$

by substituting this result into eq. 4.16, we obtain an expression for the expected value of $\dot{\omega}_c$ as:

$$\overline{\dot{\omega}}_c = -2\frac{\mu}{r_c^3}\frac{3}{2} = -3\frac{\mu}{r_c^3} \tag{4.18}$$

A final substitution is required to obtain the estimated expected value of the third variable in terms of the measurements.

We can assume $r_c \approx r_t$ hence, by substituting eq. 4.3 and eq. 4.6 into eq. 4.18, we obtain:

$$\hat{\overline{\dot{\omega}}}_c \approx -3\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}} \tag{4.19}$$

Summarising the first guess equations for the three variables, we get:

$$\widehat{k\omega^{\frac{3}{2}}} \approx \frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}};\; \hat{\omega} \approx \hat{h}\left(\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{2}{3}};\; \widehat{\dot{\omega}}_c \approx -3\frac{\widehat{a_{fz}} - \hat{\ddot{z}}}{\hat{z}} \tag{4.20}$$

We can collect these equations into the matrix form to write the final formulation of the initial guess for the Ξ matrix for the model discovery. By using the formulation of eq. 3.22, and substituting values from eq. 4.20, we can write:

$$\hat{\boldsymbol{\Xi}}_0 = \begin{bmatrix} 0 & 0 & 0 & \hat{h}^2\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{4}{3}} + 2\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\hat{z}}\right) & 3\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\hat{z}}\right) & 0 \\ 0 & 0 & 0 & -3\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\hat{z}}\right) & \hat{h}^2\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{4}{3}} - \left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\hat{z}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & \left(\frac{\hat{\ddot{z}}-\widehat{a_{fz}}}{\hat{z}}\right) \\ 1 & 0 & 0 & 0 & -2\hat{h}\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{2}{3}} & 0 \\ 0 & 1 & 0 & 2\hat{h}\left(\frac{\widehat{a_{fz}}-\hat{\ddot{z}}}{\mu\hat{z}}\right)^{\frac{2}{3}} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

where $\hat{\boldsymbol{\Xi}}_0$ denotes the estimated first guess of Ξ matrix to initialise the model discovery procedure.

4.4.2. Coefficient Specific Thresholding

We have finally developed all the prerequisite methods and formulations to tackle the problem of coefficient specific thresholding. The problem of coefficient specific thresholding arises due to the nature of the Ξ matrix which was introduced in eq. 3.33. The coefficients comprising the matrix typically range from $10^{-8} - 10^{0}$ in their orders of magnitude. Hence, a single scalar threshold cannot be used to eliminate the discovered regression coefficients in every iteration.

We can utilise the knowledge about the estimated first guess coefficient matrix $\hat{\boldsymbol{\Xi}}_0$ to inform the expected order of magnitude of the coefficients and formulate their corresponding thresholds. The shape of the coefficient specific thresholds matrix $\boldsymbol{\eta}$ is same as the Ξ matrix, with each element in Ξ mapped to its corresponding threshold in $\boldsymbol{\eta}$. The formulation adopted for the $\boldsymbol{\eta}$ matrix is as follows:

$$\hat{\boldsymbol{\Xi}} = \begin{bmatrix} 0 & 0 & 0 & \left(\omega^2 + 2\widehat{k\omega^{\frac{3}{2}}}\right) & -\hat{\omega} & 0 \\ 0 & 0 & 0 & \hat{\omega} & \left(\omega^2 - \widehat{k\omega^{\frac{3}{2}}}\right) & 0 \\ 0 & 0 & 0 & 0 & 0 & -\widehat{k\omega^{\frac{3}{2}}} \\ 1 & 0 & 0 & 0 & -2\hat{\omega} & 0 \\ 0 & 1 & 0 & 2\hat{\omega} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

By adopting the notation utilised in eq. 3.22, we can write the formulation of $\boldsymbol{\eta}$ as:

$$\boldsymbol{\eta} = \begin{bmatrix} \infty & \infty & \infty & \left(\omega^2 + 2\widehat{k\omega^{\frac{3}{2}}}\right)\mathcal{T} & -\hat{\omega}\mathcal{T} & \infty \\ \infty & \infty & \infty & \hat{\omega}\mathcal{T} & \left(\omega^2 - \widehat{k\omega^{\frac{3}{2}}}\right)\mathcal{T} & \infty \\ \infty & \infty & \infty & \infty & \infty & -\widehat{k\omega^{\frac{3}{2}}}\mathcal{T} \\ 0 & \infty & \infty & \infty & -2\hat{\omega}\mathcal{T} & \infty \\ \infty & 0 & \infty & 2\hat{\omega}\mathcal{T} & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix} \tag{4.23}$$

where $\mathcal{T}$ denotes the thresholding factor whose value can be changed to tune the relaxation in thresholding. If a very loose thresholding is required, the value of $\mathcal{T}$ can be very small, for tighter thresholding the value of $\mathcal{T}$ should approach 1. The exact value chosen for implementation is $\mathcal{T} = 10^{-2}$. This means that during iterations of the model discovery regression, any value which is less than one-hundredth of the estimated first guess value would be rejected. The thresholding value of ∞ represents the physical non-correlation and hence any non-zero value of that coefficient is inaccurate.

To avoid incompatibilities and clashing behaviours between thresholds and constraints, all coefficients that have been constrained with absolute equality constraints have their corresponding thresholds set to 0. Hence, the elements in $\boldsymbol{\eta}$ where a corresponding value of 1 is expected in $\hat{\Xi}$, has been thresholded to 0. This is because the actual value is regulated by the already imposed absolute constraints, discussed previously in subsection 3.3.5. On the contrary, the constraints imposed on other coefficients are not absolute, they are constraints imposed on the linear combination of multiple coefficients, hence we can also impose specific thresholds correspondingly, without constraint violations.

4.4.3. Final Additional Constraint

The discussions preceding this section have addressed mathematical models of relative motion, the model discovery framework modelling and optimisation. The goal of this subsection is to introduce an additional constraint on the model discovery model which, although not strictly required, improves the convergence performance as discussed in Section 3.3.5.

Since the model discovery framework utilises measurements and parameters which are scalar values, it lacks insight into the vectorial information regarding the orientation in the coordinate system. Examining the model discovery model, we can see that the coefficient matrix Ξ is computed numerically through regression. We have also imposed relative components to ensure that the discovered coefficients maintain the linear combination relationship which is necessitated by the physics. The coefficient dependent thresholding ensures fast model discovery and eliminates non-typical values during iterations. Still, there is a problem of an inversion of the discovered value of the ω in the Ξ matrix.

There exist specific combinations of chaser spacecraft states where ambiguous values of ω can be derived through solving the inverse problem. Hence, there are some situations in the relative motion where the value of ω can be substituted with its additive inverse $-\omega$ and still obtain the same sensor measurements.

To solve this, an additional constraint on the model discovery framework is introduced. This constraint fixes the values that can be assumed by the coefficient matrix elements corresponding to the 2ω. This can be denoted using the notation used in eq. 3.24:

$$|\boldsymbol{\xi}_{5,4} - 2\hat{\omega}| = \epsilon \tag{4.24}$$

where ϵ is the constraint tolerance. Unlike other absolute and relative constraints previously imposed, this is essentially a softer constraint due to the nature inequality constraints. This concludes the setup of the model discovery framework and associated optimisations. We can now proceed to discuss the prediction accuracy testing campaign.

4.5. Testing Campaign

The purpose of the model discovery framework is to deliver an accurate system model which can be used for predicting the chaser spacecraft trajectory and generate an optimised control sequence in the **MPC**. This can be performed only if the trajectory prediction of the model discovery framework provides reliable and accurate predictions. Hence, it is imperative to simulate the model discovery framework under varied scenarios and benchmark its performance.

The testing campaign is designed to cover a wide range of operating scenarios of the chaser spacecraft, although not all-encompassing, it is designed to establish the effect of measurement sampling frequency, relative motion non-linearity and initial conditions on the prediction performance.

4.5.1. Initial Conditions and Simulation Parameters

On closer inspection of the **NERM**, we can see that the degree of nonlinearity is a function of the orbit eccentricity e and the true anomaly f, both of which are used as simulation variables. The model discovery framework discovers the linearised model which is linearised in the interval of the measurement sampling used for the process. This means that the linearised models must be discovered multiple times during the orbit to ensure the accuracy of the prediction algorithm. Hence, the linearisation interval is also used as a simulation variable. Finally, the dependency of the model discovery framework on the measurement sampling frequency leads to the assumption of the sampling time as the final simulation variable.

However, the actual measurement sampling time is not used. This is due to the simultaneous model discovery algorithm dividing the actual measurement data-set into distributed data-sets. Hence, the effective sampling time of the distributed data-set $T_{s,\ eff}$ is used as the final simulation variable. The simulation output variable used as the figure of merit for the prediction performance is the trajectory prediction mean squared error compared to the "ground-truth" data-set.

Two different outputs are obtained, each corresponding to two types of performance tests. The first performance test is aimed at the interpolatory performance of the model discovery framework. It calculates the error of the discovered model trajectory propagation within the measurement interval used for the model discovery. Hence, it corresponds to how good of a fit does the discovered model offer compared to the actual physics. The second test tests the stability of the discovered model by extrapolating the trajectory propagation outside the measurement interval used for model discovery. Then it computes the extrapolated trajectory error compared with the "ground truth" measurements for the corresponding interval. An important note is that the extrapolation performance test only computes the error based on the "ground truth" data outside the linearisation interval. It does not use measurements outside the linearisation interval to "discover" the model. So essentially, it is a test to see if a model which is "discovered" based on past data can be used for accurate trajectory prediction in the future.

The initial conditions used to initialise every simulation consists of fixed parameters and simulation variables. The initial state vector for each spacecraft is generated based on the classic orbital Keplerian elements formulation in eq. 4.25, below:

$$\begin{aligned} Kep_T &= \begin{bmatrix} a_T & e_T & i_T & \Omega_T & \omega_T & f_{0T} \end{bmatrix} \\ Kep_C &= \begin{bmatrix} a_C & (e_T + \delta e) & (i_T + \delta i) & \Omega_C & \omega_C & f_{0C} \end{bmatrix} \end{aligned} \tag{4.25}$$

The fixed parameters used for each simulation is summarised below:

Parameters	**Value**
$\boldsymbol{a_T, \ a_C}$	$7106.14\ km$
$\boldsymbol{\delta e}$	0.001
$\boldsymbol{i_T}$	$98.3°$
$\boldsymbol{\delta i}$	$0.01°$
$\boldsymbol{\Omega_T, \ \Omega_C}$	$0°$
$\boldsymbol{\omega_T, \ \omega_C}$	$270°$
$\boldsymbol{T_s}$	$10^{-3}\ s$

Table 4.2: Test campaign: Fixed parameters.

where T_s denotes the sensor measurement sampling time. The variable parameters are cycled iteratively between every simulation to cover all combinations of variable parameters. They are summarised in Table 4.3:

The total number of simultaneous model discovery frameworks used for each simulation can be found using the formula:

$$N_{sm} = \frac{T_{s,\ eff}}{T_s} \tag{4.26}$$

Parameters	Range	Discretisation
$f_{0T},\ f_{0C}$	$[0\ -\ \pi]\ Rad$	5
$e_T,\ e_C$	$[0\ -\ 0.6]$	5
T_{lin}	$[10\ -\ 300]\ s$	5
$T_{s,\ eff}$	$[0.01\ -\ 1]\ s$	5

Table 4.3: Test campaign: Variable parameters.

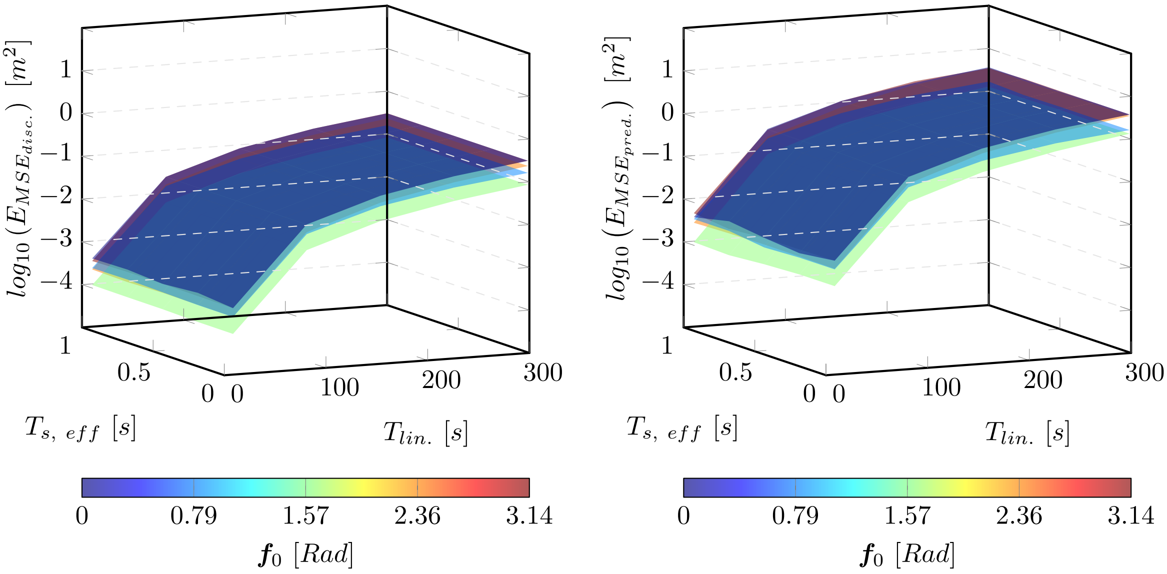
From table 4.3, we can see that a total of 625 unique combinations of variable parameters exist. And each combination is tested for interpolatory and extrapolatory performance.

4.5.2. Results

A total of 1250 simulations were run, encompassing all possible combinations of the variable parameters, simulating all possible scenarios that the relative motion can undergo which affect the **NERM**.

Testing Campaign Plots

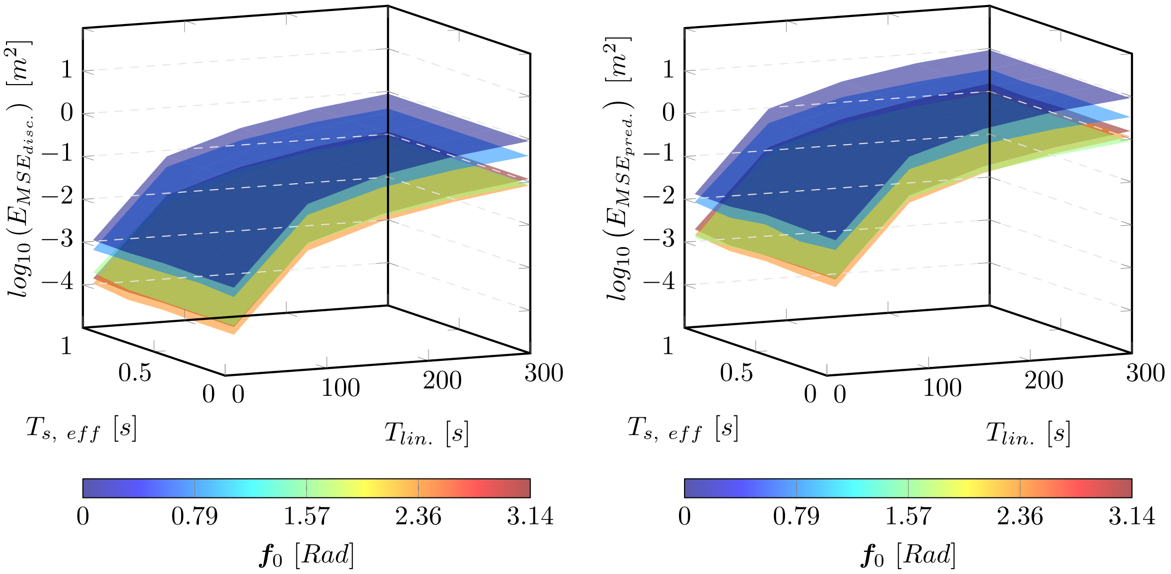
The 4^{th} dimension which is represented in different plots was chosen to be the orbital eccentricity. The results are shown in this section in a series of 3-dimensional plots.



(a) Model discovery error.

(b) Model prediction error.

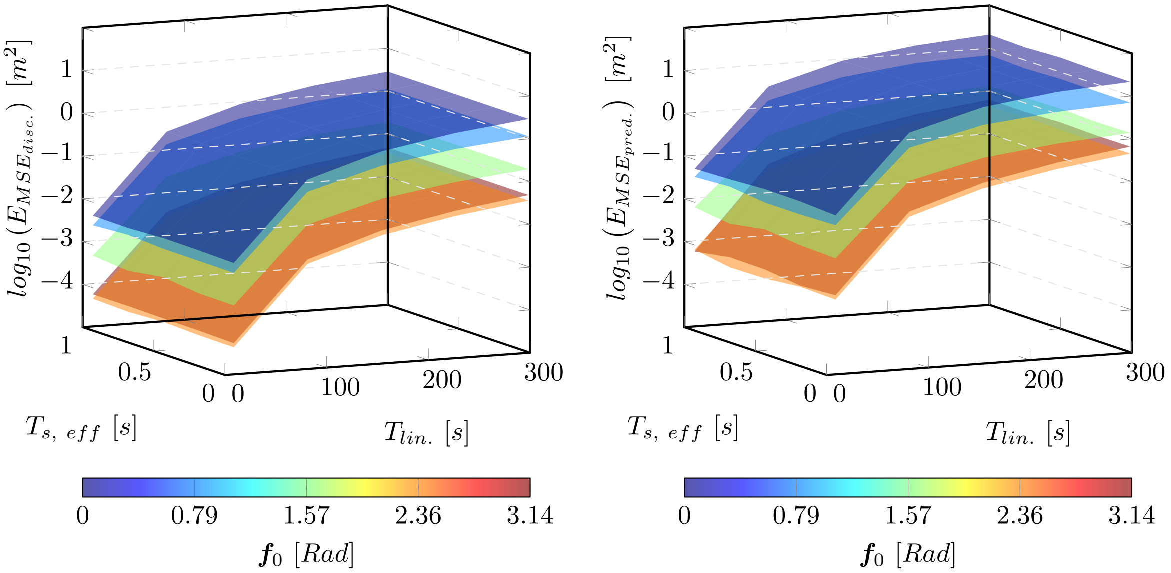
Figure 4.4: Test campaign results: $Eccentricity = 0.0$.



(a) Model discovery error.

(b) Model prediction error.

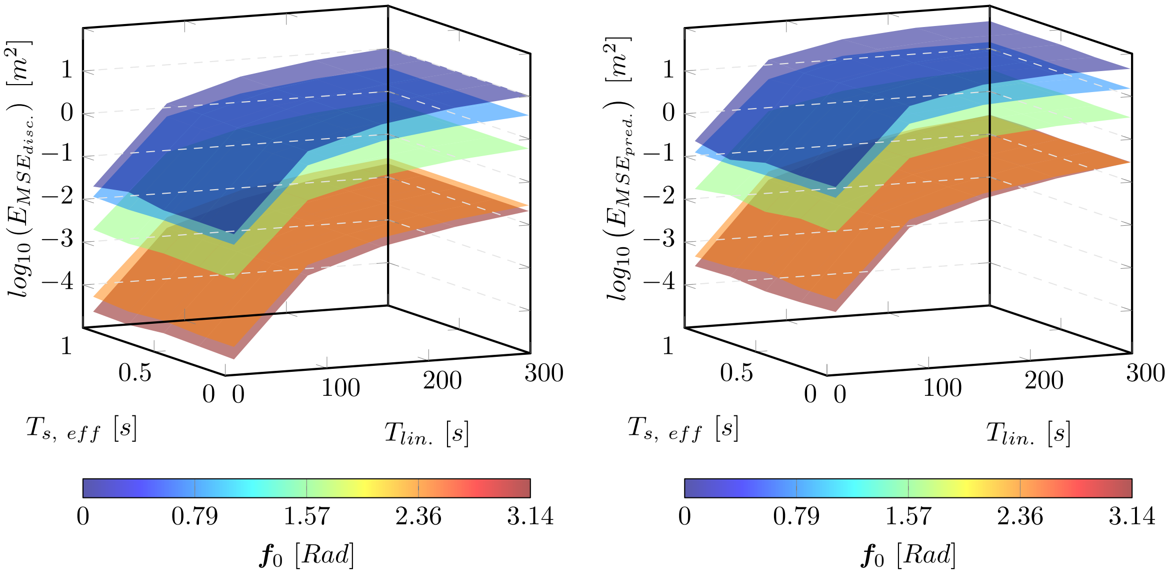
Figure 4.5: Test campaign results: $Eccentricity = 0.15$.



(a) Model discovery error.

(b) Model prediction error.

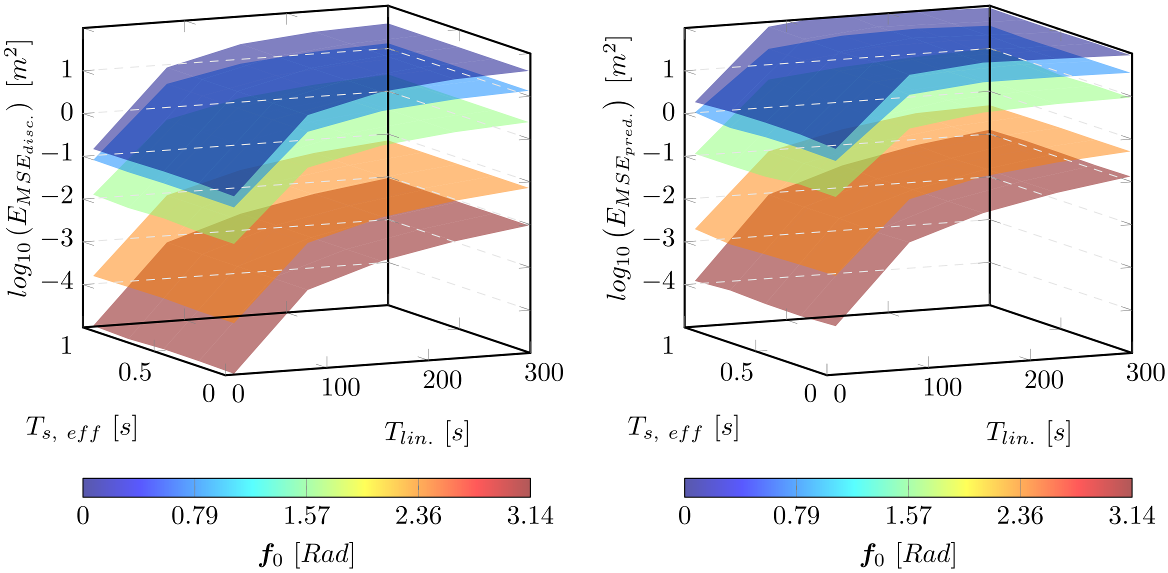
Figure 4.6: Test campaign results: $Eccentricity = 0.3$.



(a) Model discovery error.

(b) Model prediction error.

Figure 4.7: Test campaign results: $Eccentricity = 0.45$.



(a) Model discovery error.

(b) Model prediction error.

Figure 4.8: Test campaign results: $Eccentricity = 0.6$.

Testing Campaign Inferences

The main inferences drawn from the testing of the model discovery framework with all possible scenarios can be summarised in the following list:

1. The trajectory error in discovery $E_{MSE_{disc.}}$ is always smaller than the trajectory error in prediction $E_{MSE_{pred.}}$, for the same combination of simulation parameters;
2. The changes $T_{s,\ eff}$ do not seem to affect the error in trajectory prediction, which keeping the other parameters constant;
3. Increasing the linearisation interval T_{lin} leads to a faster increase in the trajectory prediction error. However, the rate of change of error seems to reduce considerably after a certain extent of increasing the linearisation interval T_{lin}.
4. The increase in the orbital eccentricity e results in an increase in both the errors.
5. The minimum recorded value of prediction error is $log_{10}\left(E_{MSE_{pred.}}\right) = -3.94\ m^2$, which corresponds to $E_{MSE_{pred.}} \approx 0.000115\ m^2$ with $e = 0$ and $T_{lin} = 10\ s$.
6. The maximum recorded value of prediction error is $log_{10}\left(E_{MSE_{pred.}}\right) = 1.98\ m^2$, which corresponds to $E_{MSE_{pred.}} \approx 95.5\ m^2$ with $e = 0.6$ and $T_{lin} = 300\ s$.

The first inference is expected due to the nature of linearisation. The extrapolatory error will always be greater than the interpolatory error due to the linearisation being done in the past and extrapolation being done outside the linearisation interval in the future. The second inference can be explained by understanding that the **NERM** dynamics have a typical characteristic period in the order of orbital time periods. The change in the sampling times are very marginal and do not approach the orbital period, hence, the dynamics information is not lost due to increasing sampling times.

Since the nature of **NERM** is non-linear, it is expected that the discovered model would diverge from the actual dynamics. By increasing the linearisation interval T_{lin}, we are forcing the model discovery framework to linearise the model when the measurements show higher degree of nonlinearity. Hence, it results in loss of information of the actual dynamics and results in poor performance, which is the reason for the third inference.

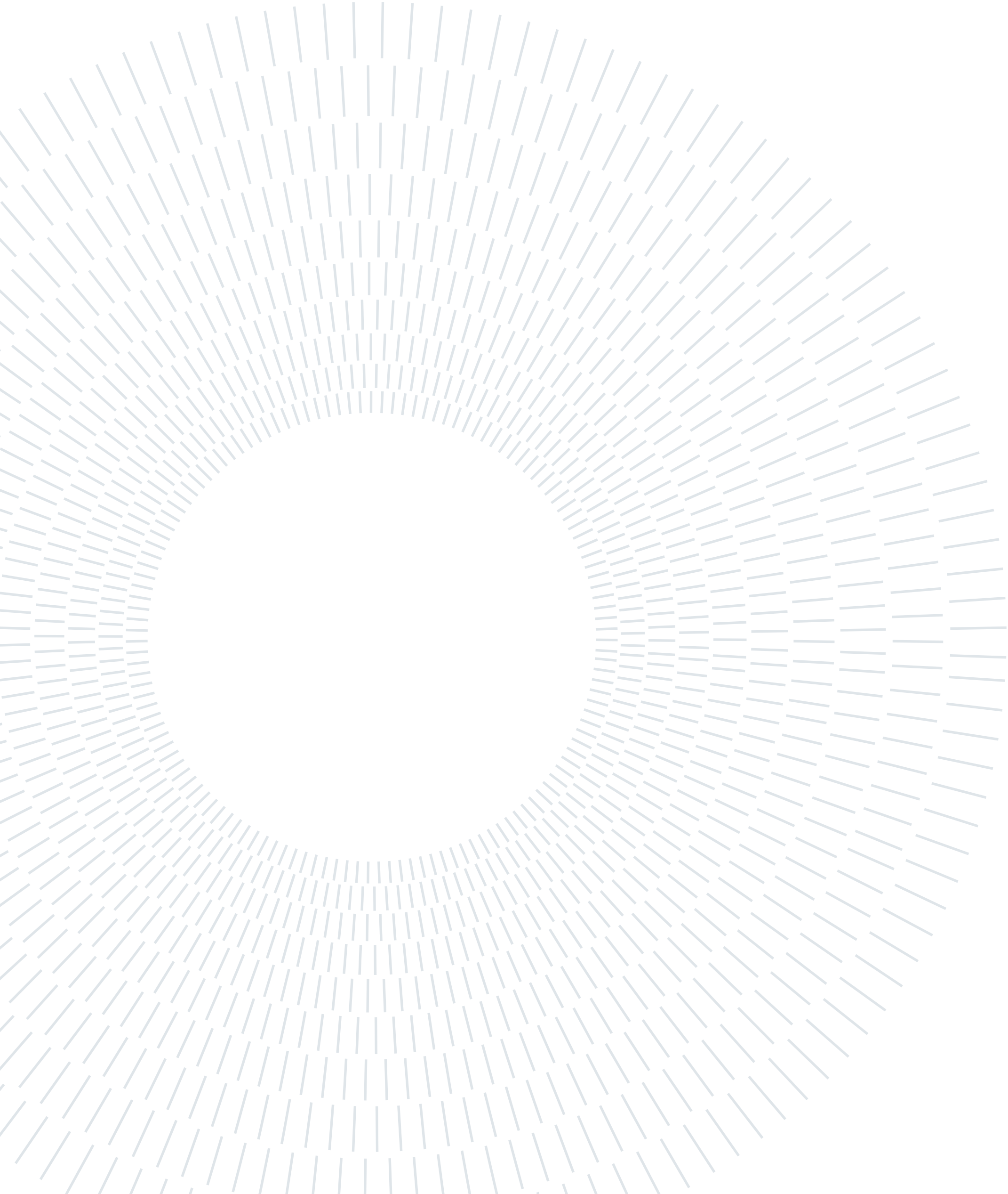
The fourth and the fifth inferences are helpful to decide if the developed model discovery framework can actually be implemented for **MPC** purposes. As it can be seen that the maximum mean-squared prediction error in trajectory is nearly 100 m^2, it is understood that this framework cannot be used in applications which require higher precision. This framework can be utilised for applications where the controller is expected to have an average accuracy in the order of 10 m. However, this is only in the worst-case scenario i.e. very high orbital eccentricity of $e = 0.6$, and linearisation interval $T_{lin} = 300\ s$. The model discovery framework has better accuracy even in a highly eccentric orbit with $e - 0.6$, if the linearisation interval is made smaller.

For example, with a linearisation interval of $T_{lin} \approx 85\ s$, the prediction error drops to the order of 1 m. However, the drawback of this approach is that the prediction horizon of the **MPC** is reduced. Hence a balance between the accuracy and prediction horizon is required in such scenarios.

Testing Campaign Conclusion

To decide what parameters would be optimal, a look-up table is generated with the results above. This can be used by the control system to dynamically change the parameters of the model prediction framework, based on the linear interpolation conducted on the look-up table with constraints on maximum error. This allows the control system to decide and find the best compromise between the acceptable error and the prediction horizon of the **MPC**.

Finally, it can be concluded that the model discovery framework offers a very good performance even in highly eccentric orbits. Hence, it can be a good algorithm to implement as the prediction algorithm in the **MPC**. The integration of this framework with the **MPC** is discussed in the next chapter, along with the associated design and implementation.



5 | Control System Implementation

The focus of this chapter is to introduce the concept of a **MPC**, discuss the various mathematical models used to implement the controller and finally the parameters used to tune the **MPC**. The chapter starts with the history and the general outline of a **MPC** and slowly builds on it to develop the mathematical models.

5.1. Introduction

Historically, **MPC** is a relatively new technology, only being developed in 1976. While primal implementations of this scheme utilised linear impulse responses and approximate piecewise functions for plant models for forecasting, modern **MPC**s are able to utilise full nonlinear state-space matrix plant models along with constraints and bounds optimised using quadratic programming techniques.

MPC are also commonly called finite time moving horizon controllers. They are a novel method utilised in system controls and are considered as advanced implementations. The core principle of control using an **MPC** is to solve an optimal control problem within a set of constraints, boundary conditions, bounds and a finite time boundary, also called the horizon. A graphical representation of this problem can be found in Fig. 5.1.

The **MPC** utilises the state estimates and propagates them, treating them as **IC**. Then it finds the optimum sequence of control inputs in time that produce an optimal control trajectory based on a cost function or objective function, subject to constraints and bounds. Some implementations use a different horizon for control inputs and trajectory forecast. This can be used to tune the response speed of the **MPC**.

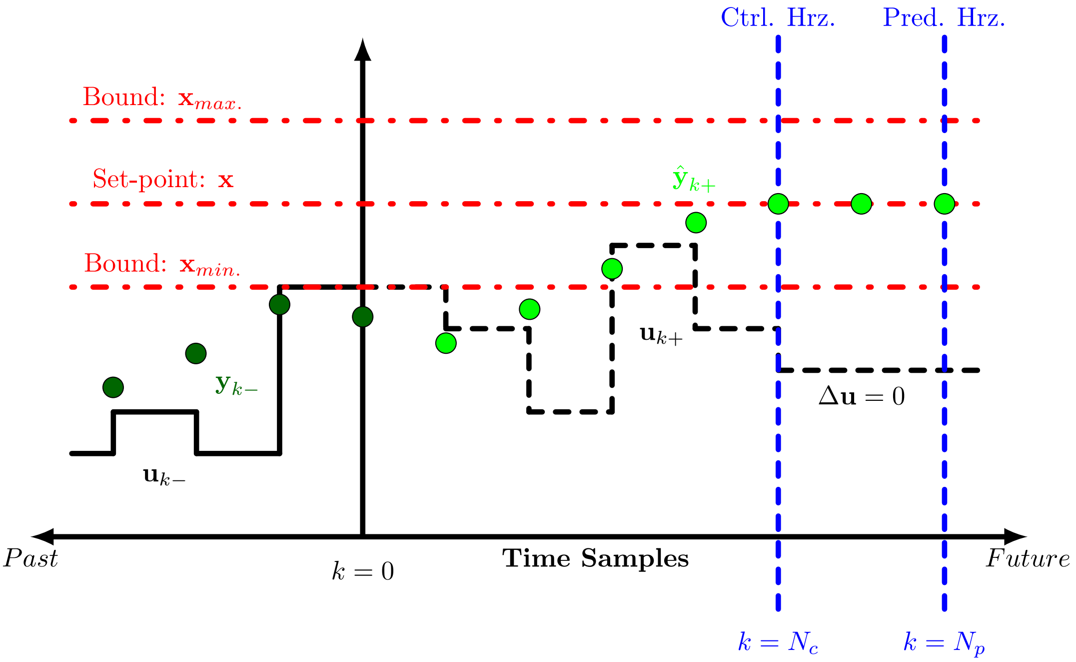


Figure 5.1: Model predictive controller principle.

While the **MPC** computes an optimal control sequence within the corresponding horizon, the actual control input implemented is the first input within the optimal control sequence. The ideal optimum sequence of control inputs is considered to reach a non-saturated steady-state control input at or before control horizon. This entire procedure completes one cycle of the **MPC**, which is repeated at every sampling instance.

5.2. Mathematical Modelling

Essentially, the mathematical formulation of a **MPC** is the same as an optimisation problem subject to constraints and bounds. However, to develop a customised and holistic mathematical model, it is prudent to discuss individual processes that are performed within an **MPC** and how to model them individually.

The block diagram representation of the **MPC** utilised for this thesis is shown in Fig. 5.2. The only external input to the implemented **MPC** is the current state estimate $\hat{\mathbf{x}}$, shown on the right. While the only output is the next sample for the control input $\boldsymbol{u}_{k=1}$, shown on the left.

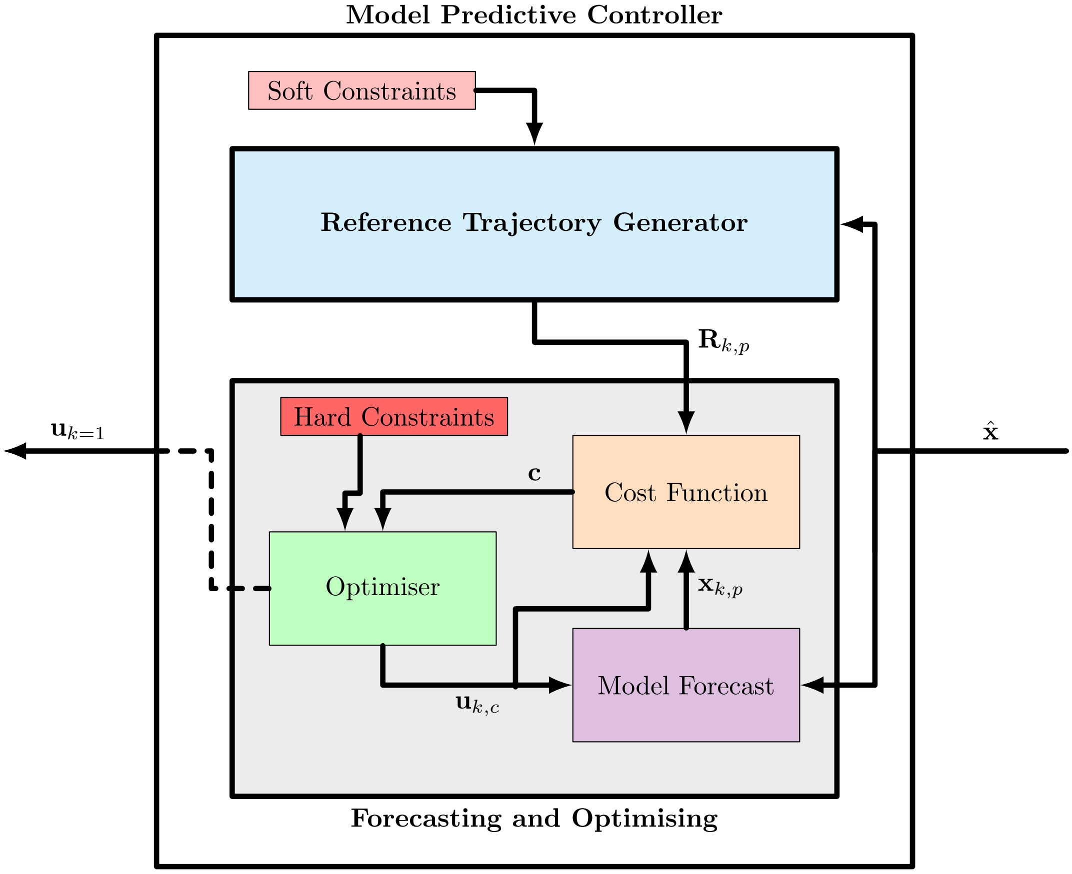


Figure 5.2: Model predictive controller basic schematic.

The current state estimate is used to generate a reference trajectory utilised as the guidance trajectory. Simultaneously, an iterative optimisation is conducted through a cycle of computing optimal control sequence, generating a trajectory forecast based on the control sequence and finally using a cost function to penalise any deviations of this trajectory prediction compared to the guidance trajectory. Once the optimisation converges, the first optimal control input in the optimal sequence, responsible for the optimal trajectory, is exported as the **MPC** output.

Throughout this optimisation, several "hard" and "soft: constraints are imposed on the optimisation problem. Usually, the range bounds are treated as "soft" constraints and the limits on the chaser spacecraft actuator thrust limits are considered as "hard" constraints. Finally, the optimiser algorithm used guarantees constraint non-violation at every iterative step. This is to ensure robust control and also to account for anomalies when the optimisation does not converge before the next sample instant and a premature control input has to be exported. The implemented optimiser ensures that even a premature export of the control input does not lead to constraint violation.

Let us now begin to methodically develop the mathematical models concerning each subsystem within the **MPC**, starting from the guidance algorithm or the reference trajectory generator.

5.2.1. MPC Guidance Algorithm

The primary goal of this thesis is to develop a **MPC** for bounding the chaser spacecraft trajectory within specified range limits with respect to the target spacecraft. Since there are no preferential or directional bounds, the trajectory limits are not individually or uniquely posed on each orthogonal basis. Rather, the trajectory is bounded by two concentric spherical surfaces centred on the target spacecraft, each having a radius corresponding to the minimum and maximum relative range limits.

Due to this particular requirement, there is no reference trajectory that guides the motion of the chaser spacecraft's motion. A simpler method was chosen to implement the relative range bounding as a relative range limit violation penalty in the optimisation problem. The penalty will be levied on any trajectory which leads to violation of the relative range limits. The mathematical formulation of the penalty will be discussed in the subsequent discussions.

Although, there is no optimal guidance algorithm, since the **MPC** already has provisions for a finite horizon optimisation, the optimal trajectory followed by the **MPC** will not be arbitrarily inefficient. One aspect of finite horizon optimisation is the lack of information on optimal trajectories beyond the horizon. This means that the **MPC** will follow a locally optimal solution in the context of a global optima computed with an infinite horizon optimisation.

To ensure that the finite horizon optimal solution closely resembles the infinite horizon solution, we must increase the prediction horizon to the **MPC**. However, due to the cyclical nature of the relative orbital motion, the optimum prediction horizon need not be arbitrarily large, it can be similar in order to the orbital period. The exact choice of the prediction horizon must also take into account the prediction model stability and accuracy degradation due to increasing prediction horizon. Further elaborate discussion regarding the trade-off is targeted later in this chapter.

5.2.2. Control Sequence Generation

The first step in the optimisation within the **MPC** is to generate the control input sequence that will be used within the forecasting algorithm to predict the effect of the application of the control sequence on the trajectory. In our case, the control inputs are the thruster actuation forces. Since there is only a single pair of thrusters in the chaser spacecraft, the thrust can be modelled with a single scalar value which can range within $[u_{min} \ - \ u_{max}]$. The chosen maximum thrust value for the thruster is 1 N.

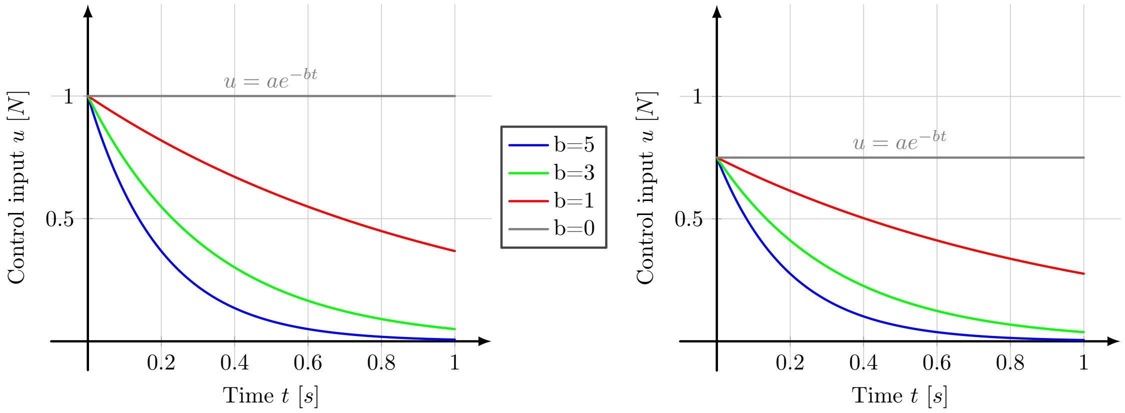
The thrust vector is always aligned pointing towards the target spacecraft, hence, we can simply use the magnitude of the thrust vector as an optimisation control variable. This is also due to the fact that the direction of the thrust cannot be controlled by the **MPC**, hence, the **MPC** must regulate the magnitude of the thrust to maintain an optimum trajectory. Hence the thrust scalar can take any values in the range $u \ \epsilon \ [-1 \ N \ - \ 1 \ N]$, where the negative values of thrust signify a thrust vector $\hat{\boldsymbol{u}}$ pointing in the opposite direction of the relative position vector $\boldsymbol{\rho}$.

Although the optimisation algorithm can be run with only limits on the minimum and maximum thrust, it is computationally inefficient. This is because the control input sequence will be generated independently and randomly for each sampling instance. This can be solved by using a parametric thrust function defined in the control horizon. The parametric thrust function is a mathematical description of the thrust profile shape which is a function of time and tuning parameters only. At the same time, it reduces the optimisation variables. Instead of optimisation control variables equal to the number of samples in the control horizon, the parametric equation needs only a few optimisation control variables equal to the number of parameters in the equation. This reduces the computational load, provides a provision to include physical limitations of the thrusters and also provides the control over what kind of behaviour is expected from the **MPC**.

There are many thrust parametric thrust profiles that can be chosen. Typical choices are impulsive profiles, continuous constant thrust profile, ramp thrust profile, logistical thrust profile, e.t.c. The particular thrust profile utilised for the purposes of this thesis is an exponential-decay parametric thrust profile. The mathematical expression of the exponential-decay parametric thrust profile is given in eq. 5.1, below:

$$u(t) = ae^{-bt}; \ \{0 \leq t \leq T_{CH}\} \tag{5.1}$$

where $u(t)$ denotes the magnitude of thrust as a function of time, a is the value of the initial thrust magnitude of the exponential curve which can take any value in $[u_{min} \ - \ u_{max}]$, b is the exponential decay factor of the thrust profile which can lie in the interval $[0 \leq b \leq 5]$ and finally, T_{CH} denotes the control horizon. The exact limiting value of $b_{max} = 5$ has been arbitrarily chosen because beyond that value, the thrust profile starts to resemble an impulsive thrust profile. Some example thrust profile curves are represented in Fig. 5.3 for representational purposes.



(a) Exponential Thrust Profile; $a = 1$.

(b) Exponential Thrust Profile; $a = 0.75$.

Figure 5.3: MPC thrust profile.

But since the **MPC** which is implemented, works in the discrete-time domain, we can write the eq. 5.1 in it's discrete-time notation as:

$$u_k = ae^{-bT_s k};\ \{0 \leq k \leq N_{CH}\} \tag{5.2}$$

where u_k represents the thrust magnitude at the k^{th} sample instance, T_s denotes the **MPC** sample time and finally, N_{CH} denotes the number of samples that comprises the control horizon.

To conclude, we must address the reasoning behind the choice of an exponential-decay parametric thrust profile. It is evident that introducing a thrust profile would deviate from the optimal control sequence. But it is also important to include physical considerations about thruster response times and performance into the thrust profile. The exponential decay curve also introduces a bias in the controller. Due to the inherent nature of the curve, more control impulse is imparted in the beginning of the control horizon and slowly the control effort is reduced. This means that the optimiser will now provide optimal control sequences which prioritise imparting most of the required impulse at the beginning of the control action. This also helps in reducing the response time of the **MPC**.

5.2.3. Trajectory Forecast

Once the optimisation iteration is initiated and a control sequence is generated using the parametric thrust profile, we must utilise the control sequence to forecast the trajectory of the chaser spacecraft. This is done by imposing the current state estimate $\hat{\mathbf{x}}$ as the **IC**, the generated control sequence as the **BC**, and propagating them using the dynamics established by the model discovery framework for the prediction horizon T_{PH}, using the sample time T_s as the integration time step.

An important aspect of the forecasting is that the model generated by the model discovery framework has been linearised in the past. Hence, the forecasting or prediction horizon T_{PH} is limited to the linearisation interval T_{lin} of the model discovery framework. This ensures good model stability and accuracy of prediction which has been discussed in the previous chapter. This process of utilising a past linearised model in the current instance trajectory forecasting of the **MPC** is shown in Fig. 5.4 below:

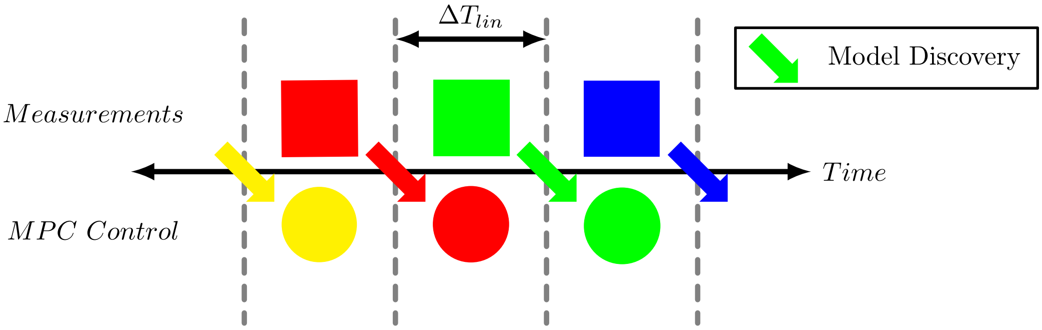


Figure 5.4: Linearisation and MPC forecasting scheme.

Until the current cycle of the linearisation interval expires, the discovered model is not updated. The same model is utilised over multiple instances of the optimisation cycle of the **MPC**. The exact number of times it is reused can be calculated by using the formula $N_{MR} = \frac{T_{lin}}{T_s}$ where, N_{MR} denotes the number of times the discovered model is re-used before updating through a new model discovery cycle initiation.

The accuracy required by the integration scheme utilised by the forecasting algorithm is determined by the order of accuracy of the discovered model in prediction. The best order of accuracy attainable by model discovery in prediction is $\approx 10^{-3}\ m$. Hence, the numerical integration scheme for trajectory forecasting is the **RK4** scheme.

The trajectory bounds are defined in relative range ρ, not the relative position vector $\boldsymbol{\rho}$. Hence, the trajectory forecasting outputs the forecasts of relative ranges ρ_k for the prediction horizon T_{PH}.

5.2.4. Cost Function Formulation

The **MPC** framework is designed to find an optimal control input sequence which is feasible, does not violate the essential constraints and which tries to keep the relative-range within the requested bounds. To understand how this can be translated into an optimisation problem, we must understand the optimisation space. In the case of the **MPC** being designed for the purpose of this thesis, we can represent the optimisation space as represented in the Fig. 5.5

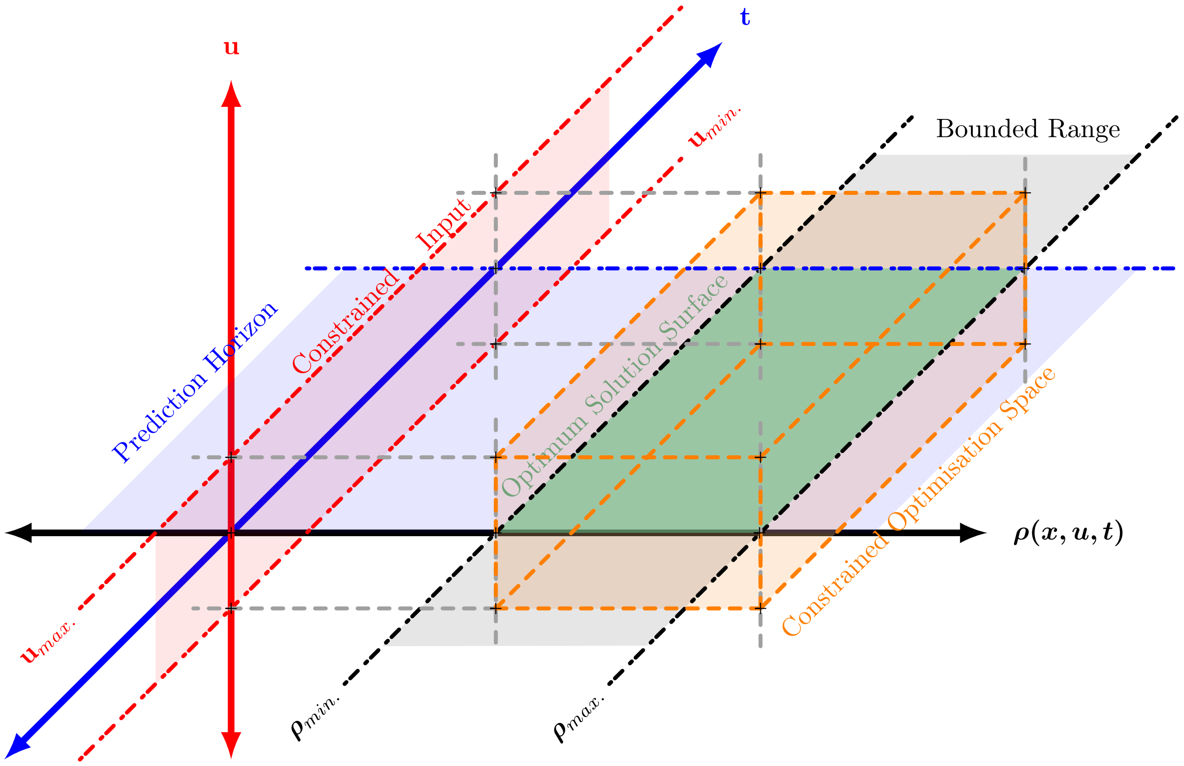


Figure 5.5: MPC optimisation space.

The optimisation space is an imaginary function space where the control optimisation problem is defined. In our case, the three coordinate directions are defined by the control input u, the relative range ρ and time t. Each of these axes is either constrained or finitely bounded, the enclosed space which is defined by the intersections of all the regions which are unconstrained or unbounded forms the constrained optimisation space. This is the region in the imaginary hyper-space where the **MPC** optimisation problem is allowed to exist. The only exception is given to violation of bounded surfaces defined in the figure as relative-range bounds. The optimisation problem can violate the limitations of these surfaces.

The optimum solution of the **MPC** problem exists as a surface in the figure, which represents all possible solutions where there is no control input and the relative-range is within the bounds and defined within the prediction horizon. The goal of our **MPC** optimiser is to find a locus in this hyperspace connecting the initial point in this space lying in $t = 0$ surface, with a final point in the $t = T_{PH}$ surface, connected by a set of points allowed by the dynamics of the system and the resulting locus being as close to the optimum solution surface.

For the optimisation problem to be able to do this, we must re-formulate this into a "hill-descent" minimisation problem. This is the standard optimisation problem format, where a minima is found for the formulated problem. In our case, we must define a function or a linear combination of functions such that it "descends" from all directions towards the optimum solution space.

Control Sequence Penalty

It is trivial to understand why any non-zero control input sequence must be penalised to find an optimum solution. However, a cost function must be formulated which allows the optimiser to find the direction of "descent" in the cost function hyper-space, hence finding the optimum solution where the cost function has its lowest value within the constrained optimisation space.

The cost function twice differentiated must give a finite value. This is essentially a condition which ensures that the function is continuous and so is its gradient. This is essential for all "gradient descent" optimisation algorithms. For a function defined in a multivariate domain, the gradient of the function assumes the continuity requirements instead of its derivatives.

The chosen cost function is a simple quadratic formulation most typically used in **QP** problems. The mathematical formulation of the control sequence cost function at any given time is given as:

$$C_U(t) = u(t)^2 \tag{5.3}$$

where $C_U(t)$ denotes the control cost at a given time and $u(t)$ denotes the control input at that same instance. To find the normalised net control sequence cost in the control horizon, we integrate it and normalise it with the integration interval, to obtain C_U:

$$C_U = \frac{\int_{t=0}^{t=T_{CH}} u(t)^2 dt}{T_{CH}} \tag{5.4}$$

by substituting the expression of thrust profile from eq. 5.1, we obtain:

$$C_U = \frac{\int_{t=0}^{t=T_{CH}} \left(ae^{-bt}\right)^2 dt}{T_{CH}} \tag{5.5}$$

by expanding it further:

$$C_U = \frac{a^2\left(1 - e^{-2bT_{CH}}\right)}{2bT_{CH}} \tag{5.6}$$

Hence, by implementing the thrust profile function, we can directly compute the normalised net control sequence cost of any optimisation iteration, just by substituting the optimisation variable a and b (thrust profile parameters) into the eq. 5.6.

The continuity of the function can be verified by computing the partial gradients of the function with respect to the optimisation variables. The partial derivatives can be found below:

$$\left.\begin{aligned} \frac{\partial C_U}{\partial a} &= \frac{a - ae^{-2bT_{CH}}}{bT_{CH}} \\ \frac{\partial^2 C_U}{\partial a^2} &= \frac{1 - e^{-2bT_{CH}}}{bT_{CH}} \end{aligned}\right\} \forall\ b \neq 0 \tag{5.7}$$

$$\left.\begin{aligned} \frac{\partial C_U}{\partial b} &= \frac{a^2 e^{-2bT_{CH}}\left(-2bT_{CH} + e^{2bT_{CH}} - 1\right)}{2b^2 T_{CH}} \\ \frac{\partial^2 C_U}{\partial b^2} &= \frac{a^2 e^{-2bT_{CH}}\left(-2b^2 T_{CH}^2 - 2bT_{CH} + e^{2bT_{CH}} - 1\right)}{b^3 T_{CH}} \end{aligned}\right\} \forall\ b \neq 0 \tag{5.8}$$

On further inspection, we can also see that the cost function is convex in the domain $b > 0$. This is because $\left(\frac{\partial^2 C_U}{\partial b^2},\ \frac{\partial^2 C_U}{\partial b^2} > 0\ \forall\ \{b,\ T_{CH} > 0\}\right)$.

Range Bounds Violation Penalty

The cost function for the range bounds violation cannot be developed using a simple quadratic cost function. This is due to the fact that there exist no reference trajectory which can be used as a baseline to compute the predicted trajectory error. The basic responsibility of the range bounds violation penalty is to penalise the portion of the predicted trajectory that violates the relative range bounds.

As discussed in the previous section, the cost function must be continuous and twice differentiable in the domain of the constrained optimisation problem. Hence a suitable candidate for the trajectory cost function must satisfy both these conditions namely, be continuous and twice differentiable and also penalise the bounds violations. An added advantage could be to have a cost function that is convex within the optimisation problem domain.

A choice was made to penalise the extent of bounds violation in a linear manner. Hence, a suitable candidate was found from the field of neural networking. One of the activation functions being used predominantly in this field is called the **ReLU**. The typical characteristics of these cost functions are that they offer linear costs activated after a threshold value. For any value less than the threshold, the **ReLU** function returns a zero. This is the required behaviour in our case, since only the trajectory which lies beyond the threshold (Range bounds) must be penalised linearly while any trajectory which lies within the bounds must not be penalised.

However, the disadvantage of a **ReLU** functions is that they are only piecewise linear, hence by definition, they do not have a continuous second derivative. This would pose a problem in the **QP** approach in the optimisation. Therefore, an approximation for the **ReLU** function was formulated, namely **CReLU**, as follows:

$$f_{CReLU}(x) = \frac{x - p}{\frac{1}{d} + e^{(p-x)}} \tag{5.9}$$

where x is the input variable, p is the function threshold and d is the slope scaling factor which controls the slope of the linear arm of the function, as shown in Fig. 5.6:

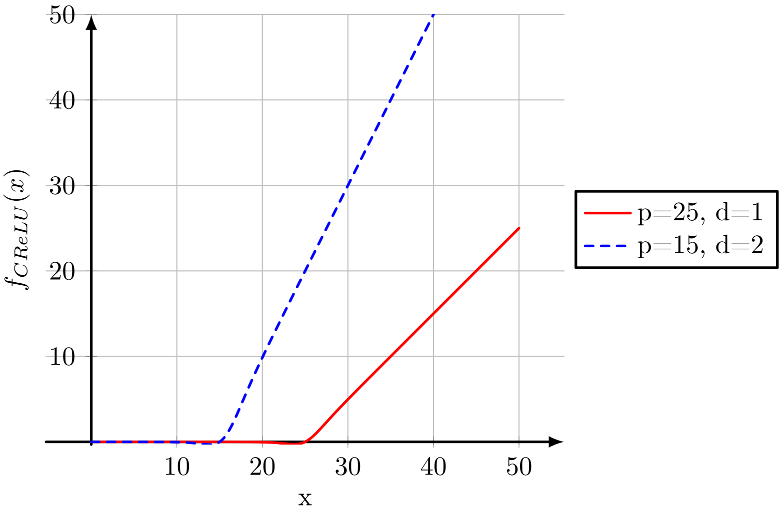


Figure 5.6: CReLU function: parametric combinations.

We can now utilise this knowledge to construct our range cost function $C_R(\rho)$ as follows:

$$C_R(\rho) = \frac{\rho_{min} - \rho}{\frac{1}{d_{min}} + e^{(\rho - \rho_{min})}} + \frac{\rho - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho)}} \tag{5.10}$$

where ρ_{min} and ρ_{max} denote the minimum and maximum range bounds, d_{min} and d_{max} are the slope scaling factor for the corresponding minimum and maximum bounds violation arms of the function. These 4 parameters can be selected appropriately to customise our cost function and to choose how to relatively penalise the range bounds violation. To penalise the maximum range bound violation more than the minimum range bound violation we can simply choose $d_{min} > d_{max}$. An example of this is shown below:

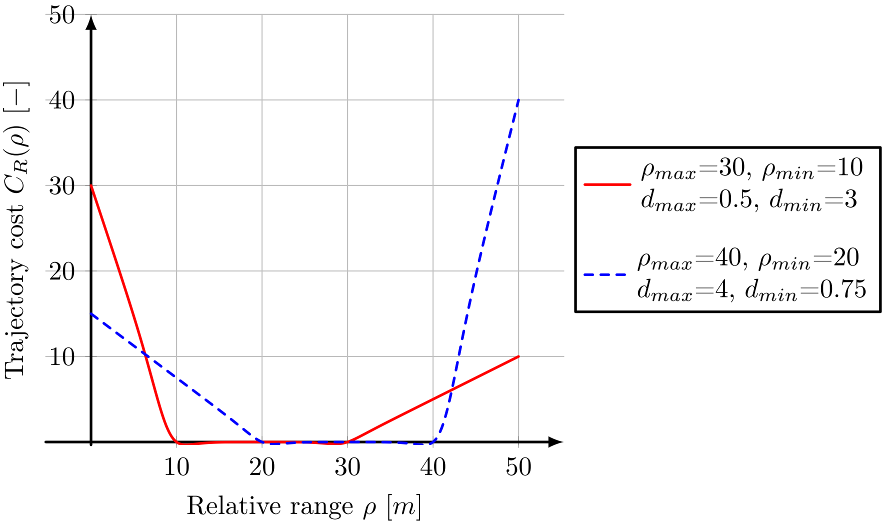


Figure 5.7: MPC range bounds violation cost function.

This results in a cost function which allows customisation to tweak the relative bound limits and also the relative penalty of maximum and minimum bounds violation independently. However, this function is defined in the continuous-time domain which cannot be actually applied in the **MPC**. This is because the **MPC** forecasting algorithm outputs a discrete-time range prediction. To make the trajectory cost function compatible with the forecasting algorithm, we must define it in discrete-time domain, which can be written as:

$$C_{R_k}(\rho_k) = \frac{\rho_{min} - \rho_k}{\frac{1}{d_{min}} + e^{(\rho_k - \rho_{min})}} + \frac{\rho_k - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho_k)}} \tag{5.11}$$

where the subscript k denotes the sampling instance at which the function is evaluated. We can also formulate the function above to get the normalised net trajectory forecast cost using the following formula:

$$C_R = \frac{\sum_{k=0}^{N_{PH}} \left(\frac{\rho_{min} - \rho_k}{\frac{1}{d_{min}} + e^{(\rho_k - \rho_{min})}} + \frac{\rho_k - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho_k)}} \right)}{N_{PH}} \tag{5.12}$$

However, the normalised net control sequence cost in eq. 5.6 is normalised with the prediction horizon but the normalised net trajectory forecast cost in eq. 5.12 is normalised with the number of samples in the prediction horizon. To make both these compatible, we must normalise the eq. 5.12 with the prediction horizon. This results in:

$$C_R = \frac{\sum_{k=0}^{N_{PH}} \left(\frac{\rho_{min} - \rho_k}{\frac{1}{d_{min}} + e^{(\rho_k - \rho_{min})}} + \frac{\rho_k - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho_k)}} \right)}{T_{PH}} \tag{5.13}$$

An important aspect of this cost function is its independence with regards to the thrust parameters a and b. Since, the thrust parameters are utilised as the optimisation variables, this function can be treated as a constant with regards to the lack of gradient in a and b.

5.3. Control Sequence Optimisation

The optimisation of the control sequence must be conducted through a minimisation of an objective function which is a function of the optimisation variables (thrust parameters) a and b. The objective function which is utilised is constructed by a combination of the normalised net trajectory forecast cost and the normalised net control sequence cost. The exact formulation can be derived by using eq. 5.6, eq. 5.13 and a relative gain G_{rel} which controls the relative penalty of trajectory cost and the control cost.

$$\begin{aligned} C_{net}(a,b) =& C_R + G_{rel} C_U \\ C_{net}(a,b) =& \frac{\sum_{k=0}^{N_{PH}} \left(\frac{\rho_{min} - \rho_k}{\frac{1}{d_{min}} + e^{(\rho_k - \rho_{min})}} + \frac{\rho_k - \rho_{max}}{\frac{1}{d_{max}} + e^{(\rho_{max} - \rho_k)}} \right)}{T_{PH}} + G_{rel} \frac{a^2 \left(1 - e^{-2bT_{CH}}\right)}{2bT_{CH}} \end{aligned} \tag{5.14}$$

We can utilise the principles of **QP** to solve for the minima of this objective function. However, we must define the domain under which the objective function is continuous and convex. Since the term C_R is independent of a and b, we can treat it as an arbitrary constant. Hence, the objective function is a linear combination of a constant and a convex function C_U as already discussed in eq. 5.8. Which makes the objective function convex in the domain of convexity of the function C_U.

According to eq. 5.8, the function C_U is convex in the domain $b > 0$. Similarly, we can incorporate the thrust limits into the thrust parameters, as per discussion of eq. 5.1. Utilising this knowledge, we can represent our optimisation problem in a new space defined by the optimisation variables (thrust parameters). This is shown in the figure below:

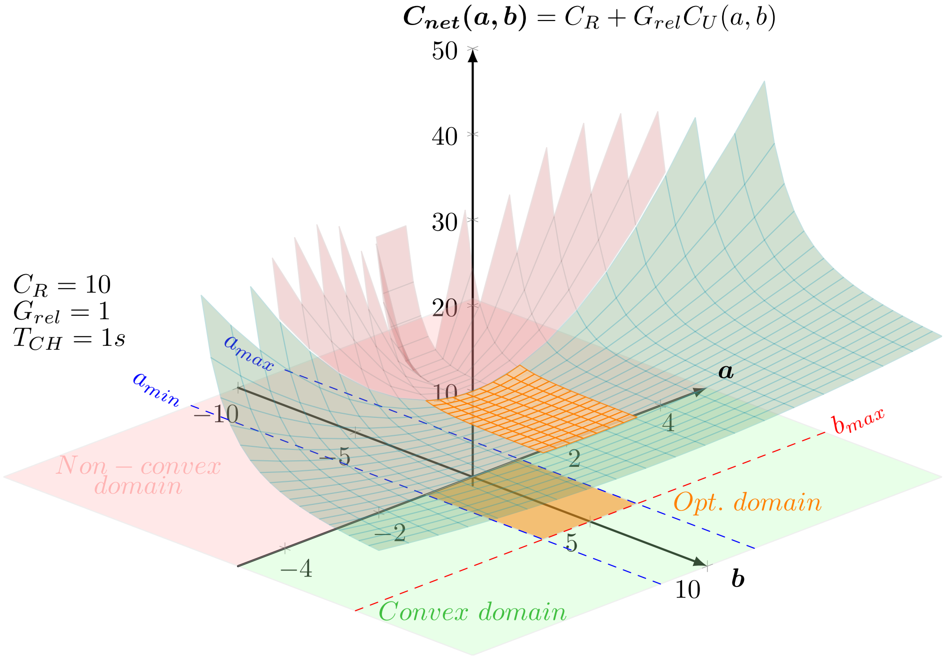


Figure 5.8: MPC constrained optimisation domain.

This is essentially a transformed representation of the principle shown in Fig. 5.5. We can finally write the mathematical expression of the optimisation problem in the **MPC**, as follows:

$$\left.\begin{aligned}
&\min_{\boldsymbol{a},\boldsymbol{b}} C_{net} = \frac{\sum_{k=0}^{N_{PH}} \left(\frac{\rho_{min}-\rho_k}{\frac{1}{d_{min}}+e^{(\rho_k-\rho_{min})}} + \frac{\rho_k-\rho_{max}}{\frac{1}{d_{max}}+e^{(\rho_{max}-\rho_k)}} \right)}{T_{PH}} + G_{rel}\frac{a^2\left(1-e^{-2bT_{CH}}\right)}{2bT_{CH}}\\
&a \in [-1,\ 1]\\
&b \in (0,\ 5]
\end{aligned}\right\} \qquad (5.15)$$

As a note, there is no explicit constraint or bounds established for the relative range control within the objective function. The control is achieved through the minimisation of the cost function within the optimisation domain defined through constraints, as denoted in Fig. 5.8.

The assumption made about the convexity of the objective function with respect to the optimisation variables a and b, is only valid under the assumption of very small prediction horizons. This stems from the fact that the normalised net range violation cost function is non-linearly dependent on the optimisation variables for a larger prediction horizon.

To tackle this issue, a sequential quadratic programming optimiser algorithm was utilised to minimise the objective function. The specific **SQP** optimiser is characterised by the use of quadratic programming subproblems at every iteration. The **SQP** optimiser is considered as a novel and cutting-edge method in quadratic optimisation methods. This helps in avoiding explicit dependence on the gradient based optimisation techniques such as Newton gradient optimiser. An additional advantage is the relaxation of the requirement of a convex objective function in the entire optimisation domain. This is because the **SQP** optimiser constructs an approximate quadratic sub-problem at every iteration, which is then used to find the "descent" direction.

Hence, **SQP** optimiser only requires a twice-differentiable continuous objective function with linear constraints. This is the primary reason behind the choice of the cost functions to be continuous and twice-differentiable in the optimisation domain. The constraints defined for our problem are inequality constraints forming a well defined boundary, which helps the **SQP** algorithm in faster and more stable convergence along with ensuring both feasibility and ensuring constraint non-violation at every iteration.

Regardless, only the choice of the optimiser algorithm is changed, there is no change in the formulation of the optimisation problem described in eq. 5.15. The exact formulation and software implementation of the algorithm is beyond the scope of this thesis, but can be found in Ref. [37].

5.4. MPC Software Integration

With a completely developed mathematical model of the **MPC** and the associated model discovery and plant dynamics, we can now assemble them into one single simulation environment. This is done in a similar way to the model discovery framework implementation, explained through Fig. 4.1. We can use the same type of flowchart representation and depict the implementation of the **MPC** framework integrated with the associated model discovery and plant dynamics, as represented below:

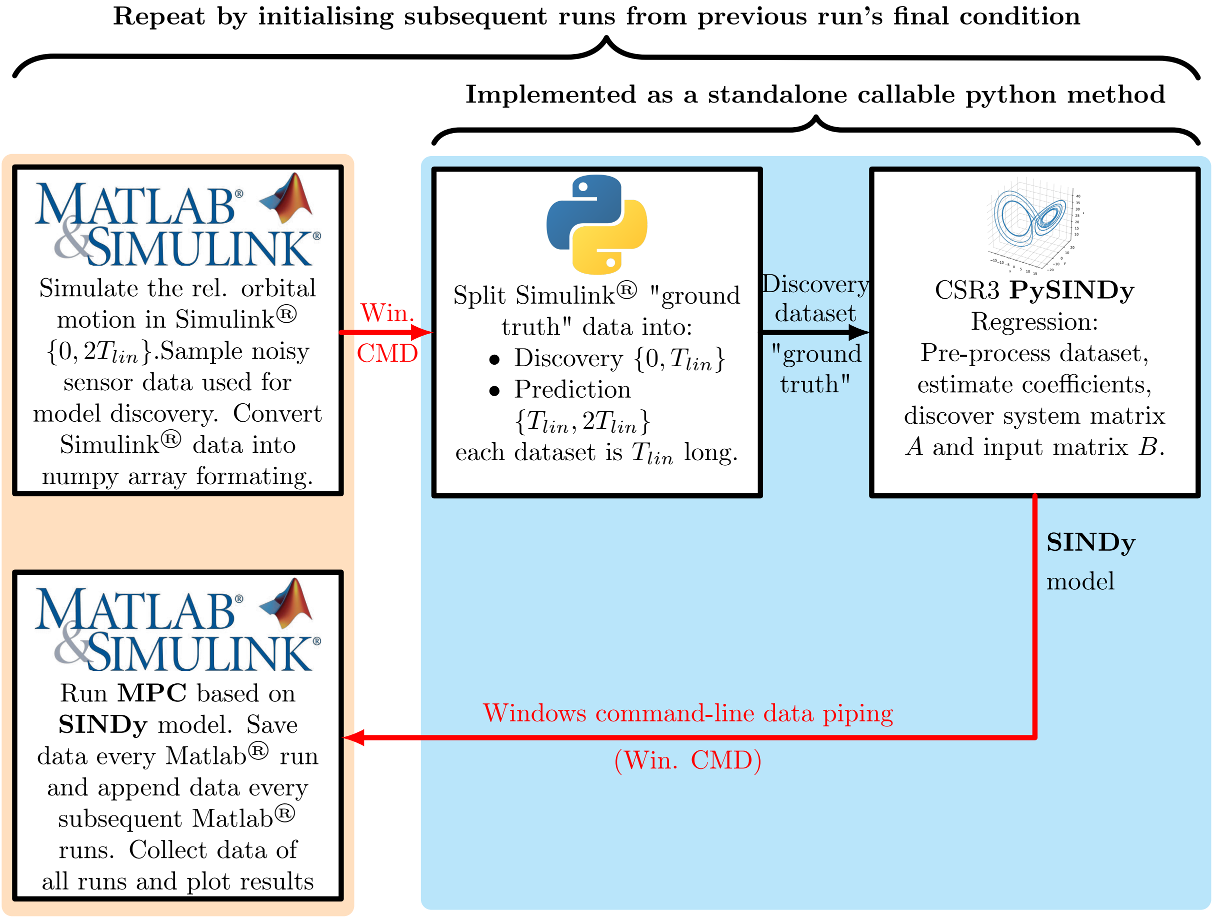


Figure 5.9: Execution flowchart.

While the figure above represents the software interactions and implementations, we need to also understand the classical block representation of the closed-loop control system. The majority of the closed-loop system is implemented in the same software, except the model discovery algorithm.

This is due to lack of software tools available to implement the chosen model discovery framework along with the closed-loop control system, within the same software. However, the model discovery framework is not actively participating in the closed-loop simulation. It is "called" only at simulation time instances when the **MPC** controller needs a new and linearised estimate of the local dynamics, i.e. every T_{lin} s. Finally, the control system design concludes with the closed-loop block representation, as shown in the classical notation below:

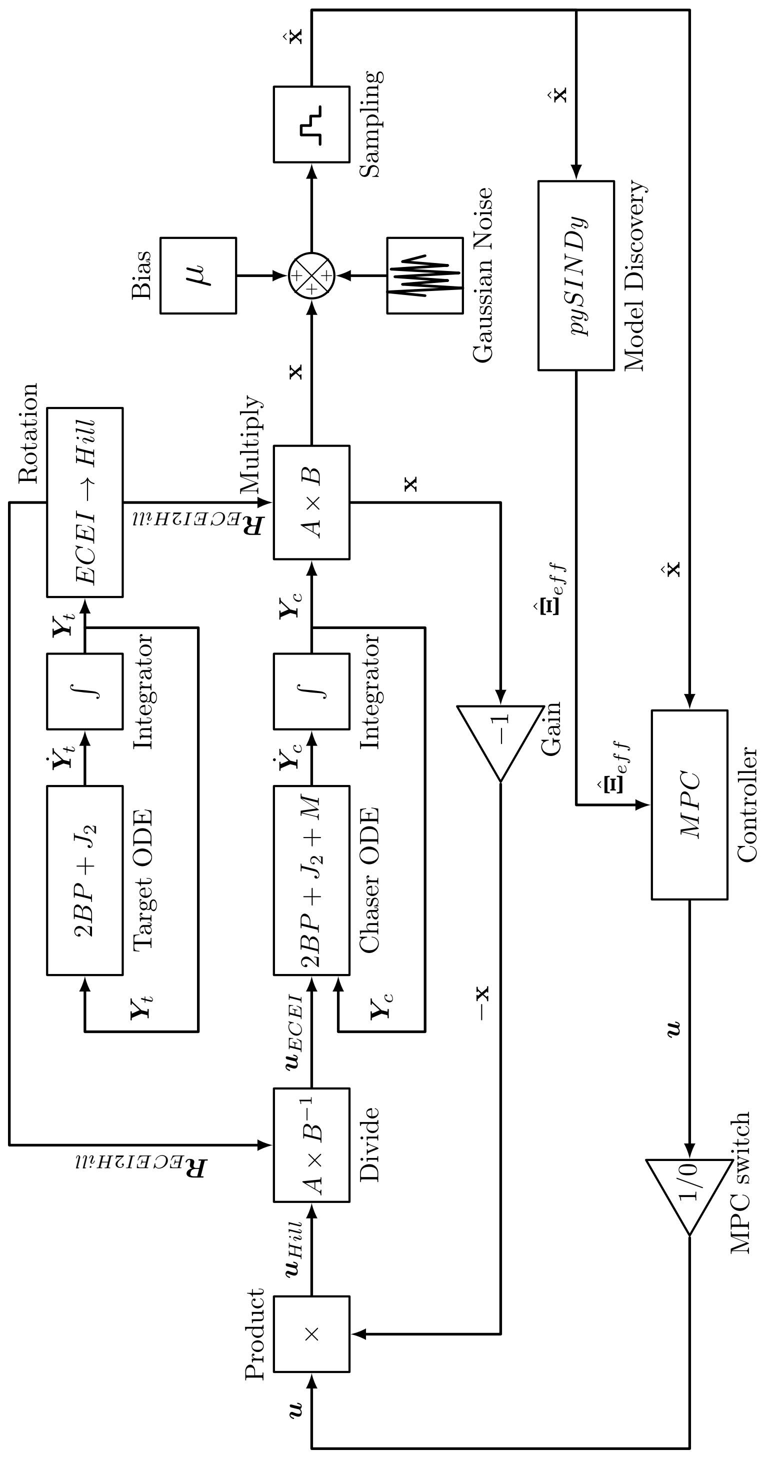
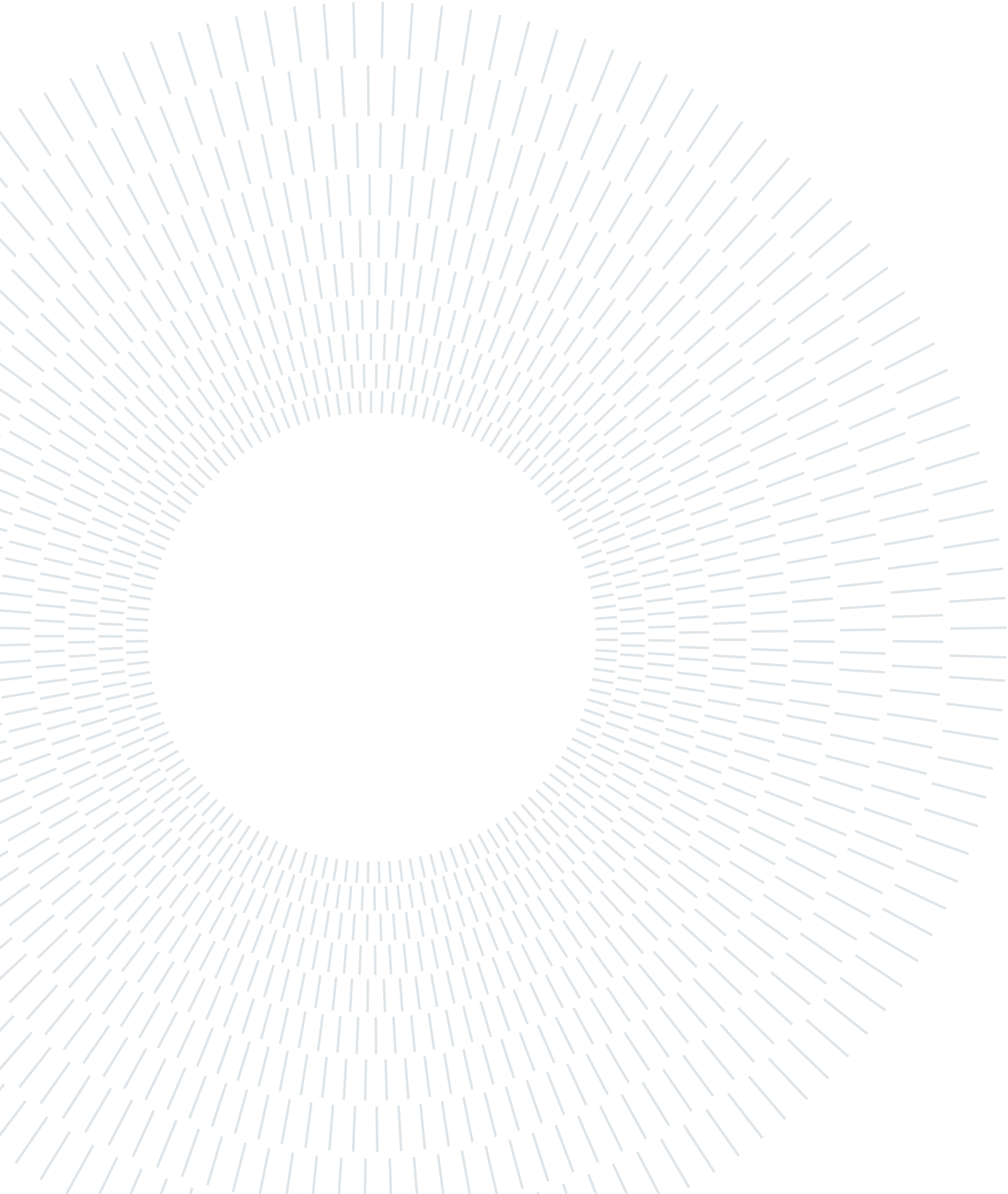


Figure 5.10: Complete block diagram representation.

This concludes the modelling of the complete closed-loop control system. We can now proceed to setup and run the simulations to test and analyse the closed-loop performance of the control system. This is discussed in detail in the following chapter.



6 | Simulation Campaign

The simulation campaign was undertaken to establish a benchmark of the performance of the designed **MPC** and model discovery framework working in tandem closed-loop control system configuration. The campaign was designed to test the functionality of the closed-loop control system and also to analyse the effects of the orbital environment, **MPC** parameters i.e. prediction and control horizons and the relative range bounds.

Before proceeding to presentation of the results, the specific notations used in the results must be introduced. The notations used and their meaning is summarised below:

Terms	**Usage**	**Meaning**
$\boldsymbol{d_T}$	$d_T : d_{min}/d_{max}$	Trajectory cost function: slope scaling factors
$\boldsymbol{G_{rel}}$	G_{rel}	Cost function relative gain
$\boldsymbol{T_s}$	T_s	**MPC** time step (samling time)
$\boldsymbol{a_T}$	a_T	Initial **SMA** of target spacecraft
$\boldsymbol{e_T}$	e_T	Initial eccentricity of target spacecraft
$\boldsymbol{f_T}$	f_T	Initial true-anomaly of target spacecraft
$\boldsymbol{N_H}$	N_{PH}/N_{CH}	**MPC** prediction and control horizon time steps
$\boldsymbol{\rho_{lim}}$	ρ_{min}/ρ_{max}	Relative range bounds
$\boldsymbol{T_{lin}}$	T_{lin}	Linearisation interval of model discovery

Table 6.1: Simulation campaign notations.

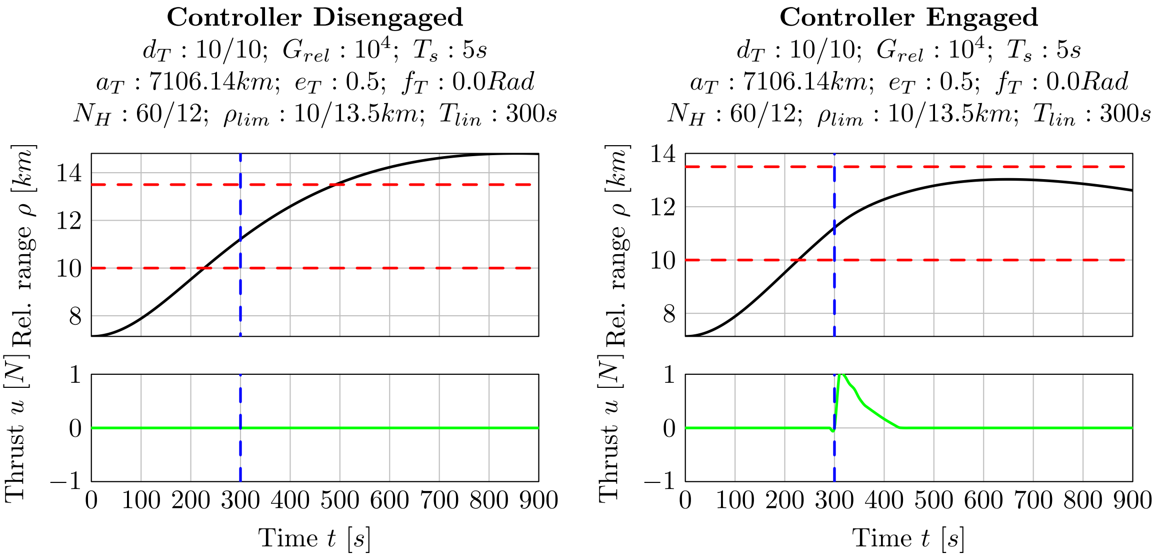
We can now proceed with the results pertaining to the simulation campaign. The campaign has been broadly divided into functional testing, orbital environment study, **MPC** horizons study and finally the **MPC** bounds study. Each focusing on understanding the effects of the corresponding factors in the control system performance. Each study has been conducted under various simulation settings to cover a wide range of cases. Primary focus was laid on the effect of degree of non-linearity on the control system performance through variation of the **SMA**, eccentricity and true-anomaly.

6.1. MPC Functional Testing

Before proceeding with the analysis of the closed-loop control system performance, it is imperative to establish its capability to effectively control the spacecraft within the constraints and bounds. To this end, a simulated spacecraft tandem system was subjected to the control system on the chaser spacecraft. The target spacecraft is under no controlled inputs and hence is governed by the homogeneous **NERM**. As discussed in the preceding chapters, the constraints of the control system are manifested in the thruster maximum thrust limit ($u_{max} = 1N$), persistently target facing thruster pair and the bounds for the control system are the relative range limits ρ_{lims} implemented indirectly through an objective function.

To offer a comparison, two simulations were run, one without any controlled input and one with it. The simulation was run with a short simulation time of $900s$, just to verify the closed-loop control system efficacy. A primary characteristic of all the conducted simulations is the lack of control inputs during the first linearisation interval. This is because during the first linearisation interval, the **MPC** lacks information of the plant model.

The results can be seen in the figure below, with the relevant information about the orbit locality, controller and bounding parameters shown in their corresponding sub-figures:



(a) Simulation A. (b) Simulation B.

Figure 6.1: Simulation Environment: LEO, elliptical, set-1.

It is evident from Fig. 6.1b that the implemented closed-loop control system is working effectively to keep the chaser spacecraft within the requested bounds while limiting the control input within the constraints. Due to the lack of exact tuning methods to choose the **MPC** objective function weights and **MPC** horizons, for a primary investigation, the chosen values were derived through a hit-and-trial basis but with basic understanding of the underlying implications.

6.2. Orbital Environment Study

The control system designed in this thesis is successful in controlling the relative range of the chaser spacecraft within the bounds. However, to study the effects of various orbital environments on the control system performance, a series of simulations were conducted for multiple initial conditions. The chosen orbital environments for the test can be summarised as follows:

Case Type	**SMA $[km]$**	**Eccentricity**
Beyond GEO$^+$: Circular	≈ 82106	0
Beyond GEO$^+$: Highly Elliptical	≈ 82106	0.5
Approx. GEO$^+$: Circular	≈ 42116	0
Approx. GEO$^+$: Highly Elliptical	≈ 42116	0.5
LEO: Circular	≈ 7106	0
LEO: Highly Elliptical*	≈ 7106	0.5

$^+$ GEO is representative of only the SMA of the case, not the orbit properties.

* LEO with $e = 0.5$ is not physically possible due to perigee inside Earth.

Table 6.2: Orbital environment study: Simulation cases.

The choice of the orbital environment is not representative of physical possibility or practicality of such an environment, it only serves to authenticate the control system performance at various degrees of non-linearity of relative dynamics, resulting from the mathematical values of the chosen Keplerian orbital elements.

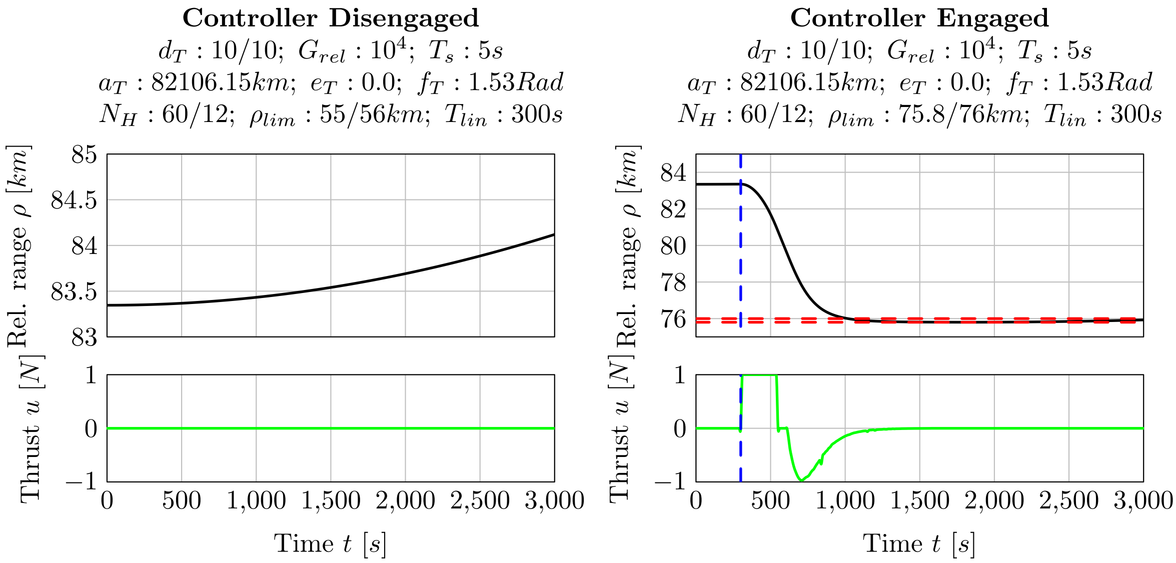
6.2.1. Beyond GEO

This section reports the results of the simulations under the orbital environment study conducted under that case "Beyond GEO" as per Table 6.2. A typical characteristic of this orbital environment is low degree of non-linearity in relative dynamics. Important comments and observations are discussed in relevant sub-sections.

Circular Orbit

Circular orbits offer ideal cases for orbital relative motion controllers. They are characterised by linear relative motion dynamics. This case is made easier by the very high **SMA** of the orbital environment. A higher **SMA** also results in the smaller augmentation of the non-linearity of the dynamics, if present.

The first set of simulations under these conditions represents motion without control effort, as shown in Fig. 6.2a, and corresponding motion controlled by the **MPC**, as shown in Fig. 6.2b. The bounds for relative range are chosen far from the initial relative range and kept narrow. This is essentially commanding the controller to maintain the chaser spacecraft at a fixed range separation with very small allowed deviation.



Controller Disengaged
$d_T : 10/10;\ G_{rel} : 10^4;\ T_s : 5s$
$a_T : 82106.15km;\ e_T : 0.0;\ f_T : 1.53Rad$
$N_H : 60/12;\ \rho_{lim} : 55/56km;\ T_{lin} : 300s$

Controller Engaged
$d_T : 10/10;\ G_{rel} : 10^4;\ T_s : 5s$
$a_T : 82106.15km;\ e_T : 0.0;\ f_T : 1.53Rad$
$N_H : 60/12;\ \rho_{lim} : 75.8/76km;\ T_{lin} : 300s$

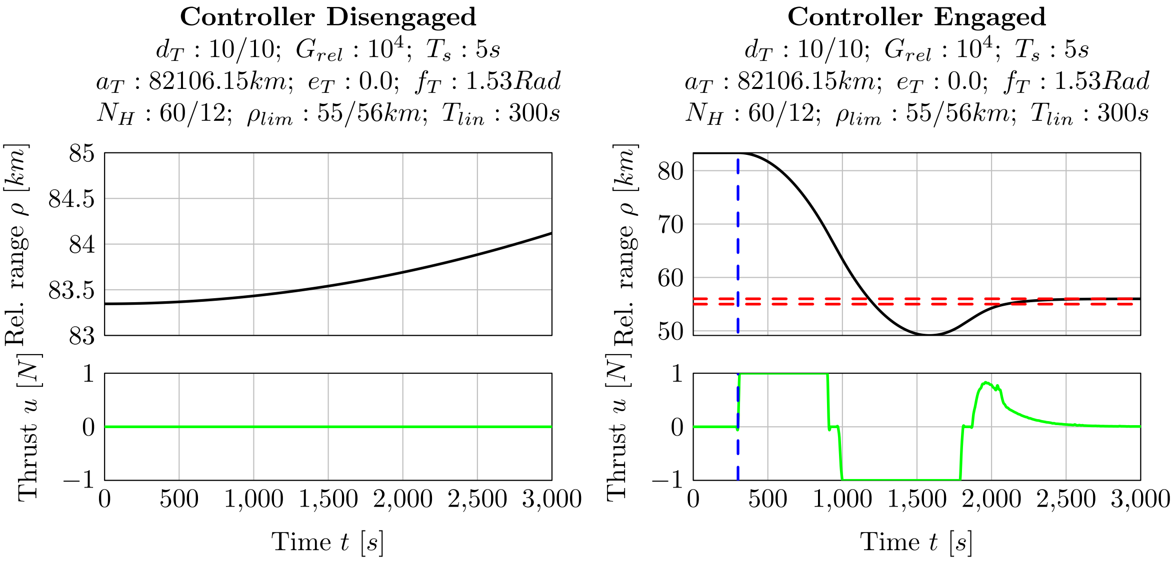
(a) Simulation A.

(b) Simulation B.

Figure 6.2: Simulation Environment: Beyond GEO, circular, set-1.

As it can be seen from the results obtained in Fig. 6.2, it is evident that the **MPC** is able to execute the requested range bounding manoeuvre with great accuracy and limited control effort. The resulting trajectory attains the requested bounding within approximately $700s$ of the controller engagement. Another point to note is a lack of considerable overshoot in the controlled trajectory. This is because the predictor applies corrective control input to prevent overshoot (*at approx.* $600s$), which is almost at the prediction horizon ($300s$) of the expected overshoot instance ($900s$), if no preventive input was applied. Which means that the **MPC** could predict the overshoot and find an optimal control sequence to prevent this, with great accuracy. This is the expected performance advantage of a typical **MPC**.

The next simulation makes the requested bounds much farther from the initial separation, as can be witnessed in Fig. 6.3. This is equivalent to giving a reference signal very far from the current output, in a classical control system perspective. As expected, there is an overshoot in the resulting trajectory, as shown in Fig. 6.3b. Without any change in the **MPC** parameters or orbital environment, the cause of the overshoot can be attributed to a lack of a larger prediction horizon. This results in the **MPC** unable to recognise an overshoot within its optimisation horizon and does not compensate for it. This results in over-actuation and a sub-optimal control sequence. However, the **MPC** can be seen to be quickly responding to correct the overshoot once it predicts it within the prediction horizon, this can be witnessed in Fig. 6.3b.

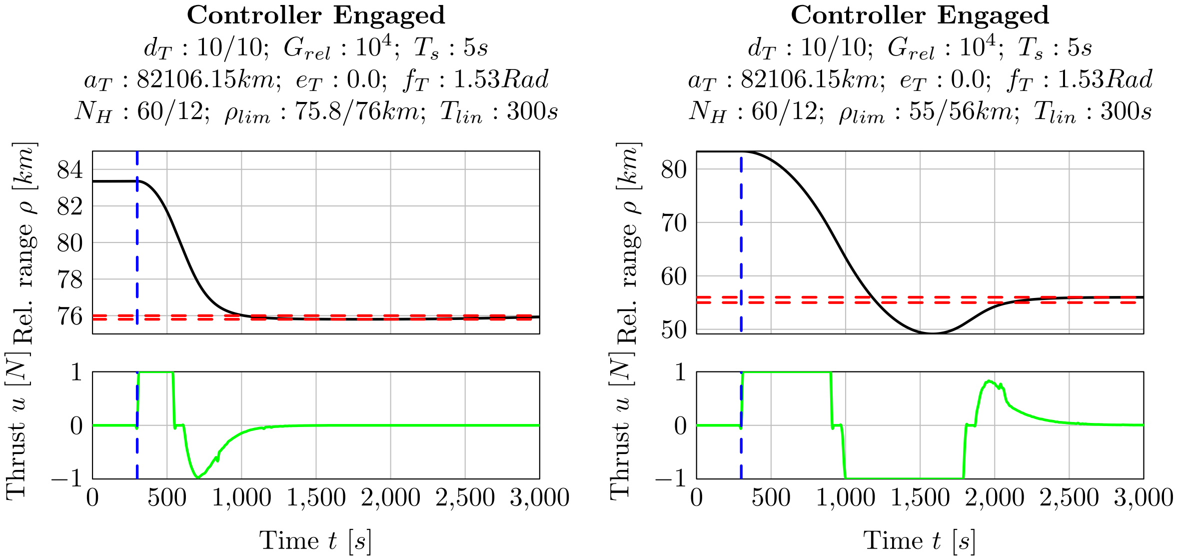


(a) Simulation A.

(b) Simulation C.

Figure 6.3: Simulation Environment: Beyond GEO, circular, set-2.

A potential solution to this problem of overshoot is to increase the prediction horizon. But due to the nature of the model discovery framework, any prediction horizons larger than the linearisation intervals would result in diverging prediction accuracy. Hence, a practical solution would be to request the bounds in a staggered manner, slowly moving from current separation to the final desired bounds. This would be analogous to a ramp reference signal in classical control theory. Since the controller is not designed to account for actuator saturation and compensate with orbital manoeuvres, we witness sluggish responses in the trajectory correction. The comparison between both these cases can be witnessed below for reference.

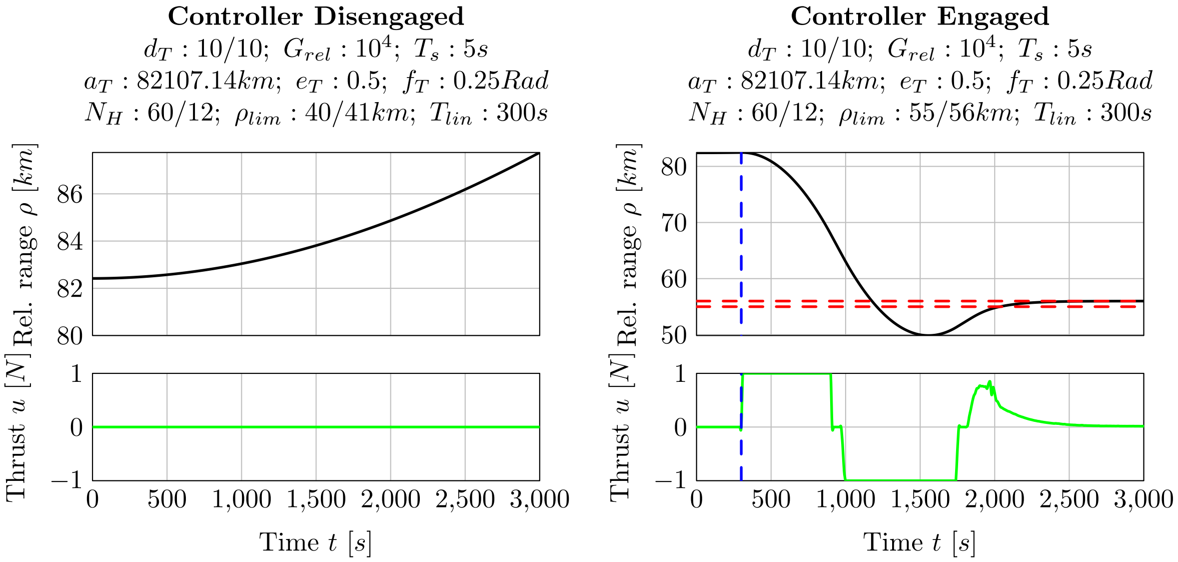


(a) Simulation B. (b) Simulation C.

Figure 6.4: Simulation Environment: Beyond GEO, circular, set-3.

Highly Elliptical Orbit

The worse performing case (Fig. 6.4b) was simulated for a highly eccentric orbit to test the limits of the control system. However, we get very similar performances as seen in Fig. 6.5. This is due to the "diluted" non-linearity in dynamics due to a very high **SMA**.



(a) Simulation A. (b) Simulation B.

Figure 6.5: Simulation Environment: Beyond GEO, elliptical, set-1.

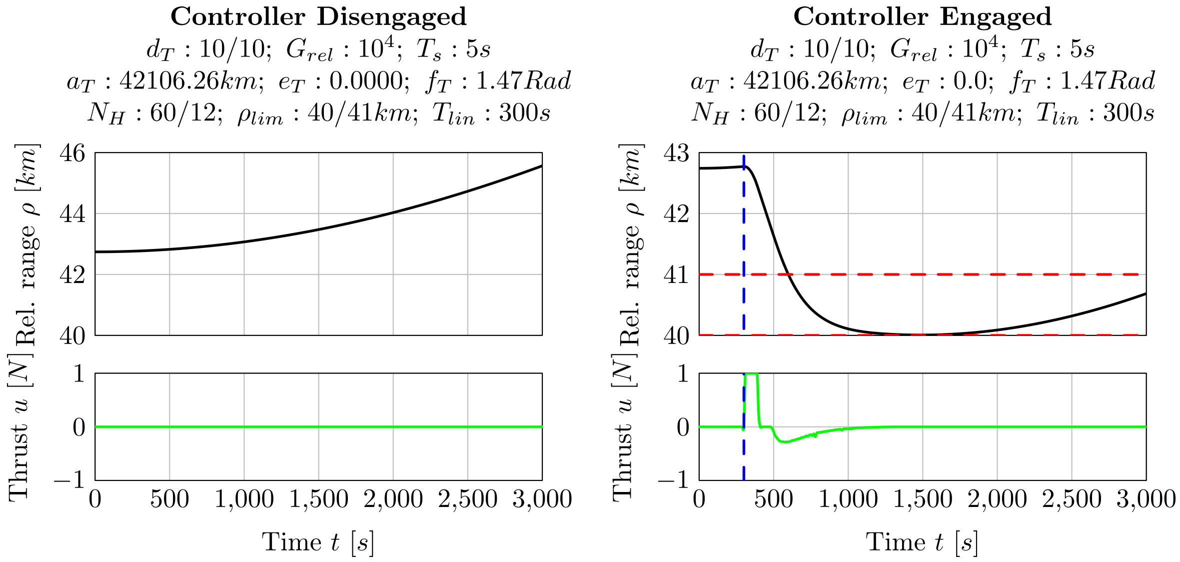
6.2.2. GEO

Let us put the control system to a tougher test in a stronger non-linear orbital environment by reducing the orbital **SMA**. With a smaller orbital **SMA**, the controller would be unable to reach the same bounds as was the case in the previous case of beyond GEO orbit. Hence, the bounds were decided to be placed closer to the initial separation. This is analogous to giving a step reference signal with a smaller magnitude to the controller, in classical control system jargon.

No changes to the **MPC** parameters are done to ensure that these simulations offer insights into the isolated effects of orbital environment on the control system performance without contamination due to effects of change is **MPC** parameters.

Circular Orbit

As discussed in the previous section, the first simulations are conducted within a circular orbit scenario, as seen in Fig. 6.6. Two individual simulations are conducted, with the **MPC** disabled (Fig. 6.6a) and with **MPC** enabled (Fig. 6.6b), for comparative purposes.



Controller Disengaged
$d_T : 10/10;\ G_{rel} : 10^4;\ T_s : 5s$
$a_T : 42106.26km;\ e_T : 0.0000;\ f_T : 1.47Rad$
$N_H : 60/12;\ \rho_{lim} : 40/41km;\ T_{lin} : 300s$

Controller Engaged
$d_T : 10/10;\ G_{rel} : 10^4;\ T_s : 5s$
$a_T : 42106.26km;\ e_T : 0.0;\ f_T : 1.47Rad$
$N_H : 60/12;\ \rho_{lim} : 40/41km;\ T_{lin} : 300s$

(a) Simulation A.

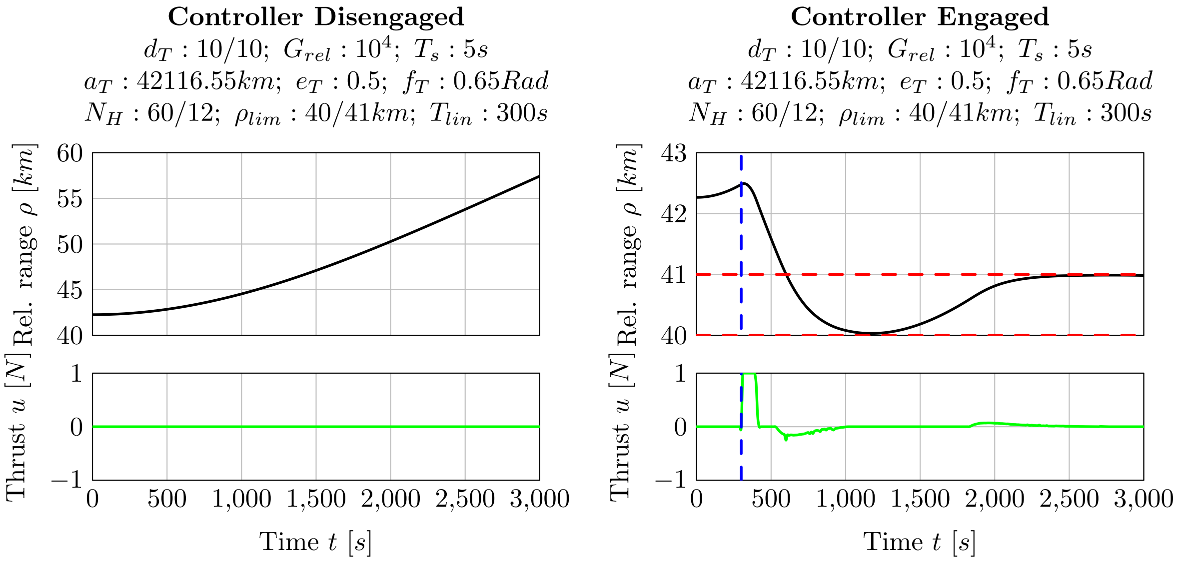
(b) Simulation B.

Figure 6.6: Simulation Environment: GEO, circular, set-1.

The **MPC** is able to accurately bring the trajectory to the requested bounds, against the direction of natural motion, and maintain it. The control sequence applied by the **MPC** can be seen to be smooth and efficient in Fig. 6.6b.

Highly Elliptical Orbit

As discussed previously, in larger orbits, the effect of higher eccentricity on the intensity of the non-linearity of the relative dynamics is low. This can be witnessed again in the Fig. 6.7, where the **MPC** can be seen to handle the requested bounds with relative ease. A slight difference in Fig. 6.7b from Fig. 6.6b can be seen approximately at $1850s$ into simulation. The **MPC** can be seen to prevent the escape from bounds in Fig. 6.6b, but this is absent in Fig. 6.7b.



(a) Simulation A.

(b) Simulation B.

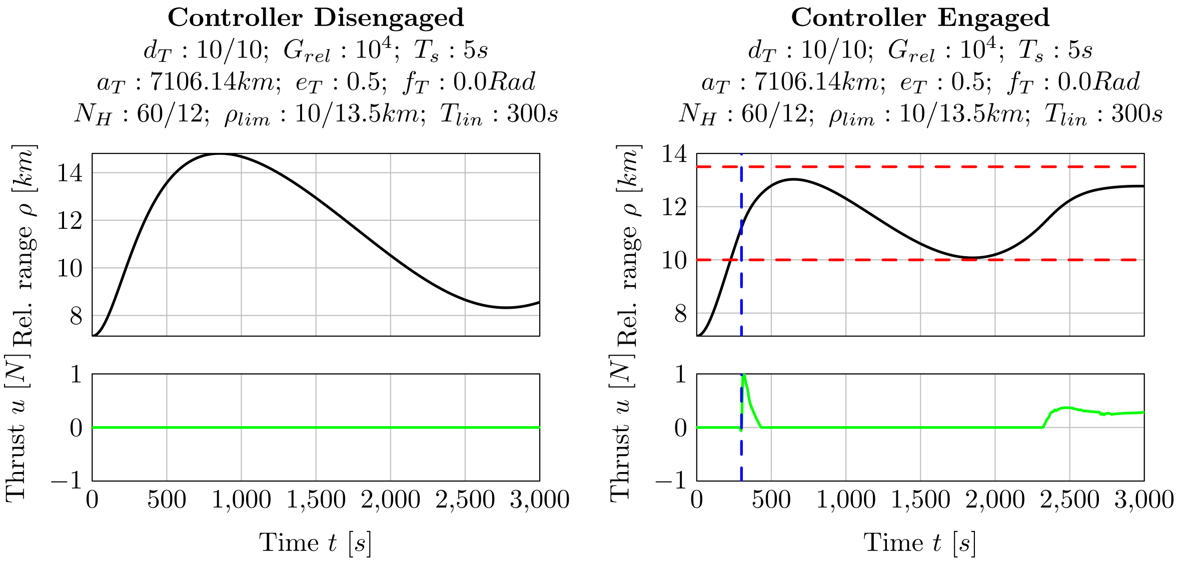
Figure 6.7: Simulation Environment: GEO, elliptical, set-1.

6.2.3. LEO

Since the **MPC** is able to handle the higher orbital **SMA**s with relative ease, we can push it to the toughest test. The low-Earth orbit is the most challenging environment due to highly non-linear behaviours at very low eccentricities and the higher effect of the J_2 perturbations which are a challenge for the model discovery framework.

Highly Elliptical Orbit

To simulate the toughest orbital environment, a choice was made to position the tandem spacecraft formation at the perigee of the low-Earth orbit. The perigee of an elliptical orbit is characterised by the highest degree of relative motion non-linearity in the entire orbit. The resulting motion under control and without is presented in Fig. 6.1.



(a) Simulation A.

(b) Simulation B.

Figure 6.8: Simulation Environment: LEO, elliptical, set-2.

With due consideration to the tough orbital environment, the bounds were chosen to be placed within the natural trajectory limits. This meant that the **MPC** would not need to employ excessive actuation which could saturate the actuators. This can be seen in Fig. 6.8b. However, even this orbital environment could be handled with ease by the designed **MPC**. However, this could be suspected due to the short simulation times. Hence, long-term simulations are conducted in the next section with experimentation on the change of **MPC** parameters and their effect on the performance.

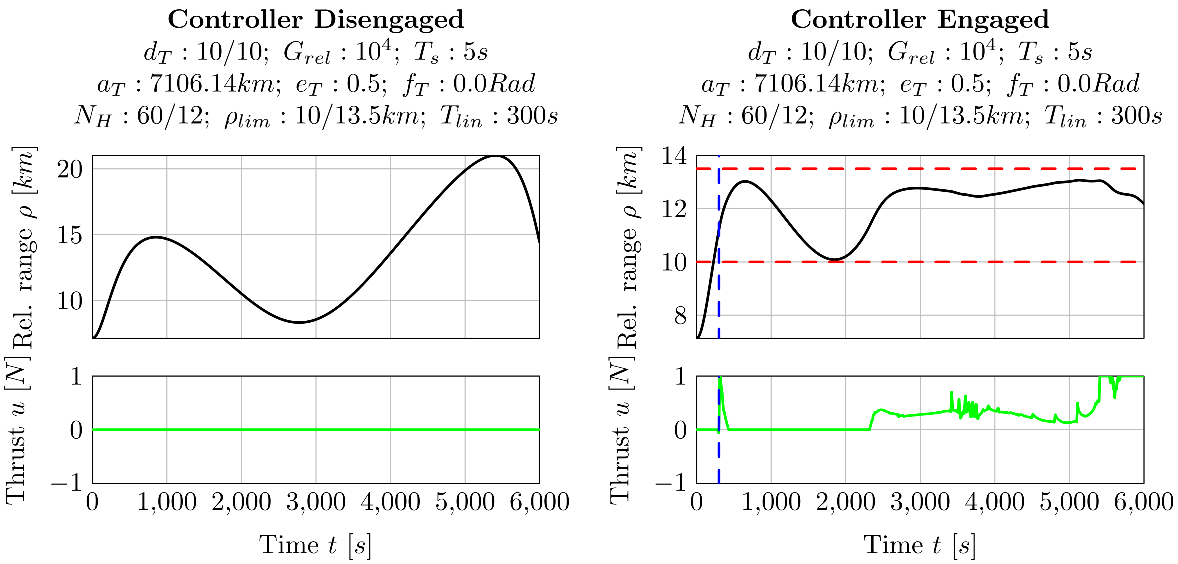
6.3. MPC Parameter Study: LEO

Within the scope of the previous sub-section, it is well established that the **MPC** can handle a wide range of orbital environments, even with the default **MPC** parameters. To investigate the effects of the change in the **MPC** parameters on the performance, the simulations were conducted on the most demanding orbital environment - low-Earth orbit with high eccentricity and J_2 perturbations.

To analyse the efficacy of the **MPC** on longer intervals, simulations were carried out for the whole duration of a single orbit. This would also ensure that the model discovery was working as expected, under varying true-anomaly and the long-term stability of the algorithm in under heavy J_2 perturbations. However, the main concern was the lack of control authority to counter the perturbations and non-linearity over long intervals.

6.3.1. Orbital Period Simulation: MPC parameters

The first set of simulations conducted under this study are done to establish a baseline performance. The same **MPC** parameters were used as in previous sections, to first understand the effect of a long interval simulation. The results are presented in Fig. 6.9, with natural motion shown in Fig. 6.9a and the controlled trajectory in Fig. 6.9b.



(a) Simulation A.

(b) Simulation B.

Figure 6.9: Simulation Environment: LEO, elliptical, set-3.

It can be clearly observed that the **MPC** is facing much more difficulty in maintaining a smooth trajectory under long-term simulations. This can be attributed to two primary factors.

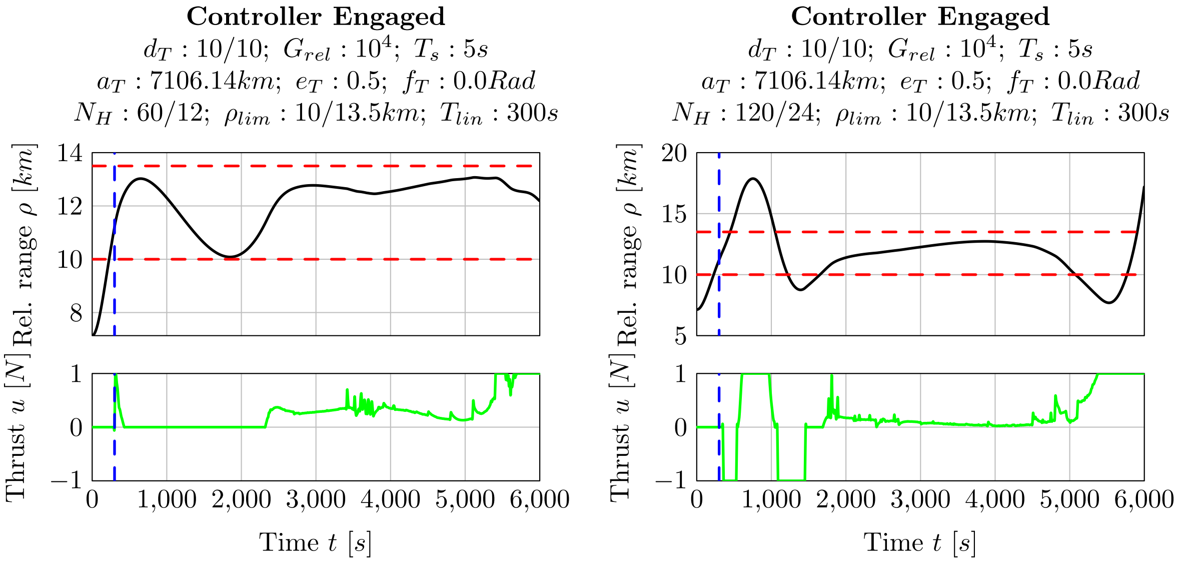
The first is the naturally occurring heavy non-linearity and extreme perturbations at the perigee. This is encountered by the **MPC** in Fig. 6.9b towards the end of the simulation. We can witness the sudden saturation of the actuators near perigee. However, an actual orbit of the same **SMA** and eccentricity cannot occur physically due to the perigee being inside the Earth. Thus, the **MPC** would not be expected to encounter such extreme dynamics in actual applications.

The second reason is the fast changing dynamics being linearised over a relatively long interval compared to the short orbital periods. This produces slight inaccuracies in the trajectory prediction algorithm. The primary assumption of utilising constantly updating linear models to predict the trajectory is that the linearisation interval must be much smaller than the orbital period. This was satisfied for the case of higher orbits, but failed for lower orbits. This problem is also exemplified at the perigee due to even closer proximity to Earth, which in turn results in a fast evolving dynamics.

A prospective solution for this would be to reduce the linearisation interval to compensate for the lower orbit's fast changing dynamics. But this would reduce the prediction horizon of the **MPC**. To strike a balance, it is suggested to implement an adaptive algorithm to select the linearisation interval from the lookup table generated from the results of the testing campaign as per sub-section 4.5.2. This would allow the **MPC** to choose the highest linearisation interval within an acceptable prediction error based on the current true-anomaly within the orbit.

MPC Horizons Study

To understand the effect of altering the **MPC** horizons on its performance, a comparative simulation was run. The chosen **MPC** horizon for the new simulation, as shown in Fig. 6.10b, is twice the default parameters shown in Fig. 6.9b.



(a) Simulation A.

(b) Simulation B.

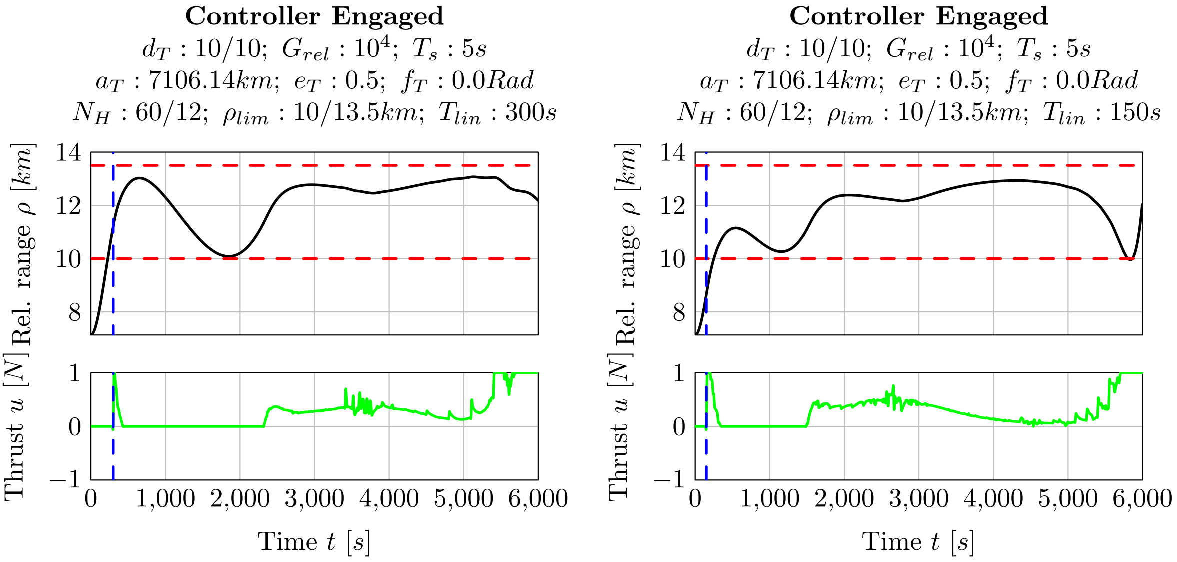
Figure 6.10: Simulation Environment: LEO, elliptical, set-4.

The primary difference observed in the new **MPC** performance compared to the default **MPC** horizons, is the trajectory smoothness at the cost of bound violations at the perigee. This can be explained by understanding the nature of the model discovery framework and its role in deciding the **MPC** prediction horizon.

As discussed previously, the accuracy of prediction quickly worsens if the prediction horizon exceeds the linearisation interval. This leads to inaccurate predictions of the trajectory, especially in orbit localities of high nonlinearity i.e. perigee. Hence, the **MPC** performance degrades at that corresponding locality. But for the rest of the orbit, the prediction accuracy is still maintained under acceptable limits, ensuring a more optimised control sequence implementation resulting in a smoother trajectory.

MPC Linearisation Interval Study

To verify the hypothesis regarding the shorter linearisation interval leading to a smoother control input and tolerance against the constantly evolving relative dynamics, a new simulation was run by reducing the linearisation interval. The **MPC** with a smaller linearisation interval of $150s$ compared to the default **MPC** linearisation interval of $300s$, does work to a certain extent according to the prediction of the hypothesis. This can be seen in the comparison shown in Fig. 6.11.



(a) Simulation A.

(b) Simulation B.

Figure 6.11: Simulation Environment: LEO, elliptical, set-5.

There are no major performance improvements due to the reduction in the linearisation interval. This might seem counter-intuitive due to the fact that a smaller linearisation interval must lead to a more accurate and updated model discovery framework. However, since the simulation was conducted by altering the linearisation interval only, without the corresponding reduction in the prediction and control horizons, the accuracy of the **MPC** predictions are worse. This compensates for any augmentation of prediction accuracy due to reduction of linearisation interval.

Hence, this simulation indicates that due to the coupled nature of the model discovery framework and the **MPC** prediction horizons, both parameters must be changed in tandem to see its effect on the actual performance. Any change in either parameter without a corresponding change in the other, would bear undesirable effects on **MPC** performance.

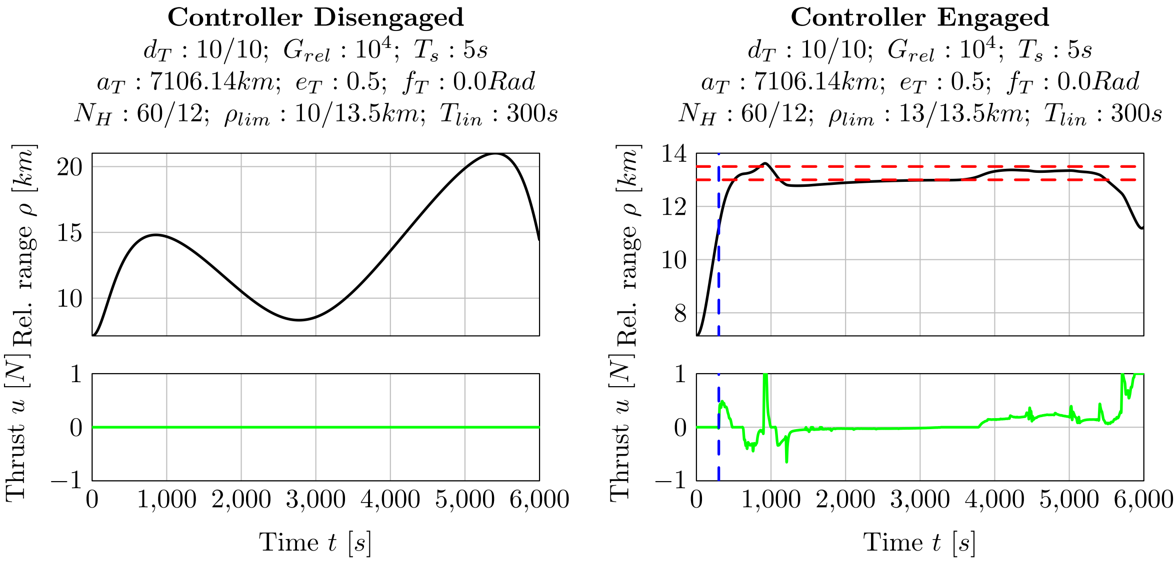
6.4. MPC Bounds Study: LEO

A final study was conducted on the effect of the nature of the relative range bounds on the **MPC** performance. Again, as a benchmark, the worst case of low-Earth highly elliptical orbit at the perigee was chosen as the initial condition for the simulation. The study aims to understand the effect of imposing narrow bounds within the natural trajectory bounds compared to imposing them outside the limits of the natural motion.

To ensure uniformity, the **MPC** parameters were reverted back to the default values as prior to the **MPC** parameters study was conducted.

6.4.1. Narrow Bounds Within Original Trajectory

For the first case, the narrow bounds were imposed within the original trajectory bounds but offset from the mean relative range. The resulting performance can be seen in Fig. 6.12. When compared to the natural trajectory shown in Fig. 6.12a, the controlled motion can be seen to maintain the requested bounds for the majority of the orbital duration.



(a) Simulation A.

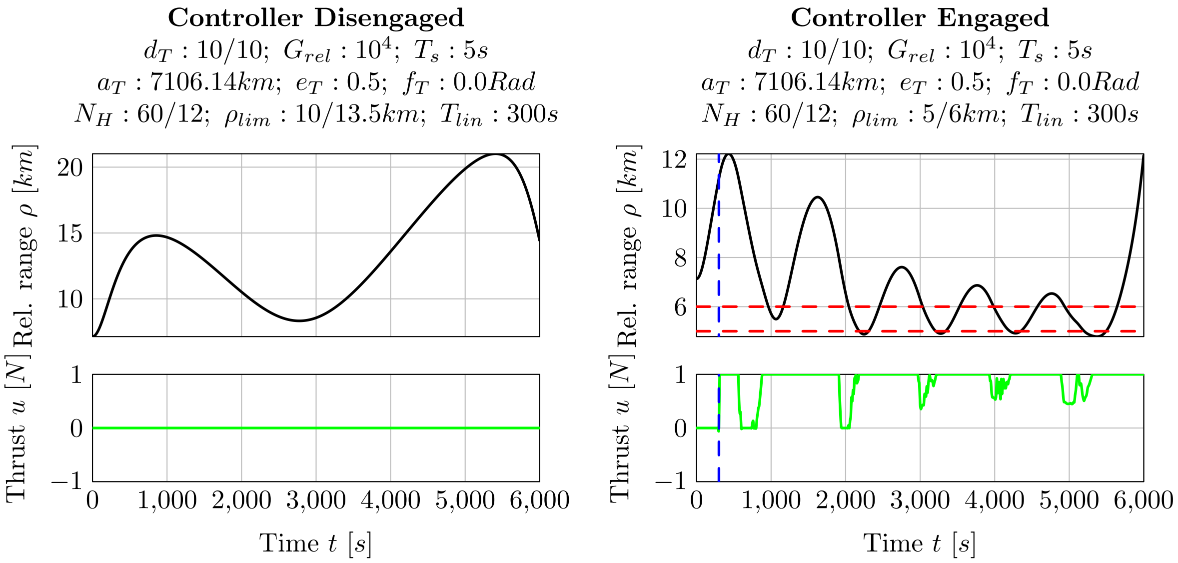
(b) Simulation B.

Figure 6.12: Simulation Environment: LEO, elliptical, set-6.

As was the case with many of the previous simulations, the **MPC** performance is hampered at the perigee towards the end of the simulation. This can be witnessed through the saturation of the actuator inputs and the violation of the requested bounds consequentially. However, this can be overcome by a higher powered actuator assembly.

6.4.2. Narrow Bounds Outside Original Trajectory

As a final case, the narrow relative range bounds are assigned outside the natural motion limits and against the natural motion drift. The natural motion can be seen drifting towards higher relative ranges in Fig. 6.12a. This can be attributed to the strong influence of the J_2 perturbations. This leads to the most difficult scenario that the **MPC** has been put to test in.



(a) Simulation A.

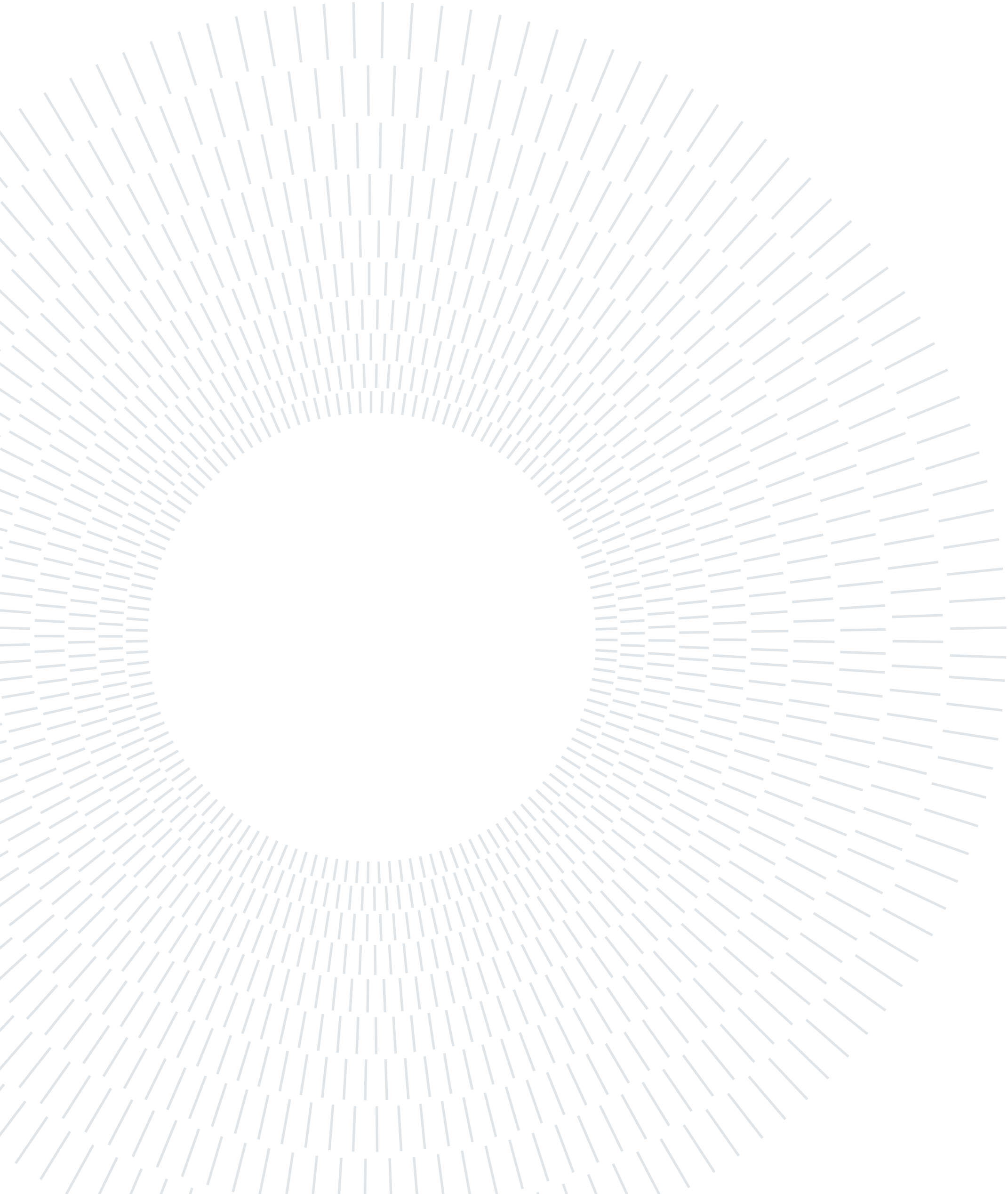
(b) Simulation B.

Figure 6.13: Simulation Environment: LEO, elliptical, set-7.

The combination of an under-actuated control system and the extreme difficulty intrinsic to the simulation conditions have made the **MPC** not attain the desired goals. The chaser spacecraft can be seen moving closer to the desired bounds, although working through actuator saturation, in Fig. 6.13b.

But nearing the end of the simulation, when approaching the perigee, the **MPC** is completely unable to control the motion as requested and leads to the chaser spacecraft drifting away due to the overpowering relative motion dynamics and the orbital J_2 perturbations.

It can be a good reminder to recall that the simulation conditions employed in this simulation is not physically possible and only acts as a worst case testing scenario. Hence, the **MPC**, designed under this thesis, is expected to perform much better in simulation conditions resembling real-life applications.



7 | Conclusion and Future Scope

In this dissertation the implementation of a model discovery based predictive controller was analysed. The main objective was to study the feasibility, scope and validity of the proposed control system. Through theoretical foundations in spaceflight mechanics, control theory and data-based numerical methods, a framework was developed and integrated to form the **MDPC**.

The validity of the solution was established through simulations of its accuracy in a two-body environment with J_2 perturbations. The feasibility and scope of the implemented control system was analysed through a simulation campaign with varying simulation environments and control system parameters. The primary findings of the campaign were discussed in the preceding sections.

This section accumulates concisely, the findings and their explanations, limitations of the solution and the future scope of improvement in the implemented solution.

7.1. Important Findings

The important findings of this dissertation will be discussed addressing the sub-system level findings and finally, the complete system level findings.

7.1.1. Model Discovery

The important findings pertaining to the standalone implementation of the model discovery framework can be listed as follows:

1. The resulting discovered model coefficients offer a high degree of insight into the actual local dynamics.
2. The discovered model is able to compensate, to a certain degree, for the presence of disturbance forces and accelerations.
3. The model discovery framework offers satisfactory accuracy in prediction outside the interval of measurement collection.

4. Stochastic methods employed ensure the model stability and robustness.
5. The constrained regression based method of model discovery is computationally more efficient than neural networks based training approaches.

7.1.2. Model Predictive Controller

The important findings pertaining to the standalone implementation of the model predictive controller can be listed as follows:

1. The **MPC** optimisation algorithm guarantees non-violation of constraints and bounds at every iteration.
2. The controller is highly responsive to predicted bounds violation.
3. The controller is capable of providing a sub-optimal but constructive control sequence in the case of non fulfilment of optimisation convergence at the global minima before the **MPC** sampling interval.

7.1.3. Complete Control System: MDPC

The important findings pertaining to the complete implementation of the model discovery and predictive controller can be listed as follows:

1. The **MDPC** parameters are adaptive, based on look-up tables which correlate the prediction error with the orbit locality and **MPC** horizons.
2. The **MDPC** is able to successfully provide effective control sequences under the presence of heavy perturbations and intensive non-linearity in the local dynamics.
3. The **MDPC** is able to predict the future trajectory while including the prospective control sequences within constraints and prioritise the maximum impulse deliverance at the earliest.

7.2. Solution Limitations

The main limitations of the complete implementation of the **MDPC** framework are primarily found in the scope of its application. They can be enumerated as follows:

1. The model discovery framework considers the coefficients of the system matrix to remain invariant within the linearisation interval.
2. The model discovery framework needs knowledge about the magnitude of the specific

orbital momentum to maintain its accuracy if the scale of the orbit alteration due to the manoeuvres are large.

3. The model discovery algorithm needs to be employed more frequently in orbital environments characterised by highly non-linear dynamics.
4. The controller prediction horizon is implicitly linked to the measurement interval utilised in the model discovery algorithm, in the context of the implemented solution.
5. The **MDPC** framework has no methods to mathematically account for significantly different gravitational fields to the idealised two-body problem such as gravitational fields of the asteroids.
6. The **MPC** is not modelled to account and compensate for actuator saturation in the control sequence optimisation.

7.3. Future Scope of Work

The understanding of the limitations of the implemented solution leads naturally to the possible branches of future work in this field. The main objective of the dissertation is satisfied through the implemented **MDPC** system, however it could be improved.

The main directions of improvement in the model discovery framework can be listed as:

1. Inclusion of system matrix coefficient variance within the linearisation interval and prediction horizon
2. Include mathematical formulation of perturbation accelerations into the state-space model and regression framework.
3. The current solution requires a very minimal ground-segment dependence to estimate the specific orbital momentum. Future work can focus on eradicating the necessity of ground-segment support for increased autonomy.
4. Account for actuator saturation into the **MPC** control sequence optimisation algorithm.
5. Include a parallel algorithm to perform large orbital scale manoeuvres to optimise the control effort by accounting for the actuation efficiencies at various locales in the orbits such as efficiency of orbit raising manoeuvres is maximal at the perigee.

7.4. Salient Features

Finally, concluding this chapter, we can list the important salient features of the proposed solution as follows:

1. The **MDPC** is effective in practically highly eccentric orbits of $e = 0.6$
2. The solution is valid for larger initial separation in order of a few hundred kilometres.
3. The range prediction algorithm is accurate and stable even in the presence of large perturbations and time-varying nonlinear dynamics.
4. The implemented solution is computationally efficient and simple.
5. The proposed solution is sparse in its actuation requirements i.e a single pair of thrusters aligned perpetually towards the target spacecraft.
6. Utilises a simple and robust optimisation algorithm characterised by convexity in the objective function enabling robust convergence and guarantees a global minima.
7. The **MDPC** can function with complete autonomy which enables impromptu manoeuvres to be conducted without ground-segment support.
8. The model discovery framework is faster than neural networks based learning methods while also being more transparent in their insights into the actual local dynamics.

Bibliography

[1] K. T. Alfriend and Y. Kashiwagi. “minimum-time orbital rendezvous between neighboring elliptic orbits, ” Journal of Optimization Theory and Applications. 4(4): 260–276, 1969.

[2] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. Nonlinear predictive control and moving horizon estimation — an introductory overview. In P. M. Frank, editor, *Advances in Control*, pages 391–449, London, 1999. Springer London. ISBN 978-1-4471-0853-5.

[3] F. Ankersen and N. Guidance. *Control and Relative Dynamics for Spacecraft Proximity Maneuvers.* PhD thesis, Aalborg University, Aalborg, Denmark, Dec. 2010.

[4] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw*, 2, 1989.

[5] A. A. Baranov. “an algorithm for calculating parameters of multi-orbit maneuvers in remote guidance. ” *Cosmic Research*, 28(1):61–67, 1990.

[6] P. W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks.* 01261 (4, June 2018.

[7] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev*, 57, 2015.

[8] L. S. Breger and J. P. How. *“J2-modified GVE-based MPC for formation flying spacecraft, ” in AIAA Guidance, Navigation, and Control Conference (GNC), (San Francisco, California, US).* Aug. 2005.

[9] C. Brezinski. *Computational Aspects of Linear Control.* Numerical Algorithms, 1. Springer US, 2002. ISBN 978-1-4020-0711-8. URL `https://books.google.it/books?id=YypLfQhopfQC`.

[10] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U. S. A*, 113, 2016.

[11] P. A. Capó-Lugo and P. M. Bainum. *Orbital Mechanics and Formation Flying: A Digital Control Perspective.* Woodhead Publishing in mechanical engineering, Woodhead Publishing Limited, 2011.

[12] T. E. Carter. “state transition matrices for terminal rendezvous studies: Brief survey and new example. ” *Journal of Guidance, Control, and Dynamics*, 21:148–155, Jan. 1998.

[13] W. H. Clohessy and P. S. Wiltshire. “terminal guidance system for satellite rendezvous, ” Journal of Aerospace Sciences. 27:653–658, Sept. 1960.

[14] H. D. Curtis. *Orbital Mechanics for Engineering Students.* Elsevier Aerospace Engineering Series, Butterworth-Heinemann, 2005.

[15] C. R. Cutler and B. L. Ramaker. Dynamic matrix control a computer control algorithm. *Joint Automatic Control Conference*, 17:72, 1980. doi: 10.1109/JACC.1980.4232009.

[16] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven. Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas*, 24:022310, 2017.

[17] W. Fehse. *Automated Rendezvous and Docking of Spacecraft.* Cambridge University Press, 2003.

[18] P. A. Felisiak. Control of spacecraft for rendezvous maneuver in an elliptical orbit. 2016.

[19] G. W. Hill. “researches in the lunar theory. ” *American Journal of Mathematics*, 1: 5–26, 1878.

[20] M. Hoffmann, C. Fröhner, and F. Noé. Reactive SINDy: Discovering governing reactions from concentration data. *J. Chem. Phys. 150*, 150:025101, 2019.

[21] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A*, 474:20180335, 2018.

[22] R. E. Kálmán. A new approach to linear filtering and prediction problems" transaction of the asme journal of basic. 1960.

[23] R. E. Kálmán. Contributions to the theory of optimal control. 1960.

[24] C. D. Karlgaard and F. H. Lutze. “second-order relative motion equations, ” Journal of Guidance, Control, and Dynamics. 26(1):41–49, 2003.

[25] Z. Lai and S. Nagarajaiah. Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Process*, 117, 2019.

[26] H. S. London. “second approximation to the solution of the rendezvous equations. ” *AIAA Journal*, 1(7):1691–1693, 1963.

[27] Y. Luo, J. Zhang, and G. Tang. Survey of orbital dynamics and control of space rendezvous. *Chinese Journal of Aeronautics*, 27(1):1–11, 2014. ISSN 1000-9361. doi: https://doi.org/10.1016/j.cja.2013.07.042. URL `https://www.sciencedirect.com/science/article/pii/S1000936113001787`.

[28] P. Marquis and J. P. Broustail. Smoc, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. *IFAC Proceedings Volumes*, 21:37–45, 1988.

[29] M. Okasha and B. Newman. *“Relative motion guidance, navigation and control for autonomous orbital rendezvous, ” in Proceedings of AIAA Guidance, Navigation and Control Conference.* 2011.

[30] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett. 120*, 120:024102, 2018.

[31] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv:1711*, page 10566, Nov. 2017.

[32] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Paper: Model predictive heuristic control. *Automatica*, 14(5):413–428, sep 1978. ISSN 0005-1098. doi: 10.1016/0005-1098(78)90001-8. URL `https://doi.org/10.1016/0005-1098(78)90001-8`.

[33] I. M. Ross. “linearized dynamic equations for spacecraft subject to j perturbations. ” *Journal of Guidance, Control, and Dynamics*, 26(4):657–659, 2003.

[34] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Sci. Adv*, 3, 2017.

[35] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proc. Natl. Acad. Sci. U. S. A*, 110, 2013.

[36] H. Schaub and J. L. Junkins. *Analytical Mechanics of Space Systems.* Reston, Virginia, US: AIAA Education Series, Oct. 2003.

[37] K. Schittkowski and C. Zillober. Nonlinear programming: Algorithms, software, and applications. volume 166, pages 73–107, 01 2006. ISBN 978-1-4020-7760-9. doi: 10.1007/0-387-23467-5_5.

[38] M. Schmidt and H. Lipson. *Distilling free-form natural laws from experimental data.* 2009.

[39] S. A. Schweighart and R. J. Sedwick. “high-fidelity linearized j model for satellite formation flight, ” Journal of Guidance, Control, and Dynamics. 25(6):1073–1080, 2002.

[40] R. E. Sherrill. *Dynamics and Control of Satellite Relative Motion in Elliptic Orbits using Lyapunov-Floquet Theory.* PhD thesis, Auburn University, Auburn, Alabama, US, May 2013.

[41] M. J. Sidi, S. Dynamics, and C. A. P. E. Approach. *Cambridge Aerospace Series, Cambridge University Press.* 1997.

[42] M. Sorokina, S. Sygletos, and S. Turitsyn. Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express*, 24:30433, 2016.

[43] J. Tschauner and P. Hempel. “rendezvous zu einem in elliptischer bahn umlaufenden ziel, ” Astronautica Acta. 11(2):104–109, 1965.

[44] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long-short term memory networks. *Proc. R. Soc. A*, 474:20170844, 2018.

[45] K. Yamanaka and F. Ankersen. “new state transition matrix for relative motion on an arbitrary elliptical orbit, ” Journal of Guidance, Control, and Dynamics. 25: 60–66, Jan. 2002.

[46] E. Yeung, S. Kundu, and N. Hodas. *“Learning deep neural network representations for Koopman operators of nonlinear dynamical systems” in 2019 American Control Conference (IEEE.* New York), 2019.

[47] Z. Zhang, L. Deng, J. Feng, L. Chang, D. Li, and Y. Qin. A survey of precision formation relative state measurement technology for distributed spacecraft. *Aerospace*, 9(7):362, 2022.

[48] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin. *A unified framework for sparse relaxed regularized regression: Sr3.* 2019.

A | Appendix A

Nonlinear Equations of Relative Motion: LVLH Frame

The relative motion in the **LVLH** frame can be described using exact non-linear equations of relative motion, also called **NERM**. The derivation of said equations are performed in a slightly altered manner to suit the necessities of the control system. The classical derivation of the same can be found in Ref. [36, 40].

Let us represent a new reference frame called **LVLH**. The diagrammatic representation of this reference frame is shown below:

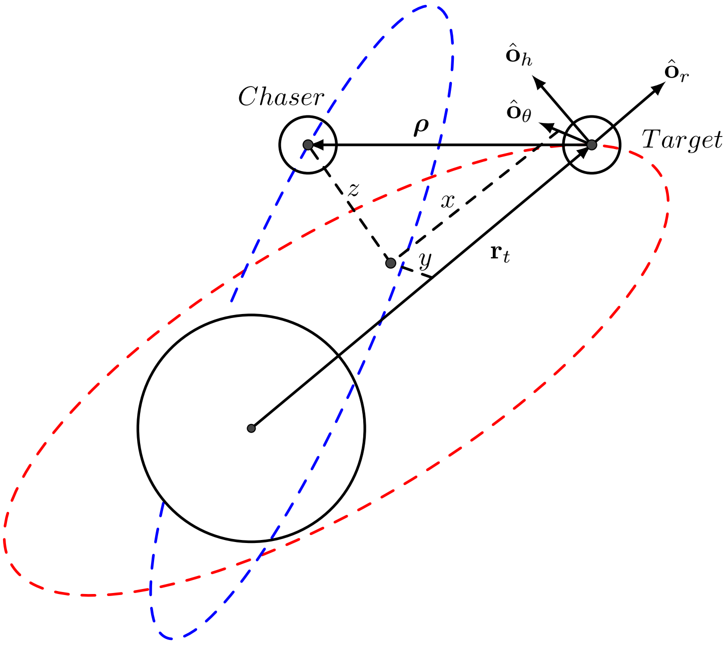


Figure A.1: Local-vertical local-horizon (LVLH) coordinate frame [18].

With reference to Fig. A.1, the position vector $\mathbf{r}_c$ for the chaser spacecraft can be written as follows:

$$\boldsymbol{\rho} = x\hat{\mathbf{o}}_r + y\hat{\mathbf{o}}_\theta + z\hat{\mathbf{o}}_h \tag{A.1}$$

or alternatively:

$$\mathbf{r}_c = \mathbf{r}_t + \boldsymbol{\rho} = (r_t + x)\,\hat{\mathbf{o}}_r + y\hat{\mathbf{o}}_\theta + z\hat{\mathbf{o}}_h \tag{A.2}$$

where the vector $\mathbf{r}_t$ denotes the position vector of target spacecraft corresponding with position vector $\mathbf{r}_c$ of the chaser spacecraft, both in the **LVLH** frame.

In the continuing analysis and derivations the term f is assumed to denote the true anomaly of the target spacecraft. We can write the specific orbital momentum of target spacecraft as:

$$h = r_t^2 \dot{f} \tag{A.3}$$

We can now impose the specific angular momentum of target spacecraft to be constant, assuming Keplerian orbital motion, resulting in:

$$\dot{h} = 0 = 2r_c\dot{r}_t\dot{f} + r_t^2\ddot{f} \tag{A.4}$$

by rearranging eq. A.4 we can demonstrate:

$$\ddot{f} = -2\frac{\dot{r}_t\dot{f}}{r_t} \tag{A.5}$$

To express the rate of change of true anomaly of the target spacecraft in terms of orbital semi-latus rectum p instead of $\boldsymbol{r}$, we employ $h^2 = \mu p$ and obtain:

$$\dot{f} = \sqrt{\frac{\mu p}{r_t^4}} \tag{A.6}$$

We can analytically see that the angular velocity vector ω of the target spacecraft is aligned with its specific angular momentum vector $\boldsymbol{h}$. Hence, we can express it as:

$$\omega = \dot{f}\hat{\mathbf{o}}_h = \frac{h}{r_t^2}\hat{\mathbf{o}}_h \tag{A.7}$$

From eq. A.5, we write the equation for angular acceleration of the target spacecraft as:

$$\dot{\omega} = \ddot{f}\hat{\mathbf{o}}_h = -2\frac{\dot{r}_t\dot{f}}{r_t}\hat{\mathbf{o}}_h \tag{A.8}$$

Keeping in mind the formulation of the relative angular acceleration of the target spacecraft, the eq. A.2 can be differentiated twice with respect to time as follows:

$$\ddot{\mathbf{r}}_c = \ddot{\mathbf{r}}_t + \dot{\omega} \times \rho + \omega \times (\omega \times \rho) + 2\omega \times \dot{\rho} + \ddot{\rho} \tag{A.9}$$

The target spacecraft position written in **LVLH** frame can be formulated as:

$$\mathbf{r}_t = r_t\hat{\mathbf{o}}_r \tag{A.10}$$

Now, we can combine the eq. A.1, eq. A.7 and eq. A.8 by substituting them in eq. A.9. By using eq. A.10, we can now write:

$$\begin{aligned}
\ddot{\mathbf{r}}_c = \quad & \ddot{r}_t\hat{\mathbf{o}}_r + \ddot{f}\hat{\mathbf{o}}_h \times (x\hat{\mathbf{o}}_r + y\hat{\mathbf{o}}_\theta + z\hat{\mathbf{o}}_h) + \\
& \dot{f}\hat{\mathbf{o}}_h \times \left(\dot{f}\hat{\mathbf{o}}_h \times (x\hat{\mathbf{o}}_r + y\hat{\mathbf{o}}_\theta + z\hat{\mathbf{o}}_h)\right) + 2\dot{f}\hat{\mathbf{o}}_h \times (\dot{x}\hat{\mathbf{o}}_r + \dot{y}\hat{\mathbf{o}}_\theta + \dot{z}\hat{\mathbf{o}}_h) + \\
& (\ddot{x}\hat{\mathbf{o}}_r + \ddot{y}\hat{\mathbf{o}}_\theta + \ddot{z}\hat{\mathbf{o}}_h) \\
= \quad & \left(\ddot{r}_t - \ddot{f}y - 2\dot{y}\dot{f} + \ddot{x}\right)\hat{\mathbf{o}}_r + \\
& \left(\ddot{f}x - \dot{f}^2y + 2\dot{x}\dot{f} + \ddot{y}\right)\hat{\mathbf{o}}_\theta + \\
& \ddot{z}\hat{\mathbf{o}}_h
\end{aligned} \tag{A.11}$$

Furthermore, we can apply Newtonian two-body equation, ($\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\ \boldsymbol{r}$), to describe the Keplerian orbital motion under purely gravitational interactions. By substituting eq. A.10 into the Newtonian two-body equation, we can find the relative acceleration vector of the target spacecraft as:

$$\ddot{\mathbf{r}}_t = -\frac{\mu}{r_t^3}\mathbf{r}_t = -\frac{\mu}{r_t^2}\hat{\mathbf{o}}_r \tag{A.12}$$

Hence, we can rewrite the eq. A.11 by substituting the scalar component of eq. A.12:

$$\begin{aligned}\ddot{\mathbf{r}}_c = &\left(-\frac{\mu}{r_t^2} + 2\frac{\dot{r}_t}{r_t}\dot{f}y - \dot{f}^2x + \ddot{x}1\right)\hat{\mathbf{o}}_r + \\ &\left(-2\frac{\dot{r}_t}{r_t}\dot{f}x - \dot{f}^2y + 2\dot{x}\dot{f} + \ddot{y}\right)\hat{\mathbf{o}}_\theta + \ddot{z}\hat{\mathbf{o}}_h\end{aligned} \tag{A.13}$$

To move away from the idealised model of two-body problem, let us introduce additional forces encountered by the spacecrafts. Maintaining the classical terminology, the gravitational two-body forces will be assumed as discussed previously. Addition of two forces namely, control force $\boldsymbol{u}$ and random disturbance force denoted as $\boldsymbol{u}_d$ are introduced. The nature of disturbance force is defined such that it accounts for all forces that are uncontrolled such as spacecraft actuation error, drag, etc.

The definition adopted for control force is:

$$\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T \tag{A.14}$$

where the scalar components are aligned with the **LVLH** basis unit vectors along the radial direction, along-track direction and out-of-plane direction in the order u_x, u_y and u_z respectively. The overall disturbance force encountered by the chaser spacecraft can be expressed in terms of the disturbance acceleration $\boldsymbol{a}_d$:

$$\mathbf{u}_d = \begin{bmatrix} u_{dx} & u_{dy} & u_{dz} \end{bmatrix}^T = m_c\mathbf{a}_d = \mathbf{u} + \mathbf{u}_p \tag{A.15}$$

where m_c represents the chaser spacecraft mass and $\boldsymbol{u}_p$ represents the perturbation force (eg. J_2 perturbation) vector defined in **LVLH** frame. Accounting for disturbance, to find the acceleration vector for the chaser spacecraft, we can employ:

$$\ddot{\mathbf{r}} = \ddot{\mathbf{r}}_2 - \ddot{\mathbf{r}}_1 = -\frac{\mu}{r^3}\,\mathbf{r} + \mathbf{a}_d \tag{A.16}$$

$$\ddot{\mathbf{r}}_c = -\frac{\mu}{r_c^3}\mathbf{r}_c + \mathbf{a}_d = -\frac{\mu}{r_c^3}\begin{bmatrix} r_t + x \\ y \\ z \end{bmatrix} + \frac{1}{m_d}\begin{bmatrix} u_{dx} \\ u_{dy} \\ u_{dz} \end{bmatrix} \tag{A.17}$$

The chaser spacecraft's current radius, in presence of disturbances, can be written as:

$$r_c = \sqrt{(r_t + x)^2 + y^2 + z^2} \tag{A.18}$$

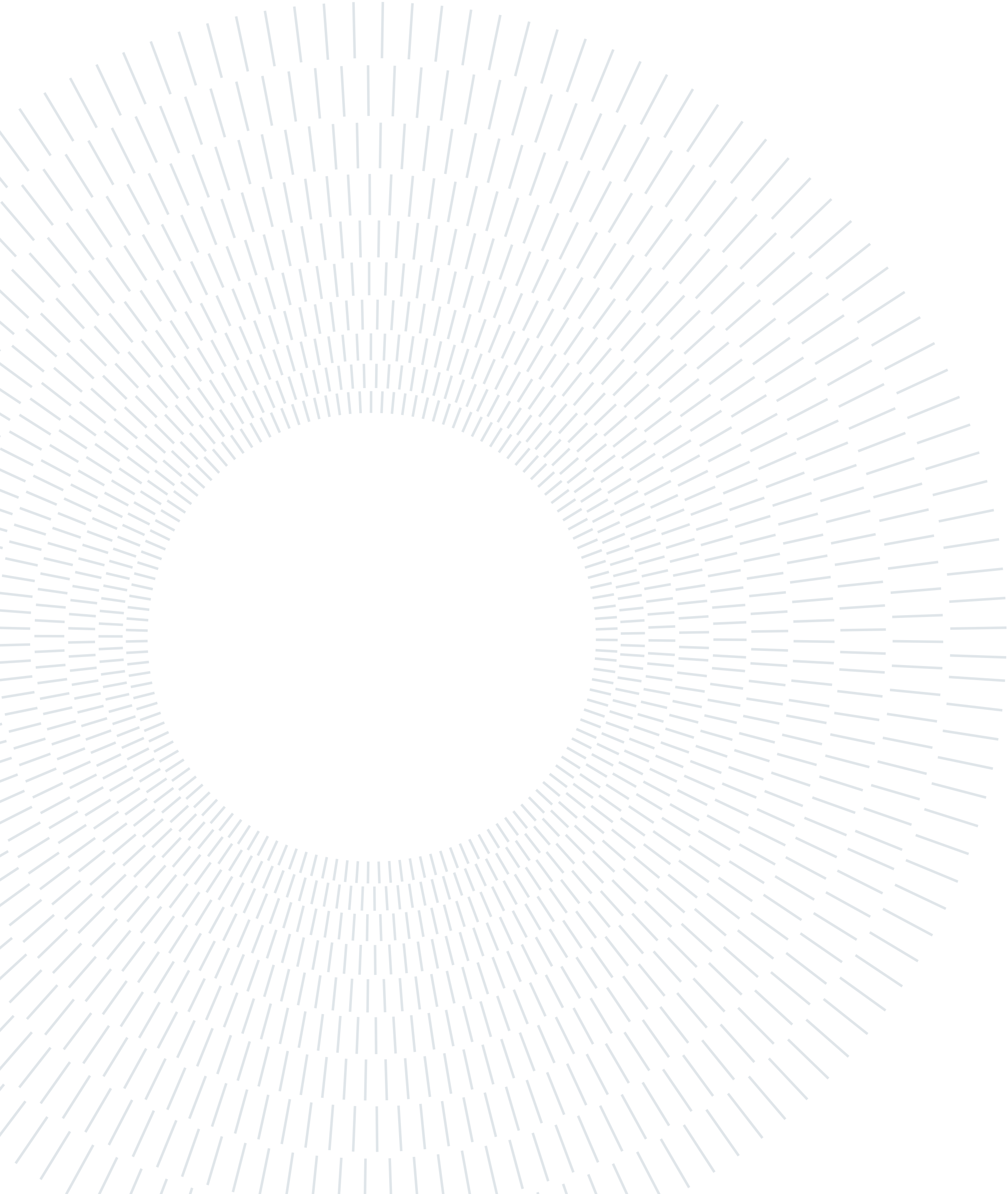
By combining eq. A.13 and eq. A.17, we can find the formulation of the **NERM**, described previously as:

$$\ddot{x} - 2\dot{f}\left(\dot{y} - y\frac{\dot{r}_t}{r_t}\right) - x\dot{f}^2 - \frac{\mu}{r_t^2} = -\frac{\mu}{r_c^3}\left(r_t + x\right) + \frac{u_{dx}}{m_d} \tag{A.19}$$

$$\ddot{y} - 2\dot{f}\left(\dot{x} - x\frac{\dot{r}_t}{r_t}\right) - y\dot{f}^2 = -\frac{\mu}{r_c^3}y + \frac{u_{dy}}{m_d} \tag{A.20}$$

$$\ddot{z} = -\frac{\mu}{r_c^3}z + \frac{u_{dz}}{m_d} \tag{A.21}$$

The eq. A.19, eq. A.20 and eq. A.21 form the set of exact non-linear equations describing the chaser spacecraft motion relative to the target spacecraft.



B | Appendix B

J2 Perturbation

The equatorial radius of the Earth is approximately 20 kilometres greater than the polar radius. The internal structure of the Earth is not uniform and is characterised by clusters of mantle and irregular continental mass distributions and tidal effects. This results in a non-uniform gravitational field which is modelled as a perturbation called J_2 perturbation.

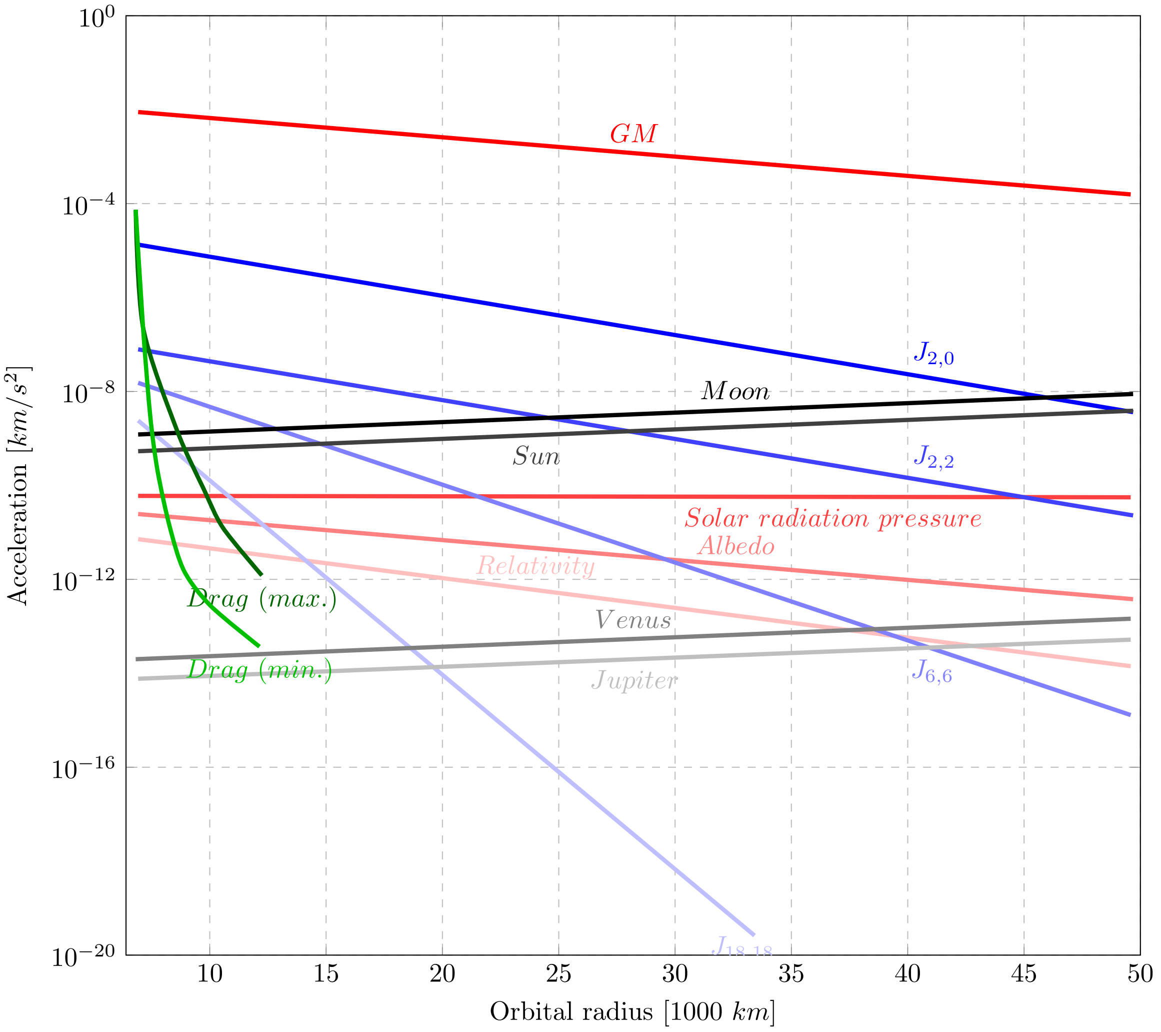


Figure B.1: Orbital perturbations for Earth orbit (reproduced from [9]).

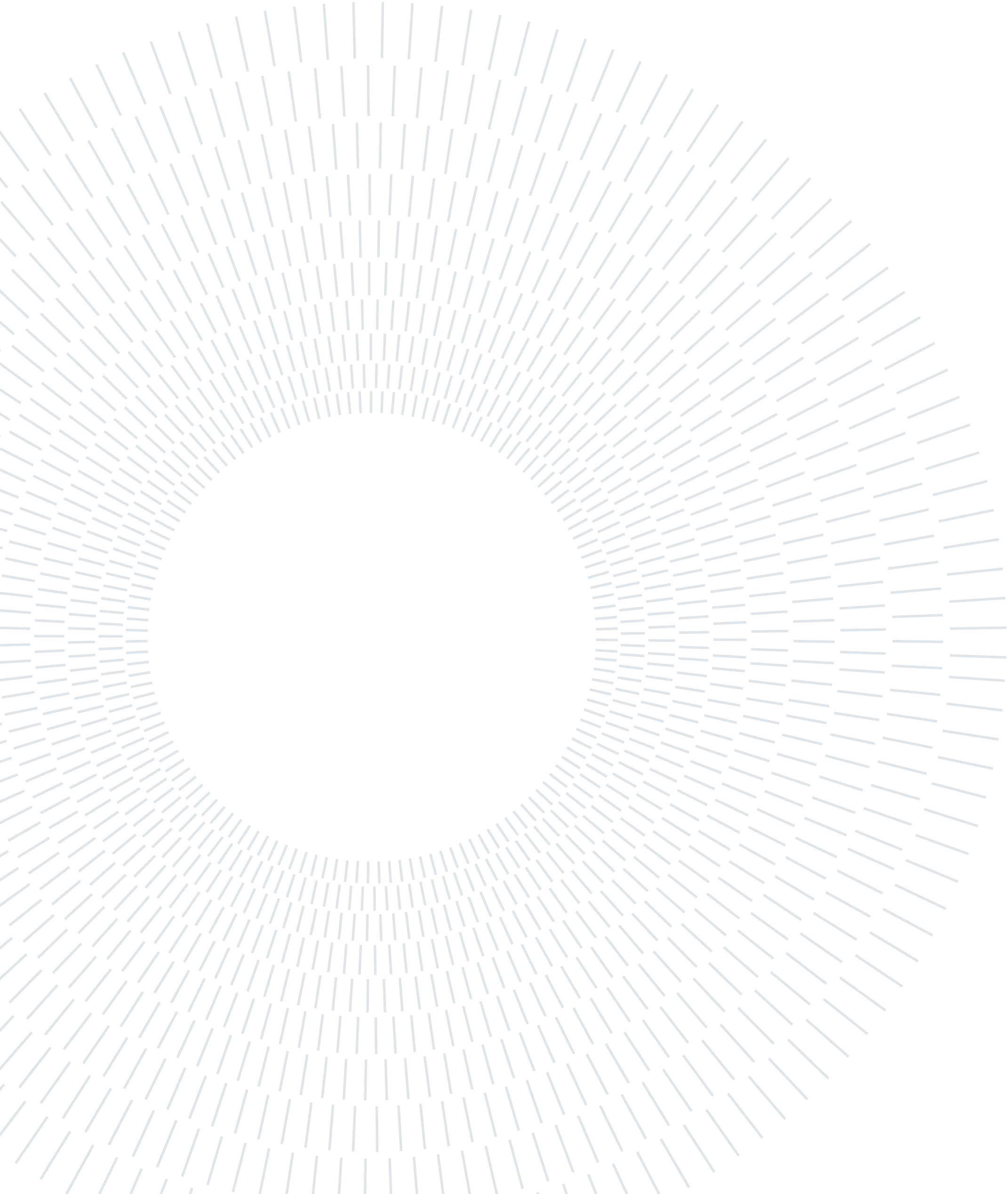
The mathematical derivation of the acceleration due to J_2 perturbation is beyond the scope of this dissertation. The equation for the same is borrowed from Ref. [14] and reported below:

$$\boldsymbol{a}_{J_2}^{ECI} = \frac{3}{2}\frac{J_2 \mu R_e^2}{r^4}\left[\frac{x}{r}\left(5\frac{z^2}{r^2} - 1\right)\hat{\boldsymbol{i}}_x + \frac{y}{r}\left(5\frac{z^2}{r^2} - 1\right)\hat{\boldsymbol{i}}_y + \frac{z}{r}\left(5\frac{z^2}{r^2} - 3\right)\hat{\boldsymbol{i}}_z\right] \tag{B.1}$$

where J_2 is the constant of perturbation magnitude and R_e denotes the radius of the Earth. This enables us model the J_2 perturbations into the "ground-truth" simulations.

List of Figures

List of Tables



List of Abbreviations and Symbols

List of Abbreviations

2BP	Two Body Problem
BC	Boundary Conditions
CReLU	Continuous ReLU
CSR3	Constrained Sparse Relaxed Regularised
DMC	Dynamic Matrix Control
DoF	Degrees of Freedom
ECEI	Earth Centred Equatorial Inertial
GNSS	Global Navigation Satellite System
GVE	Gauss' Variational Equations
HCW	Hill-Clohessy-Wiltshire
IC	Initial Condition
IDCOM	IDentification and COMmand
IMU	Inertial Measurement Unit
IVP	Initial Value Problem
L.H.S	Left Hand Side
LERM	Linearised Equations of Relative Motion
LQR	Linear Quadratic Regulator
LVLH	Local Vertical Local Horizontal
MDPC	Model Discovery and Predictive Controller

MDPC Model Discovery and Predictive Controller

NERM Non-linear Equations of Relative Motion

ODE Ordinary Differential Equation

PID Proportional,Integral and Derivative

PRISMA Prototype Research Instruments and Space Mission Technology Advancement

QP Quadratic Programming

R.H.S Right Hand Side

ReLU Rectified Linear Activation Function

RK4 Runge-Kutta Order 4

RNMSE Root of Normalised Mean-Squared Error

SINDy Sparse Identification of Nonlinear Dynamics

SMA Semi Major Axis

SQP Sequential Quadratic Programming

SR3 Sparse Relaxed Regularised

STLSQ Sequentially Thresholded Least Squares

STM State Transition Matrix

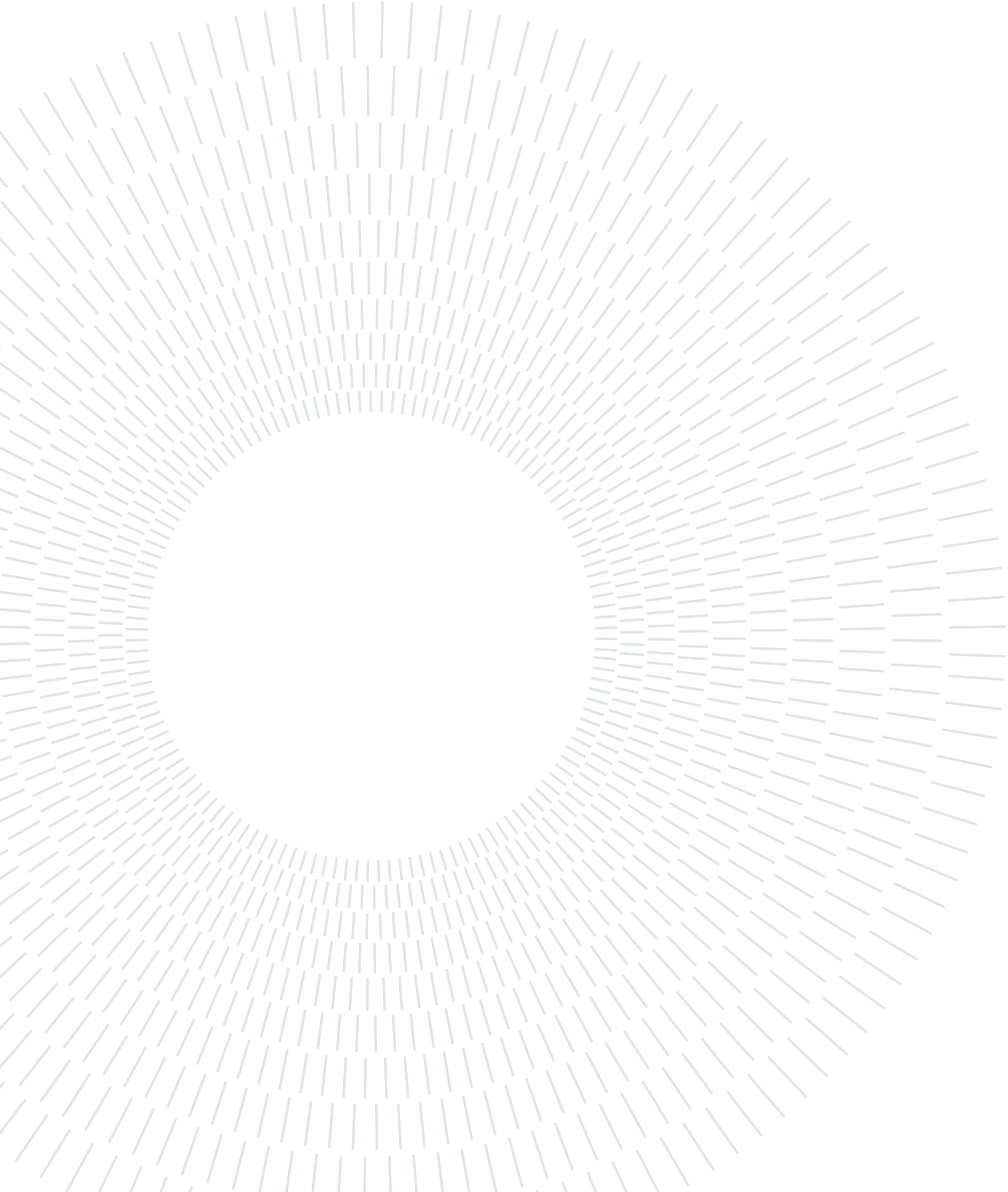
List of Symbols

$\boldsymbol{\eta}$	Coefficient specific thresholds matrix: SINDy
$\boldsymbol{\Theta}$	Coordinates: SINDy
$\boldsymbol{\Upsilon}$	Perturbation acceleration state vector: SINDy
$\boldsymbol{\Xi}$	Coefficients: SINDy
$\boldsymbol{\Xi}_{eff}$	Effective coefficients matrix: SINDy
ϵ	Constraint tolerance: SINDy
$_0$	First estimated guess coefficient matrix: SINDy
λ	Regression thresholding hyper-parameter

T	Thresholding factor: SINDy
μ	Standard gravitational parameter
$\odot$	Element-wise multiplication operator
ω	Target spacecraft true-anomlay rate
ω_Q	Angular frequency of sinusoidal thrust input
ω_{orb}	Angular frequency of the orbital motion
ρ_{max}	Maximum range bound
ρ_{min}	Minimum range bound
$\mathbf{A}$	Chaser spacecraft system matrix
$\mathbf{a}_d$	External acceleration
$\mathbf{A}_p$	Chaser spacecraft propellant system matrix
$\mathbf{A}_{NL}$	Chaser spacecraft nonlinear system matrix
$\mathbf{B}$	Chaser spacecraft input matrix
$\mathbf{B}_p$	Chaser spacecraft propellant input matrix
$\mathbf{C}_i$	Constraints coefficients: SINDy
$\mathbf{C}_{eq}$	Equivalent constraint coefficients: SINDy
$\mathbf{d}_i$	Constraints equalities: SINDy
$\mathbf{d}_{eq}$	Equivalent constraint equalities: SINDy
$\mathbf{h}$	Specific angular momentum
$\mathbf{r}$	Orbital position vector
$\mathbf{u}$	Control force
$\mathbf{u}_d$	Disturbance force
$\mathbf{u}_p$	Perturbation force vector: Hill's frame
$\mathbf{u}_{k=1}$	Control input vector at first sample instance within MPC control horizon
$\mathbf{V}$	Chaser spacecraft nonlinear coefficient matrix
$\mathbf{W}$	Relaxed coefficients: SINDy

$\mathbf{x}$	Chaser spacecraft relative motion state vector
$\mathbf{x}_p$	Chaser spacecraft propellant state vector
a	Initial thrust profile magnitude
a	Semi-major axis
b	Thrust profile exponential decay factor
$\mathrm{C}_R(\rho)$	Range cost function
$\mathrm{C}_U(t)$	Control cost function
C_U	Control cost
d_{max}	Maximum range violation cost slope scaling factor
d_{min}	Minimum range violation cost slope scaling factor
e	Orbit eccentricity
$\mathrm{E}_{MSE_{disc.}}$	Model discovery range propagation mean squared error
$\mathrm{E}_{MSE_{pred.}}$	Model range prediction mean squared error
$\mathrm{E}_{RNMSE,\ n}$	Root normalised mean squared error
f	True anomaly
F_{thrust}	Chaser spacecraft actuation force vecctor: Hill's frame
g_0	Acceleration due to gravity at sea-level
G_{rel}	Cost relative gain
I_{sp}	Chaser spacecraft effective actuator specific impulse
J_2	J_2 perturbation constant
J_2	J_2 perturbation constant
L0	L_0 Norm
L1	L_1 norm
L2	L_2 norm
m_c	Chaser spacecraft mass
m_{dry}	Chaser spacecraft dry mass

m_{p0}	Chaser spacecraft initial propellant mass
n	Number of data-sets for simultaneous model discovery
N_c	Number of constraints: SINDy
N_f	Number of features: SINDy
N_t	Number of targets: SINDy
N_{CH}	Number of sampling instances in control horizon
N_{MR}	Number of times the discovered model is reused before new model discovery
p	Semi-latus rectum
T_s	Sensor measurement sampling time
T_{CH}	MPC Control horizon
T_{lin}	Linearisation interval of the discovered model
T_{PH}	MPC Prediciton horizon
$T_{s,\ eff}$	Effective sampling time of distributed measurement data-set
v	Arbitrary vector
v_i	Arbitrary vector components
x	Chaser spacecraft relative position along Target radial vector: Hill's frame
y	Chaser spacecraft relative position along Target along-track vector: Hill's frame
z	Chaser spacecraft relative position along Target normal vector: Hill's frame
r	Orbit radius



Acknowledgements

At the very outset, I would like to attribute the successful completion of this dissertation to the timely and prudent guidance of my advisor, professor Mauro Massari. His frequent pruning of my approach acted as the much needed blinkers during the course of the research. The patience and tolerance he exhibited towards my naivety cannot be overstated.

It would be amiss to omit the role of my professors, who have imparted curiosity along with the rigour for scientific inquiry. Without their diligent efforts, it would not be possible to undertake this journey. I would like to thank the institutional environment at Politecnico di Milano for nurturing and stimulating my academic efforts. It is my privilege to be accepted as a part of a historical and prominent institution.

A great thanks to all my teachers from school, who have shaped my mind at the most momentous stages of my life. I am most obliged to you for your gift of knowledge.

I am extremely grateful to my family for supporting my pursuit of education and absorbing my failures. They have sacrificed, so that I should not need to. Thank you!