Design and Application of Thrust Manoeuvres in a Constrained Spacecraft Rendezvous Context Using Optimal Control Techniques

Afonso Duarte Amaral Cardoso Botelho

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. João Manuel Lage de Miranda Lemos
Eng. Baltazar Parreira

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Eng. Baltazar Parreira
Member of the Committee: Prof. Rita Maria Mendes de Almeida Correia da Cunha

November 2019

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First, I express my deep gratitude to my supervisors for their continuous support and for motivating me to overcome myself. Namely Professor João Miranda Lemos, for his vast knowledge, wisdom and belief in me, and Baltazar Parreira, for his practical perspective, critical thinking and readiness to help. I would also like to thank Paulo Rosa, who granted me the opportunity of working on my dream Master Thesis and who, despite not being an official supervisor, was always present and provided me with invaluable advice and feedback.

Secondly, thank you to Deimos for providing me with an everyday workspace, and to my colleagues there for treating me like a member of that great team.

Last but not least, to my loving Parents, Grandparents, my brothers Martim, Tomás and Diogo, and my great friends, for all their love and support and for making me who I am today.

Part of this work was funded by project HARMONY, Distributed Optimal Control for Cyber-Physical Systems Applications through POR-Lisboa under LISBOA-01-0145–FEDER-031411 and by FCT through INESC-ID plurianual funds under contract UID/CEC/50021/2019. DEIMOS Engenharia provided the working framework for Afonso Botelho.

Resumo

É desenvolvido um algoritmo de Controlo Preditivo baseado em Modelo (MPC) para a execução de manobras de rendezvous em órbitas elíticas. O algoritmo de MPC calcula uma trajetória entre um estado inicial e desejado, minimizando o combustível e tendo em conta restrições de impulso máximo e de segurança de colisão passiva, mantendo uma implementação computacionalmente viável de implementar em tempo-real. O modelo de predição do MPC é baseado nas equações de Yamanaka-Ankersen e na solução particular de Ankersen do tipo “zero-order hold”.

O MPC de Horizonte-Finito é a formulação padrão para este problema, permitindo gerar trajetórias de rendezvous com combustível mínimo através de programação linear, dada a duração da manobra. A formulação de Horizonte Variável permite também optimizar a duração da manobra, e os dois métodos são comparados relativamente ao desempenho e carga computational. Demonstra-se também que estas abordagens são superiores às manobras de dois-impulsos tipicamente utilizadas em algoritmos de navegação para rendezvous.

A formulação padrão é expandida com o desenvolvimento de duas abordagens para a definição de restrições de segurança de colisão passiva, que são tipicamante não-convexas, através de restrições lineares. Contribui-se também com duas novas técnicas de robustez que melhoram o desempenho do controlador na presença de perturbações, mantendo a complexidade computacional. Por último, o algoritmo de MPC é aplicado no cenário da experiência de rendezvous da missão PROBA-3, na qual os satélites se encontram numa órbita altamente elítica.

Palavras-chave: Controlo Preditivo, Dinâmica Orbital Relativa, Rendezvous de Combustível-Óptimo, Segurança Passiva, Rendezvous Robusto

Abstract

The design of a Model Predictive Control (MPC) algorithm for performing orbital rendezvous manoeuvres in elliptical orbits is addressed. The MPC algorithm computes a trajectory between an initial and desired states, minimizing fuel and taking into account limited thrust authority and passive collision safety constraints, while ensuring a computationally feasible real-time implementation. The MPC prediction model is based on the Yamanaka-Ankersen equations and Ankersen's zero-order hold particular solution.

Finite-Horizon MPC is the standard formulation for this application, allowing for generating fuel-optimal rendezvous trajectories via linear programming given a pre-defined manoeuvre duration. The Variable-Horizon MPC framework also allows for optimizing the manoeuvre duration, and the two methods are compared regarding performance and computational complexity. We also demonstrate that these methods are superior to the two-impulse transfers typically used in rendezvous guidance algorithms.

We extend the standard framework by presenting two new approaches for formulating passive collision avoidance constraints, which are naturally non-convex, as linear constraints. We also contribute with two new robustness techniques which improve the controller performance in the presence of disturbances while maintaining the computational complexity. Finally, the MPC algorithm is applied to the scenario of the PROBA-3 rendezvous experiment, in which the spacecraft lie in a highly elliptical orbit.

Keywords: Model Predictive Control, Relative Orbital Dynamics, Fuel-Optimal Rendezvous, Passive Safety, Robust Rendezvous

Contents

List of Tables

List of Figures

Glossary

FH-MPC	Fixed-Horizon Model Predictive Control
GNC	Guidance, Navigation and Control
LERM	Linearised Equations of Relative Motion
LMPC	Linear Model Predictive Control
LP	Linear Program
LTV	Linear Time-Variant system
LVLH	Local-Vertical/Local-Horizontal frame
MILP	Mixed-Integer Linear Program
MPC	Model Predictive Control
NLP	Nonlinear Program
NMPC	Nonlinear Model Predictive Control
OAILP	Obstacle Avoidance with Iterative Linear Programming
OAONP	Obstacle Avoidance with Offline Nonlinear Programming
PWA	Piecewise Affine
PWM	Pulse-Width Modulation
QP	Quadratic Program
RVX	(PROBA-3) Rendezvous Experiment
ZOH	Zero-Order Hold discretization
mpQP	Multi-Parametric Quadratic Program

Chapter 1

Introduction

Orbital rendezvous is a highly useful procedure in which two separate spacecraft meet at the same orbit, as illustrated in figure 1.1, therefore approximately matching their orbital velocity and position [1]. Such manoeuvres allowed for the feasibility of the Apollo moon landing missions, with the rendezvous of the Lunar Excursion Module with the Command Module in lunar orbit, and for the construction and resupply of modular space stations, such as Mir and the International Space Station. Other applications include, for example, the exploration of smaller celestial bodies, such as asteroids, comets and small moons, the in-orbit servicing of satellites, for instance the multiple repair missions to the Hubble Space Telescope, or the active removal of space debris. Often the rendezvous process is followed by a docking or berthing procedure, which results in the physical connection of the two spacecraft, although this is not addressed in this work.

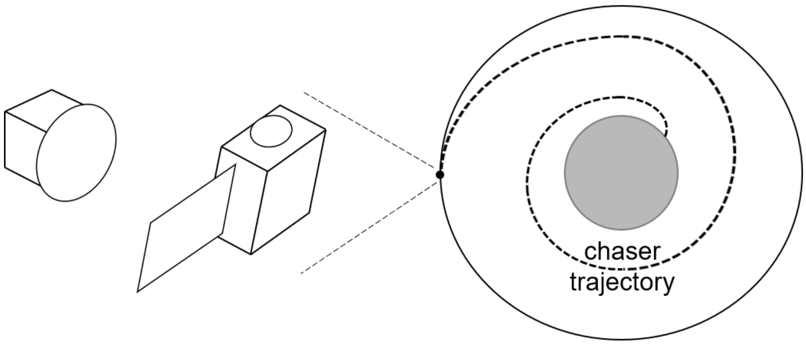


Figure 1.1: Illustration of the orbital rendezvous manoeuvre.

The first attempt at a rendezvous was performed in the Gemini 4 manned mission in 1965, which was unsuccessful due to the method of approach being simply ”point-and-shoot”, resulting in a further separation of the spacecraft. This revealed the challenge in performing a space rendezvous, and proved that the relative orbital dynamics involving the two spacecraft must be taken into consideration. Since then, rendezvous missions have been performed successfully hundreds of times, both by manned and unmanned spacecraft, and using various different control approaches. In this context, we consider in this work the use of Model Predictive Control (MPC) [2] for performing rendezvous manoeuvres, which is a widely successful optimal control strategy that naturally considers the system dynamics and can handle various different operational constraints. The use of MPC for this purpose can grant more autonomy to the spacecraft and increase the optimality of the approach trajectories, when compared to the traditional techniques.

The literature for MPC applied to rendezvous is now quite considerable, and this remains an active area of research. Despite these facts, MPC has been tested in real spaceflight only once, to the best of the author's knowledge, by the PRISMA mission [3]. Although it was in a formation flying context, not rendezvous, the underlying principles are identical. The main difficulty with the use of MPC for a real rendezvous mission is that it requires a considerable online computational effort, which can prove to be a challenge given the typically limited computing power available onboard. Furthermore, there is not yet a standard approach for robustness in face of all disturbances possibly interfering with a rendezvous mission which is both feasible to implement in real-time and maintains good operational performance, and thus more research into this topic is required.

1.1 Motivation

The motivation for this work is the PROBA-3 mission by the European Space Agency, scheduled for launch in 2021, in which two satellites will be launched together into a highly elliptical Earth orbit to test new formation flying technologies. To this end, one of the spacecraft, designated Occulter Spacecraft, will eclipse the Sun for the other, the Chronograph Spacecraft, which flies at a distance and carries a solar telescope for observation of the Sun's corona, as illustrated in figure 1.2. Uninterrupted observations are designed to last up to six hours, which requires remarkable precision in the relative position and velocity of the two spacecraft.

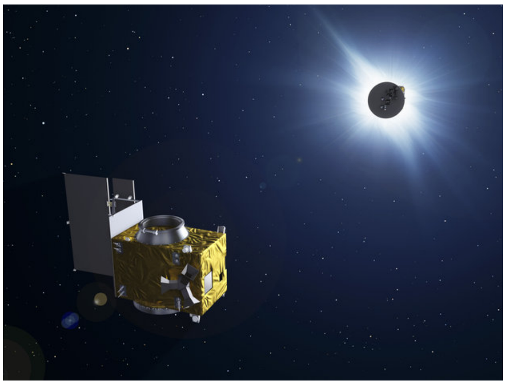


Figure 1.2: Illustration of the PROBA-3 mission. Taken from the ESA website.

A secondary objective of this mission is to perform a rendezvous experiment (RVX), which is led by DEIMOS [4]. The spacecraft will be separated by just over one kilometre, and a series of manoeuvres will be executed which will be the first of its kind in such a highly elliptical orbit, until the spacecraft are brought together again. Thus, this work was motivated as a feasibility study for DEIMOS in the use of MPC for the guidance and control systems in the PROBA-3 RVX.

1.2 Problem Formulation

A rendezvous mission generally adheres to the following sequence of events: launch, phasing, far-range rendezvous, close-range rendezvous and mating [1]. The launch phase ends with orbital insertion, nearly coplanar with the target orbit and typically at a lower altitude and behind the target, and is completely out of the scope of this work. Phasing consists of small corrections to the orbit parameters, and of passive waiting that takes advantage of the different orbital periods, allowing for the launched spacecraft to catch up with the target. This phase can last a few days and does not require great precision, such that correction manoeuvres are performed in open-loop without the need for the use of MPC. The rendezvous process itself then starts with the far-range phase, when relative navigation is possible, typically at a few tens of kilometres. This phase ends and close-range rendezvous begins when the relative distance requires safety-critical manoeuvres, typically at a few kilometres. Thus, it is these two phases that can benefit from the use of MPC to perform the approach manoeuvres, with the latter phase being the focus of this work given the specifications of the PROBA-3 RVX. As mentioned before, the mating phase (docking or berthing) is also out of the scope of this work.

An onboard automatic control system for a rendezvous mission contains three sub-systems tasked with the execution of thrust manoeuvres: guidance, navigation and control (GNC) [1]. The guidance system generates the reference trajectory and spacecraft attitude; navigation provides state measurements and estimates; control commands the force and torque necessary to drive the spacecraft to the desired state. MPC can simultaneously handle both guidance and control functions, while navigation is not considered in this work. Furthermore, because the translational and attitude control are typically decoupled in the far and close-range rendezvous phases [1], only translational control is dealt with, with attitude control not being addressed here. Finally, note that while all real rendezvous missions have been performed in a circular or near-circular target orbit, the PROBA-3 spacecraft will be in a highly elliptical orbit, which implies an increased difficulty since the dynamics become time-varying and more complex.

1.3 State-of-the-Art

This work covers several different research areas, and thus we will address the state-of-the-art for these separately. MPC was first introduced in the 1970's and is now a very mature framework, with an extensive theoretical basis [2] and a vast history of successful applications, mostly in the process industry [5]. It remains an active area of research, with recent work being dedicated to the application of MPC to specific problems, for example the rendezvous scenario [6]. Research is also devoted into improving the real-time feasibility of MPC, with the design of new optimization algorithms that exploit the MPC problem structure, for example [7–9], and with the further development of the popular Explicit MPC framework [10]. New sub-fields of MPC have also emerged in recent years, such as Distributed MPC, Stochastic MPC, or Neural Network MPC, among many others.

Concerning relative orbital mechanics, although the nonlinear dynamics can be easily derived from

Newton's laws, these equations do not have a closed-form solution, which limits their usefulness. Thus, research related to this topic, which is still active today, is dedicated to determining approximated dynamics with a closed-form solution. A set of linearised equations for the relative motion represented in a local non-inertial frame of reference and for a circular orbit was first derived by Hill in 1878 [11], in his study of the perturbations in the motions of the Moon. They were first applied and solved in the context of orbital rendezvous in 1959, most famously by Clohessy and Wiltshire [12], although this solution is only accurate for circular or near-circular orbits. The equations were extended to an elliptic orbit by De Vries in 1963 [13], and simplified via a change of the independent variable to the true anomaly. They were then solved and applied to spacecraft rendezvous in elliptic orbits by Tschauner and Hempel in 1965 [14], after whom the simplified equations became known. In 1998, Carter presented a simpler solution to the Tschauner-Hempel equations in the form of a state transition matrix [15], which is valid for any orbit eccentricity. Later, in 2002, Yamanaka and Ankersen introduced a simpler state transition matrix [16], although it is only valid for circular or elliptical orbits. Ankersen later complemented this solution by including a forced regime [17].

Although research to determine different solutions to the Tschauner-Hempel equations and other approximated models continues, the Yamanaka-Ankersen state transition matrix is considered to be the state-of-the-art solution for use in the design of rendezvous missions in elliptic orbits. Another common approach is to use a linearized model based on Gauss's Variational Equations [18], where the state vector is the Keplerian orbital elements instead of Cartesian coordinates. The resulting linear equations easily allow for the inclusion of the effect of the geopotential anomaly, as well as remaining accurate for larger relative distances than the other approach.

One of the first applications of MPC to the rendezvous problem was by Richards and How [19], where the basic formulations employed by most of the literature that followed were presented, namely Fixed-Horizon MPC and Variable-Horizon MPC, although the optimization formulations had been introduced earlier but not in an MPC context. These formulations explicitly minimize the fuel for the rendezvous manoeuvres and, in the case of the former, express the optimization problem as a linear program, which allows for feasible online computation. Thus, current research is mostly dedicated to extending these formulations, for example to grant robustness in face of the many disturbances interfering in a rendezvous mission, while ensuring convergence, constraint satisfaction and performance. The topic of robust MPC techniques for rendezvous is approached in Chapter 4 of this work .

1.4 Contributions

This work features a basic and easy-to-follow introduction to general MPC theory, which can serve as a practical tutorial for those uninitiated on this topic. It also contains a simple introduction to relative orbital mechanics, required to fully understand the rendezvous dynamics and manoeuvres.

Our contributions to MPC applied to rendezvous include the application of the Ankersen zero-order-hold particular solution [17], which provides a more realistic thrust profile than the commonly used impulsive discretization, but seems to have gone unnoticed in the literature. In this work we also consider rendezvous manoeuvres in elliptical orbits, which is common in the literature, but usually an eccentricity as high as that of the PROBA-3 mission is not considered. With this in mind, a new approach for sampling the dynamics for the prediction horizon is proposed here, based on constant eccentric anomaly sampling intervals, which better deals with the fact that the dynamics are highly time-varying for an such a highly elliptical orbit. We compare the Finite-Horizon and Variable-Horizon MPC formulations, regarding performance and computational complexity, and compare them with the traditional two-impulse transfer approach used in traditional rendezvous guidance algorithms.

A new method for formulating a passive safety constraint with online linear programming is also developed, which relies on offline nonlinear optimization and can be slightly suboptimal in face of disturbances, although it can allow for the feasible inclusion of this constraint in a real-time application. A variation of this technique which relies on iterative linear optimization is also presented, which in some cases can converge to the solution of the nonlinear optimization, and with much less computational effort. Finally, we contribute with new robustness techniques, with the use of a terminal quadratic controller for a more accurate and robust final braking manoeuvre, and the dynamic relaxation of the terminal constraint, in order keep the control sparse and avoid the overcorrection of disturbances and waste of fuel.

1.5 Thesis Outline

In Chapter 2, we first cover general MPC theory, with a focus on MPC for linear system models, given the context of this work. The basic general formulation is discussed, and some specific techniques are also presented, such as reference tracking and the use of different cost functions. This chapter also features several simulations that show the effect of different cost functions and of the controller parameters, with some consideration also given to the computational performance.

The relative orbital dynamics between two satellites, which is crucial to understand for the design of a rendezvous mission, are presented in Chapter 3. We introduce and derive linearized models of the relative dynamics, which will then be used for MPC in the next chapter. Several simulations of the relative motion between two satellites are also presented, both in the circular and elliptic orbit cases, and the non-intuitive free-drift motions and thrust manoeuvres are explained.

Finally, in Chapter 4 the MPC framework is applied to the rendezvous problem. We start by considering the most naive approach, and develop stepwise towards the ideal formulation. We then consider the presence of disturbances, and provide a short literature review on robust techniques in MPC for rendezvous, before presenting our own contributions. Several rendezvous simulations with the presented methods are featured, some of them being in the conditions of the PROBA-3 mission.

Chapter 2

Model Predictive Control

Model Predictive Control (MPC) is a Control design method based on iterative online optimization [2]. The strategy is to obtain a control decision by solving an optimization problem which factors in future states of the system in a finite horizon, predicted using a (generally discrete) system model. Figure 2.1 illustrates this approach.

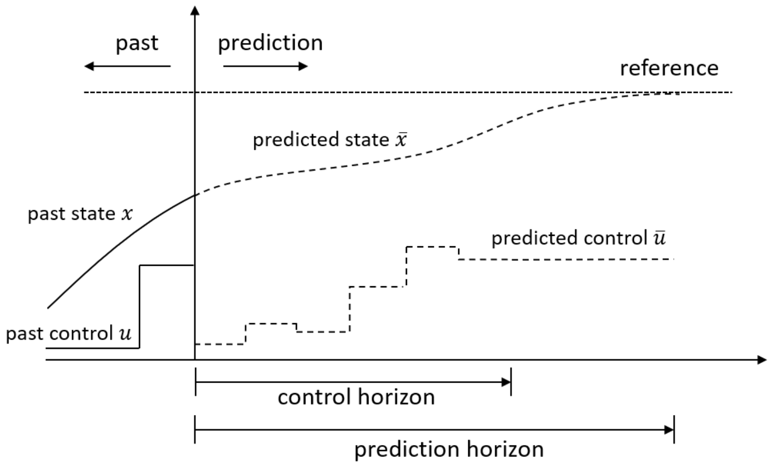


Figure 2.1: Illustration of the Model Predictive Control strategy.

At each time step, the problem is solved with the most recent state measurement or estimate as the initial condition for the prediction, and a control strategy for future steps within the prediction horizon is obtained. The first control value in the obtained sequence is executed, and the problem is solved again in the next time step, with an updated state and with the prediction horizon shifted forward. For this reason, this method is also known as Moving/Receding Horizon Control.

Since MPC is formulated as an optimization problem, it allows for the inclusion of control and state constraints. This is a powerful tool and one of the major advantages of MPC in respect to other control methods, since it allows to limit the control action and to model complex state restrictions, such as safety constraints. Also, MPC naturally considers the system dynamics and can easily handle multivariate systems. Another feature is that it allows for the use of nonlinear system models (NMPC), which generate better state predictions.

By definition, the MPC strategy requires that an optimization problem be solved online, at each time step. The computation time of the MPC problem depends on many factors, such as the order of the system model, if it is linear or nonlinear, the complexity of the control and state constraints, and the length of the prediction horizon. The optimal control action must be computed and executed before the next sample, and thus the problem is required to be solved faster than the system sampling time, which makes its implementation infeasible in fast systems. The computational requirement is the greatest limitation for MPC, although modern technology and methods allow for MPC to be implemented in increasingly more complex systems, such as those in the aerospace industry.

In this chapter, we first introduce the basic concepts of MPC theory in Section 2.1, and then discuss different methods, including MPC for linear system in Section 2.2 and for nonlinear systems in Section 2.3. Section 2.4 presents a commonly used suboptimal complexity reduction technique. In Section 2.5 we show several MPC experiments with two different systems: a linear two-dimension pure inertial system, and a nonlinear unicycle model.

2.1 Model Predictive Control Formulation

MPC theory is traditionally formulated in discrete-time, and so it is here. A discrete-time system, with state variables x, inputs u and outputs y, is generally described by the difference equation

$$x^+ = f(x, u), \tag{2.1}$$

where x^+ is the system state at the next time step and $f(x, u)$ is the system model, and by the output equation

$$y = g(x, u), \tag{2.2}$$

where $g(x, u)$ is the sensor model. If models f and g are linear, the system can instead be described by the state-space model

$$x^+ = Ax + Bu \tag{2.3}$$

$$y = Cx + Du, \tag{2.4}$$

where usually $D = 0$, except for systems with an instantaneous response. For literature on discrete-time systems and digital control see [20].

Model Predictive Control solves an open-loop optimal control problem every time-step. At a time-instant t the following optimization problem is solved

$$\min_{\substack{\bar{u}_t,\dots,\bar{u}_{t+N-1} \\ \bar{x}_t,\dots,\bar{x}_{t+N}}} \quad \sum_{i=0}^{N-1} l(\bar{x}_{t+i}, \bar{u}_{t+i}) \; + \; V_f(\bar{x}_{k+N}) \tag{2.5a}$$

$$\text{s.t.} \qquad \bar{x}_t = x_t, \tag{2.5b}$$

$$\bar{x}^+ = f(\bar{x}, \bar{u}), \tag{2.5c}$$

$$\bar{x}_k \in \mathcal{X}_k,\ k = t, \dots, t+N, \tag{2.5d}$$

$$\bar{u}_k \in \mathcal{U}_k,\ k = t, \dots, t+N-1 \tag{2.5e}$$

where $\bar{x}$ and $\bar{u}$ are the predictions of x and u, N is the length of the prediction horizon, and $l(\cdot,\cdot)$ and $V_f(\cdot)$ are cost functions. The minimization is subject to ("s.t.") constraints (2.5b-e). The constraint in (2.5b) sets the initial condition for the prediction, and the state and control predictions are subject to the system model in constraint (2.5c). The sets $\mathcal{X}$ and $\mathcal{U}$ in (2.5d) and (2.5e) represent constraints on the state and control variables, respectively. Solving this open-loop problem yields an optimal control sequence $\bar{u}^*$, of which only the first is applied, meaning that $u_t = \bar{u}_t^*$. The problem is then solved again at the next time-step, at $t+1$, with the updated state measurement/estimate x_{t+1} that is the response to the applied control, thus closing the control loop.

If the system is time-invariant, we have that $f(x_i, u_i) = f(x_j, u_j)$ for $x_i = x_j$ and $u_i = u_j$, where the dependence of $f(\cdot,\cdot)$ on time was previously implicit. The formulation then simplifies to

$$\min_{\substack{\bar{u}_0,\dots,\bar{u}_{N-1} \\ \bar{x}_0,\dots,\bar{x}_N}} \quad \sum_{i=0}^{N-1} l(\bar{x}_i, \bar{u}_i) \ + \ V_f(\bar{x}_N) \tag{2.6a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{2.6b}$$

$$\bar{x}^+ = f(\bar{x}, \bar{u}), \tag{2.6c}$$

$$\bar{x}_k \in \mathcal{X}_k,\ k = 0, \dots, N, \tag{2.6d}$$

$$\bar{u}_k \in \mathcal{U}_k,\ k = 0, \dots, N-1 \tag{2.6e}$$

For literature on more MPC theory, such as stability of the closed-loop system, see, for example, the book by Rawlings et al. [2].

2.2 Linear MPC

Linear MPC refers to the control of linear systems. For a linear system the MPC formulation is

$$\min_{\substack{\bar{u}_0,\dots,\bar{u}_{N-1} \\ \bar{x}_0,\dots,\bar{x}_N}} \quad \sum_{i=0}^{N-1} l(\bar{x}_i, \bar{u}_i) \ + \ V_f(\bar{x}_N) \tag{2.7a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{2.7b}$$

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k,\ k = 0, \dots, N-1, \tag{2.7c}$$

$$\bar{x}_k \in \mathcal{X}_k,\ k = 0, \dots, N, \tag{2.7d}$$

$$\bar{u}_k \in \mathcal{U}_k,\ k = 0, \dots, N-1. \tag{2.7e}$$

where the constraint (2.7c) is now the linear state-space model of the system. Since this constraint is linear, the problem is easier and faster to solve, in comparison to one with a nonlinear system.

A common and effective cost for the MPC optimization problem is the quadratic cost, where the cost functions become

$$l(x,u) = x^\top Q x + u^\top R u$$
$$V_f(x) = x^\top Q_f x, \tag{2.8}$$

where Q and Q_f are positive semi-definite matrices, and R is a positive definite matrix. These cost matrices are used to tune the controller: increasing the elements in R relative to Q and Q_f increases the penalization of the control variable in the cost function, and so the optimal solution will have limited actuator action. This formulation, called 'regulator', controls the system to the origin. A common choice for the terminal state matrix Q_f is the solution to the algebraic Riccati equation, since in some cases it guarantees closed-loop stability [21].

For an easy and efficient implementation of linear quadratic MPC, it is convenient to adopt a matrix representation. Concatenating the predicted state and control variables into X and U, we have the following matrix equation that satisfies the system model

$$\underbrace{\begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_N \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 0 & \dots & \dots & 0 \\ A & \dots & 0 & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & A & 0 \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_N \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ B & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & B \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} \bar{u}_0 \\ \bar{u}_1 \\ \vdots \\ \bar{u}_{N-1} \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{E} x_t. \tag{2.9}$$

Matrices $\tilde{A}$ and $\tilde{B}$ are augmented system model matrices, and matrix E ensures the initial condition of the prediction. We also define the augmented cost matrices

$$\tilde{Q} = \begin{bmatrix} Q & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Q_f \end{bmatrix} \qquad \tilde{R} = \begin{bmatrix} R & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & R \end{bmatrix}. \tag{2.10}$$

These augmented matrices can be easily generated by performing the Kronecker tensor product with the identity matrix; in MATLAB this can be achieved with function *kron*.

With this matrix representation, the problem in (2.7) with the quadratic cost in (2.8) simplifies to

$$\min_{X,U} \quad X^\top \tilde{Q} X + U^\top \tilde{R} U \tag{2.11a}$$
$$\text{s.t.} \quad X = \tilde{A}X + \tilde{B}U + Ex(t), \tag{2.11b}$$
$$X \in \tilde{\mathcal{X}}, \tag{2.11c}$$
$$U \in \tilde{\mathcal{U}}, \tag{2.11d}$$

where $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{U}}$ represent the state and control constraints over the whole prediction horizon.

It is sometimes useful to formulate the MPC problem with the system output, instead of the state.

The MPC output form with quadratic cost is then

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{y}_0,\ldots,\bar{y}_N}} \quad \sum_{i=0}^{N-1} \bar{y}_i^\top Q \bar{y}_i + \bar{u}_i^\top R \bar{u}_i \;+\; \bar{y}_N^\top Q_f \bar{y}_N \tag{2.12a}$$

$$\text{s.t.} \qquad \bar{x}_0 = x_t, \tag{2.12b}$$

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k,\ k = 0,\ldots,N-1, \tag{2.12c}$$

$$\bar{y}_k = C\bar{x}_k + D\bar{u}_k, \tag{2.12d}$$

$$\bar{y}_k \in \mathcal{Y}_k,\ k = 0,\ldots,N, \tag{2.12e}$$

$$\bar{x}_k \in \mathcal{X}_k,\ k = 0,\ldots,N, \tag{2.12f}$$

$$\bar{u}_k \in \mathcal{U}_k,\ k = 0,\ldots,N-1 \tag{2.12g}$$

where $\bar{y}$ is the predicted output and $\mathcal{Y}$ represents output constraints. This formulation now includes the system output equation in constraint (2.12d), as well as output constraints in (2.12e).

2.2.1 Reference Tracking

To control the system to a reference set point x_{ref} instead of to the origin, the tracking error must penalized, and so the cost functions become

$$\begin{aligned} l(x,u) &= (x - x_{ref})^\top Q (x - x_{ref}) + u^\top R u \\ V_f(x) &= (x - x_{ref})^\top Q_f (x - x_{ref}). \end{aligned} \tag{2.13}$$

For systems without integral action, however, the optimal solution will not be at $x = x_{ref}$ for x_{ref} different than zero. For these systems, maintaining the state at a value different than the origin requires a constant non-zero control, which will weigh on the cost function and distance x from x_{ref}. Therefore, this formulation presents a static error for systems without integral action. To achieve an error-free reference tracking for these systems, a control reference u_{ref} must be added

$$l(x,u) = (x - x_{ref})^\top Q (x - x_{ref}) + (u - u_{ref})^\top R (u - u_{ref}) \tag{2.14}$$

The control reference u_{ref} must be the control value that in steady-state makes the state be equal to the reference, and so from the system model we have

$$u_{ref} = B^{-1}(I - A)x_{ref}. \tag{2.15}$$

Note, however, that the state reference x_{ref} cannot be arbitrarily chosen, since some states cannot be maintained in steady-state. For example, a vehicle cannot maintain the same position while simultaneously having non-zero velocity. To check if the reference x_{ref} can be tracked in steady-state, the result in (2.15) can be inserted back into the system model; if the state at the next time step is not equal to the reference, then it is not admissible. For systems with integral action, u_{ref} will be zero for admissible state references.

The reference tracking formulation in matrix form is

$$\min_{X,U} \quad (X - X_{ref})^\top \tilde{Q}(X - X_{ref}) + (U - U_{ref})^\top \tilde{R}(U - U_{ref}) \tag{2.16a}$$

$$\text{s.t.} \quad X = \tilde{A}X + \tilde{B}U + Ex(t), \tag{2.16b}$$

$$U \in \tilde{\mathcal{U}}, \tag{2.16c}$$

$$X \in \tilde{\mathcal{X}} \tag{2.16d}$$

where $X_{ref} = [x_{ref}^\top \dots x_{ref}^\top]^\top$ and $U_{ref} = [u_{ref}^\top \dots u_{ref}^\top]^\top$. Because in (2.15) the control reference is determined from the system model, this technique only completely eliminates the static error if the model is perfect.

An alternative way to achieve reference tracking without static error is by adding integral action to the controller. This can be performed by penalizing the control increment Δu between samples, instead of the full control action u. Thus, in steady-state the control action will remain constant and the increment Δu will be zero, eliminating the static error. The optimization problem then becomes

$$\min_{\substack{\bar{u}_0,\dots,\bar{u}_{N-1}\\ \Delta\bar{u}_0,\dots,\Delta\bar{u}_{N-1}\\ \bar{x}_0,\dots,\bar{x}_N}} \quad \sum_{i=0}^{N-1} (\bar{x}_i - x_{ref})^\top Q(\bar{x}_i - x_{ref}) + \Delta\bar{u}_i^\top R \Delta\bar{u}_i \; + \; (\bar{x}_N - x_{ref})^\top Q_f(\bar{x}_N - x_{ref}) \tag{2.17a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{2.17b}$$

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k, \; k = 0, \dots, N-1, \tag{2.17c}$$

$$\bar{u}_0 = u_{t-1} + \Delta\bar{u}_0, \tag{2.17d}$$

$$\bar{u}_k = \bar{u}_{k-1} + \Delta\bar{u}_k, \; k = 1, \dots N-1, \tag{2.17e}$$

$$\bar{x}_k \in \mathcal{X}_k, \; k = 0, \dots, N, \tag{2.17f}$$

$$\bar{u}_k \in \mathcal{U}_k, \; k = 0, \dots, N-1, \tag{2.17g}$$

$$\Delta\bar{u}_k \in \Delta\mathcal{U}_k, \; k = 0, \dots, N-1. \tag{2.17h}$$

The model in (2.17c) still uses the full control $\bar{u}$, which has to be determined from $\Delta\bar{u}$ and the previous $\bar{u}$ in (2.17e). In constraint (2.17d) the initial condition for $\bar{u}$ is set from the last control action applied u_{t-1}. Note that, in the presence of disturbances such that the previous control action u_{t-1} is not perfectly known, some static error can be introduced. Also, this formulation is not necessarily optimal since it does not penalize constant control actions, which is not desirable in systems with constrained energy/fuel. Reference [22] presents other strategies for introducing integral action.

Another method of tracking a state reference is with the use of a terminal constraint, as an optimization hard-constraint

$$\bar{x}_N = x_{ref}. \tag{2.18}$$

Because of the receding horizon strategy, this constraint does not ensure that the system will reach the reference at sample N, or at all, since only the first optimal control action is applied and then the horizon slides forward. Thus, as the horizon N increases, the further away from the reference the steady-state system will be. To counteract this, the state cost matrices Q and Q_f can be set to zero, which allows for the system to converge to the state reference if the system has integral action, although not necessarily in N samples. For systems without integral action, the strategies in (2.16) and (2.17) can be used with the terminal constraint, in which case the advantage in using this constraint is to ensure system stability.

The terminal constraint can be also be used to ensure the system reaches the reference in N samples, by decrementing the prediction horizon at each time sample, which is no longer the receding horizon strategy. However, if the prediction horizon becomes short enough, the optimization problem can become infeasible, since it may be impossible to reach the reference from the initial condition in N samples, given the system dynamics, disturbances, and the control and state constraints. This strategy is employed in Chapter 4 for the MPC controller for rendezvous.

2.2.2 State Substitution

Notice that, in an MPC problem, the state at any time can be predicted solely from the initial condition and the sequence of control actions up to that time. This structure can be exploited to eliminate the state as an optimization variable, also eliminating the optimization constraints related to the prediction model, thus greatly simplifying the problem and allowing it to be solved faster, either with analytical or numerical optimization. Note that, however, numerical optimization algorithms specific for MPC can exploit its structure and often this technique is not applied.

Given the prediction model in matrix form as defined in Section 2.2

$$X = \tilde{A}X + \tilde{B}U + Ex(t), \tag{2.19}$$

where X and U are the state and control variables in vector form, and matrices $\tilde{A}$, $\tilde{B}$ and E are the augmented state matrices, as defined in (2.9). This can be rewritten as

$$X = \underbrace{(I - \tilde{A})^{-1}\tilde{B}}_{F}U + \underbrace{(I - \tilde{A})^{-1}E}_{K}x(t). \tag{2.20}$$

Note that, since matrix $\tilde{A}$ is lower triangular, the determinant of matrix $(I - \tilde{A})$ is 1 and thus it is always invertible.

Quadratic Cost

For example, substituting X in the formulation with quadratic cost in (2.16) yields

$$\min_{U} \quad (FU + Kx(t) - X_{ref})^\top \tilde{Q}(FU + Kx(t) - X_{ref}) + (U - U_{ref})^\top \tilde{R}(U - U_{ref}) \tag{2.21a}$$

$$\text{s.t.} \quad FU + Kx(t) \in \tilde{\mathcal{X}}, \tag{2.21b}$$

$$U \in \tilde{\mathcal{U}} \tag{2.21c}$$

Simplifying and removing terms independent of U, the cost function becomes

$$V(U) = \frac{1}{2}U^\top HU + \underbrace{\Big(Jx(t) - LX_{ref} - \tilde{R}^\top U_{ref}\Big)^\top}_{f} U \tag{2.22}$$

with

$$\begin{aligned} H &= F^\top \tilde{Q} F + \tilde{R} \\ J &= F^\top \tilde{Q}^\top K \\ L &= F^\top \tilde{Q}^\top \end{aligned} \tag{2.23}$$

Notice that, by applying this substitution, the state has been eliminated as an optimization variable, as well as the constraints associated with the prediction model. Note, however, that the state is no longer directly accessible and for state constraints it must be calculated from the control variables and initial condition, as can be seen in (2.21b).

2.2.3 ℓ_1-Norm Cost

Although less common than the quadratic cost, the ℓ_1-norm is also used for the MPC cost function instead of the quadratic cost (note that, the quadratic cost is the squared ℓ_2-norm). Denoting the ℓ_1-norm of a vector w by $\|w\|_1$, the cost functions become

$$\begin{aligned} l(x,u) &= \|Q(x - x_{ref})\|_1 + \|Ru\|_1 \\ V_f(x) &= \|Q_f(x - x_{ref})\|_1. \end{aligned} \tag{2.24}$$

This cost function generates sparse solutions, and so the control becomes approximately *bang-bang*, meaning that the actuators are either fully turned on or off. In matrix form, this formulation takes the shape of

$$\min_{X,U} \quad \|\tilde{Q}(X - X_{ref})\|_1 + \|\tilde{R}U\|_1 \tag{2.25a}$$

$$\text{s.t.} \quad X = \tilde{A}X + \tilde{B}U + \tilde{E}x(t), \tag{2.25b}$$

$$X \in \tilde{\mathcal{X}}, \tag{2.25c}$$

$$U \in \tilde{\mathcal{U}}. \tag{2.25d}$$

2.2.4 LASSO Cost

Another possibility is to add an ℓ_1-norm control cost to the quadratic cost function, which is known as the LASSO cost function and is commonly used in regression analysis. The aim is to retain the desirable properties of the quadratic cost, such as robustness, while adding some sparsity due to the ℓ_1-norm. In matrix form, the cost function becomes

$$V(X,U) = (X - X_{ref})^\top \tilde{Q}(X - X_{ref}) \ + \ U^\top \tilde{R}U \ + \ \|\tilde{R}_\lambda U\|_1, \tag{2.26}$$

where R_λ is the control cost matrix associated with the ℓ_1-norm term.

A subclass of this formulation is to set $R = 0$, becoming similar to the formulation in Section 2.2.3 but with a quadratic cost on the state variables, therefore reducing sparsity regarding the tracking of the reference.

2.3 Non-Linear MPC

Nonlinear MPC (NMPC) implies the control of systems with nonlinear models. Most real systems are nonlinear and sometimes these cannot be accurately approximated by a linearised model in the whole operating region. Thus, NMPC allows for the use of a nonlinear prediction model, which produces better state predictions, allowing for better control and to operate systems closer to the boundary of the admissible operating region [23]. On the other hand, they incur a higher computational cost, due to the necessity of solving an optimization problem with nonlinear constraints.

The NMPC with quadratic cost is formulated as

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad \sum_{i=0}^{N-1} \bar{x}_i^\top Q \bar{x}_i + \bar{u}_i^\top R \bar{u}_i \ + \ \bar{x}_N^\top Q_f \bar{x}_N \tag{2.27a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{2.27b}$$

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k), \ k = 0, \ldots, N-1, \tag{2.27c}$$

$$\bar{x}_k \in \mathcal{X}_k, \ k = 0, \ldots, N, \tag{2.27d}$$

$$\bar{u}_k \in \mathcal{U}_k, \ k = 0, \ldots, N-1. \tag{2.27e}$$

where the constraint (2.27c) is now nonlinear.

For references on theory and implementation of NMPC see the introduction by Findeisen and Allgöwer [23] and the tutorial by Rawlings [24].

2.4 Move Blocking

A strategy to reduce the complexity of the optimization problem, known as *move blocking*, is to shorten the number of control decisions. However, because decreasing the prediction horizon worsens the controller performance, a *control horizon* N_u is introduced, such that control actions beyond this horizon are all equal to the last control decision at time $k = N_u - 1$, as illustrated in figure 2.1, thus reducing the number of optimization variables. The control horizon is necessarily equal to or less than the prediction horizon N. The move-blocking formulation becomes

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N_u-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad \sum_{i=0}^{N_u-1} l_1(\bar{x}_i, \bar{u}_i) \ + \sum_{j=N_u}^{N-1} l_2(\bar{x}_j, u_{N_u-1}) \ + \ V_f(\bar{x}_N) \tag{2.28a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{2.28b}$$

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k), \ k = 0, \ldots, N_u - 1, \tag{2.28c}$$

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_{N_u-1}),\ k = N_u, \ldots, N-1, \tag{2.28d}$$

$$\bar{x}_k \in \mathcal{X}_k,\ k = 0, \ldots, N, \tag{2.28e}$$

$$\bar{u}_k \in \mathcal{U}_k,\ k = 0, \ldots, N-1 \tag{2.28f}$$

Because the number of control decisions is reduced, this complexity reduction strategy is suboptimal.

2.5 Experiments and Results

This section features several simulations with Model Predictive Control, and the effect of using different cost functions is shown, as well as the effect of tuning its parameters, such as the prediction horizon and weight matrices. Both linear and nonlinear systems are considered, as well as linear and nonlinear constraints. The aim of these experiments is not to obtain optimal model predictive controllers for these specific systems and problems, with feasible real-time implementations, but rather to show the capabilities and limitations of MPC. Furthermore, no disturbances are present in these simulations.

The following experiments were performed in MATLAB. In the absence of inequality-constraints, the problem can easily be solved analytically, except with the ℓ_1-norm cost. Otherwise, the MATLAB Optimization Toolbox is used to solve the optimal control problem numerically. For a controller with a quadratic cost, linear model and linear constraints, the optimal control problem is a quadratic program (QP) and can be solved with the function *quadprog*. For problems with nonlinear models or nonlinear constraints, the function *fmincon* is required to solve the resulting optimal control problem. Note that MATLAB also features a Model Predictive Control toolbox, although it is not used here.

To solve the optimization problem faster with numerical algorithms, the solution of one MPC iteration can be used as the initial point for the next, a technique known as *warm start*. Also, for linear systems, the state-substitution method presented in Section 2.2.2 is applied in order to reduce the number of optimization variables, further decreasing the computation time. Note, however, that some optimization algorithms specialized for MPC take advantage of the problem structure and do not apply this substitution, *i.e.* [25], although that is not he case here. In the following experiments, the average computation times of solving the optimal control problem are presented, using a 4th Generation 2.4GHz Intel-i7 Processor.

2.5.1 Two-Dimension Pure Inertial System

To experiment with linear MPC, a pure inertial system in two dimensions is considered. The system is described by two double integrators, and its discrete space-state model, if sampled with a zero-order hold (ZOH), is

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} x_k + \underbrace{\begin{bmatrix} T_s^2 & 0 \\ T_s & 0 \\ 0 & T_s^2 \\ 0 & T_s \end{bmatrix}}_{B} u_k, \tag{2.29}$$

where T_s is the sampling period. In these experiments, the output model is not considered, as full state measurement is assumed. State variables x_1 and x_3 are the system position in the x-axis and y-axis, respectively, and states x_2 and x_4 are the velocity. The system has two inputs that allow it to move in any direction with no restrictions (holonomic movement). A sampling period of 0.1 s was used in the following experiments.

Unconstrained

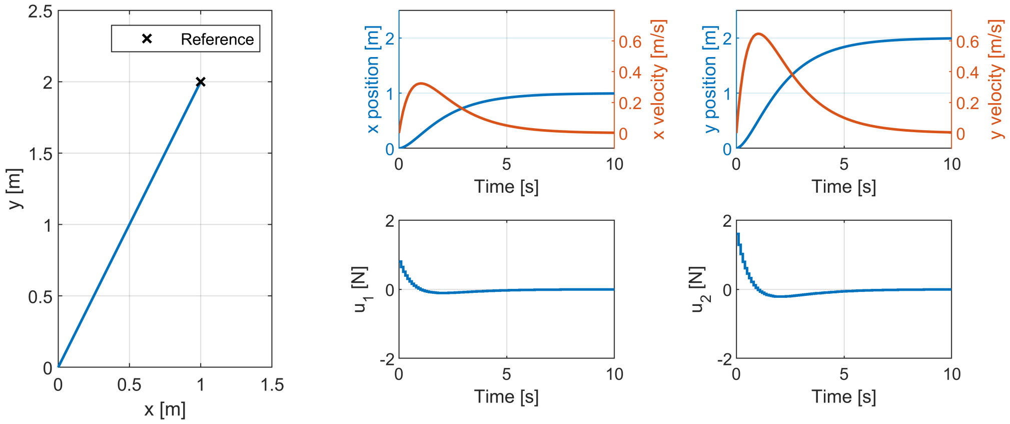
To begin with, no state or control constraints are considered, and a quadratic cost function is used. Table 2.1 contains the controller parameters used in each experiment, and the average time for solving the optimization problem (analytically).

Figure	N	R	Q	Q_f	t_{avg}
2.2	10	I	I	I	50 µs
2.3	10	$5I$	I	I	48 µs
2.4	20	$5I$	I	I	71 µs

Table 2.1: Controller parameters and computation times for unconstrained MPC experiments.

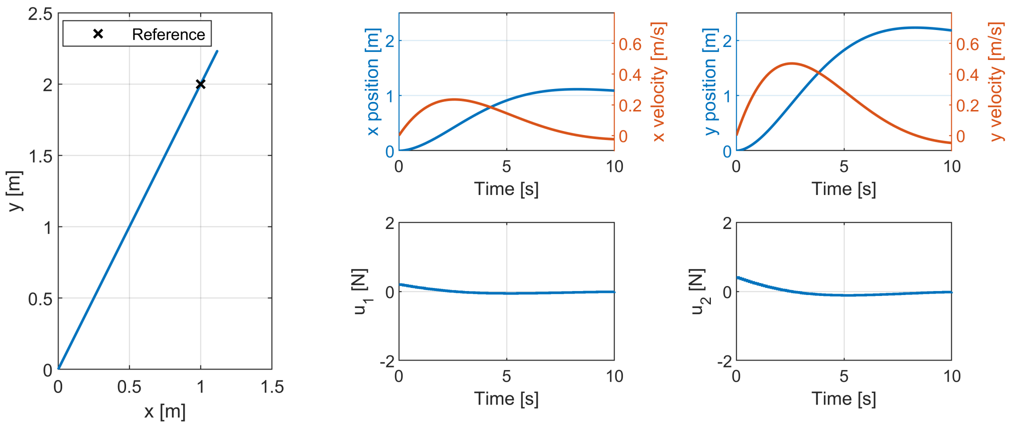
In a first experiment, the system starts from the origin with no velocity, and the reference state is at coordinates (1,2) with zero velocity. The prediction horizon is $N = 10$ samples, which together with the sampling period of 0.1 s grants a prediction of 1 s ahead, and all quadratic cost matrices are equal to the identity matrix. The results in figure 2.2 were obtained, where figure 2.2a) contains the trajectory in the 2D space, and figure 2.2b) present the state (top) and control variables (bottom) as a function of time. Note that the position and velocity vectors for each direction are plotted with different scales. The system goes in a straight line toward the reference, requiring more control action from the second input to do so, since the reference in the y-axis is further away. Furthermore, while the control action may appear to be continuous and smooth, the control values are constant in each time sample, due to the ZOH discretization.

In the next experiment, the weight of the control action in the cost function is increased. It can be observed in figure 2.3 that the control action is now less energetic, and as a result the system overshoots the reference and takes longer to converge. This is due to the fact that the control action now weighs more in the cost function, and so the minimum is such that there is less control action and a greater tracking error. On the other hand, increasing cost matrices Q and Q_f relative to R results in a more aggressive control action in order to track the reference more closely. The cost matrices can then be used to tune the trade-off between energy spent controlling the system and the speed at which it converges to the reference.



(a) Trajectory (b) State and control variables

Figure 2.2: Control of pure inertial system with no constraints.

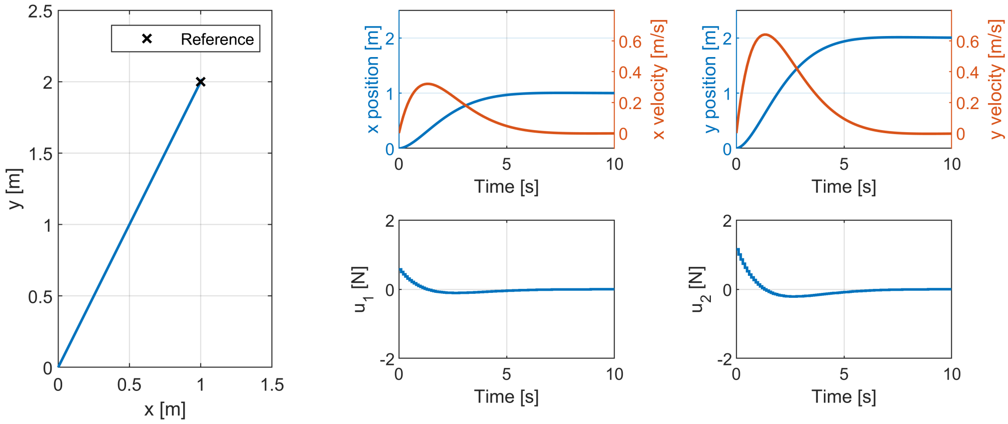


(a) Trajectory (b) State and control variables

Figure 2.3: Control of pure inertial system with no constraints and increased control cost.

Lastly, the prediction horizon is doubled, and the results in figure 2.4 are obtained. The control action is again more aggressive, and the reference tracking is better and without overshoot, an effect which is the same as that of decreasing the control cost. In fact, in the absence of any constraints, changing the prediction horizon has a similar effect as tuning the cost matrices. However, the prediction horizon also has an effect on system stability [21], and if the prediction horizon is too short, the closed-loop system can become unstable. With the increase of the prediction horizon N, the state is predicted and optimized for longer into the future, which increases stability. In the limit, an infinite prediction horizon guarantees a asymptotically stable closed-loop system, if the open-loop system is stabilizable [2]. Furthermore, it can be seen from table 2.1 that the computation time increases with the prediction horizon, as expected.

Another important consideration is that increasing N increases the number of optimization variables, and thus increases the computation time for solving the optimization problem. As can be seen from table 2.1, these optimal control problems can be solved very efficiently, which is due to the fact that the problem is unconstrained and can be solved analytically. In the presence of inequality constraints, however, numerical optimization algorithms are required.



(a) Trajectory (b) State and control variables

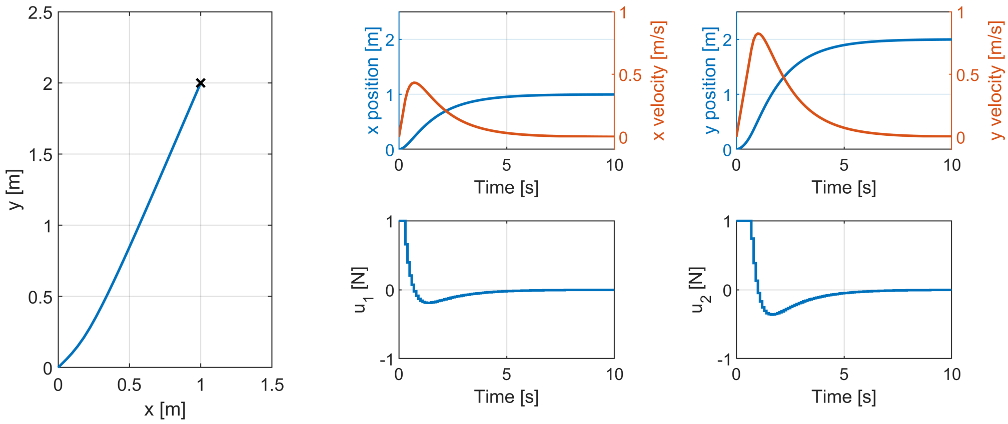
Figure 2.4: Control of pure inertial system with no constraints and increased prediction horizon.

Control and State Constraints

Applying lower and upper bounds of ± 1 N on the control variables with optimization constraints, the result in figure 2.5 is obtained. The controller parameters used are contained in table 2.2, as well as the average computation times. The system now converges asymmetrically toward the reference, since a straight-line trajectory would require more control action from u_2, and both control variables are saturated at the beginning.

Figure	N	R	Q	Q_f	Constraints	t_{avg}
2.5	10	I	$10I$	$10I$	control bounds	$1.53\,\text{ms}$
2.6	10	I	$10I$	$10I$	control bounds, circular obstacle	$18\,\text{ms}$
2.7	20	I	$10I$	$10I$	control bounds, circular obstacle	$41\,\text{ms}$
2.8	10	I	$10I$	$10I$	control bounds, 4 circular obstacles	$26\,\text{ms}$
2.9	10	I	$10I$	$10I$	control bounds, square obstacle	$21\,\text{ms}$

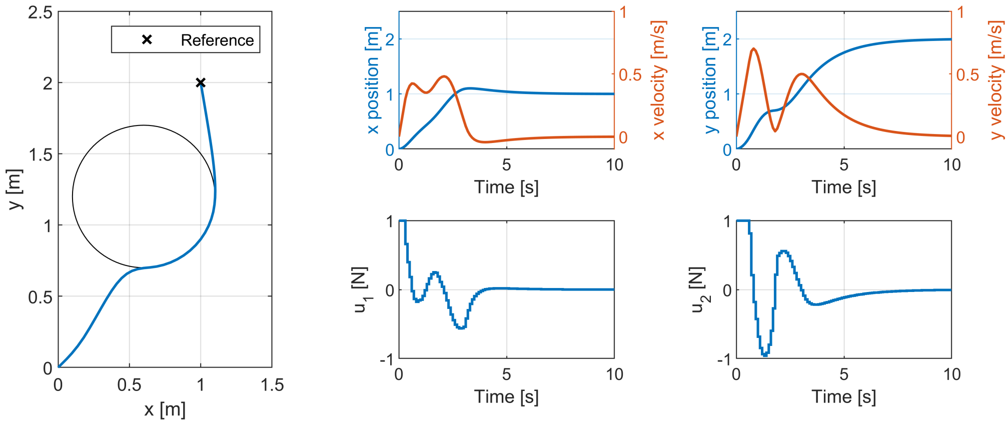
Table 2.2: Controller parameters and computation times for constrained MPC experiments.



(a) Trajectory (b) State and control variables

Figure 2.5: Control of pure inertial system with control limits.

An obstacle avoidance constraint is added, which constrains the position state variables. As seen in figure 2.6, the system goes around the obstacle, but because it is modelled as a single point object, it travels very close and along the obstacle border, since it is the most efficient trajectory that satisfies the constraints. It can also be observed that, initially, the system travels toward the object, in a similar trajectory as in figure 2.5, and, as the prediction horizon reaches it, the system starts to swerve.

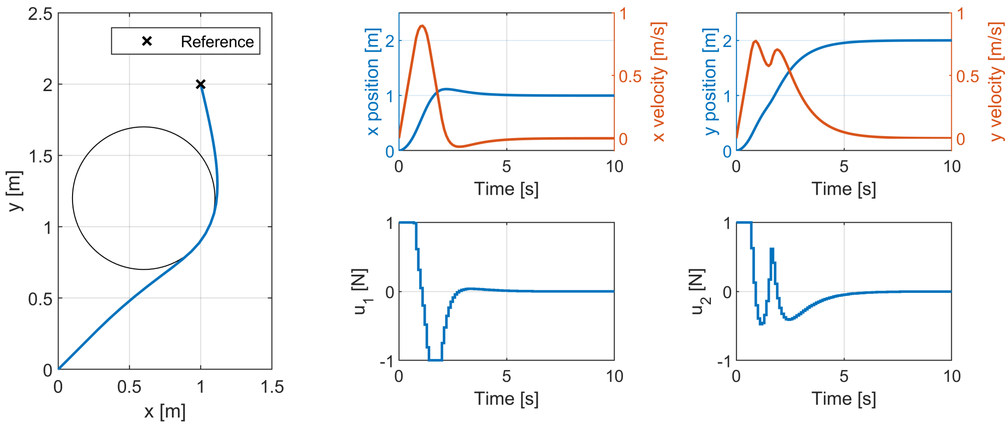


(a) Trajectory

(b) State and control variables

Figure 2.6: Control of pure inertial system with circular obstacle.

Increasing the prediction horizon, it can be observed in figure 2.7 that the system now diverts its trajectory earlier to avoid the obstacle, since it is detected earlier. Moreover, the control sequence is now cleaner, with fewer corrections along the way, because the controller has a better plan of the trajectory.

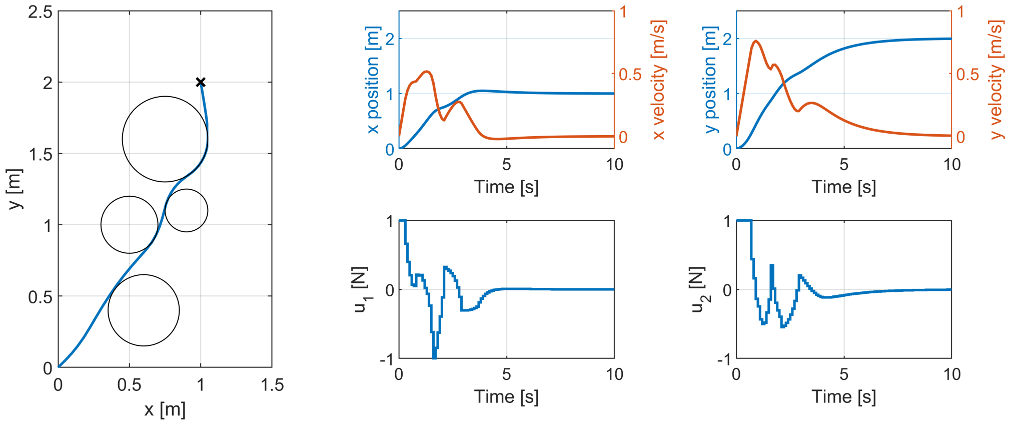


(a) Trajectory

(b) State and control variables

Figure 2.7: Control of pure inertial system with circular obstacle and increased prediction horizon.

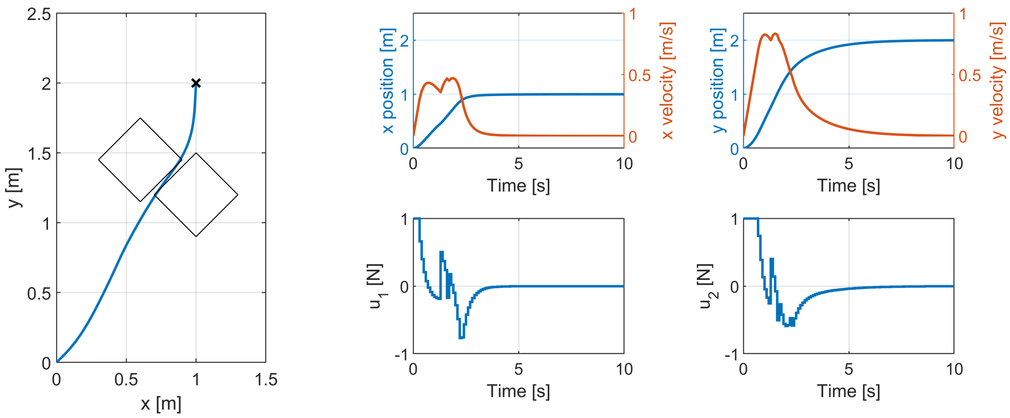
Adding multiple circular obstacles, the results in figure 2.8 are obtained. Note that obstacle avoidance constraints are non-convex, meaning that the optimization problem may have several local minima to which the algorithm can converge, depending on the initial point. For example, any solution for which the trajectory goes around a different side of an obstacle is a local minimum. While the trajectory in figure 2.8 appears to be a global minimizer, it might not be, depending on the cost function. This particular solution requires several direction corrections to avoid the obstacles, and so for some cost matrices the global minimum might be to go around all the obstacles. Globally optimizing non-convex problems is usually performed by solving the problem several times with different initial points, which can be infeasible to do in real-time. The MATLAB Global Optimization Toolbox allows for solving this type of problem, although it is not used here.



(a) Trajectory (b) State and control variables

Figure 2.8: Control of pure inertial system with multiple circular obstacles.

Figure 2.9 shows an experiment with two square obstacles, where it can be observed that the trajectory passes through the corners of the obstacle. This is a limitation of the way the obstacle constraint is formulated, since only the discrete points are constrained, and not the whole continuous trajectory. Upon inspecting the discrete positions along the trajectories, it is observed that the constraints are in fact satisfied. It is possible to formulate an obstacle constraint that restricts the line between discrete points, at the cost of extra computing power. Another way to minimize this effect is to decrease the sampling period, or to use intermediate samples for the constraints only. It is also possible to constraint the state in continuous time via semidefinite programming [26].



(a) Trajectory (b) State and control variables

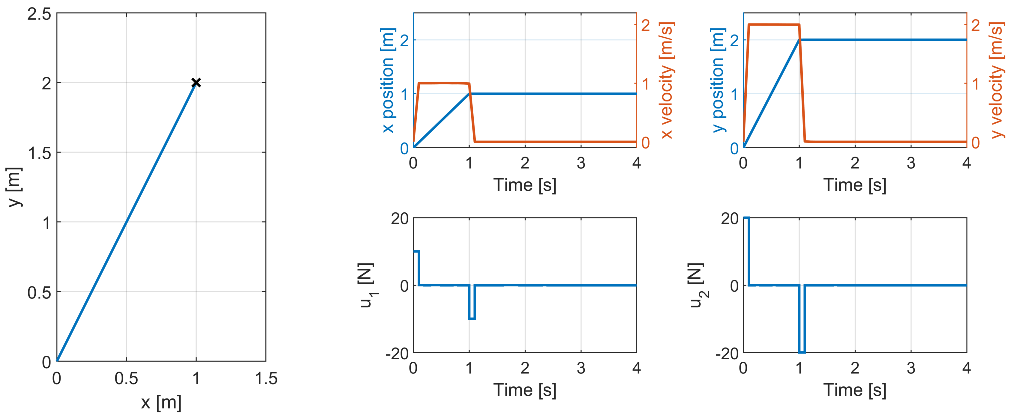
Figure 2.9: Control of pure inertial system with two square obstacles.

ℓ_1-Norm Cost

Using the ℓ_1-norm for the cost function in the absence of constraints yields the result in figure 2.10, with the controller parameters presented in Table 2.3. The cost on the velocity states has to be reduced, otherwise these would weigh too much on the cost function and the controller would generate no action, due to the sparsity of the ℓ_1-norm. It can be seen that the actuators are turned on for one sample only, resting afterwards, resulting in a constant velocity trajectory. As the system approaches the reference, its velocity is cancelled in one sample again, and thus the control is *bang-bang*.

Figure	N	R	Q	Q_f	Constraints	t_{avg}
2.10	10	I	$10\,\mathrm{diag}(1,0,1,0)$	$100\,\mathrm{diag}(1,0,1,0)$	none	179 ms
2.11	10	I	$10\,\mathrm{diag}(1,0,1,0)$	$100\,\mathrm{diag}(1,0,1,0)$	control bounds	281 ms

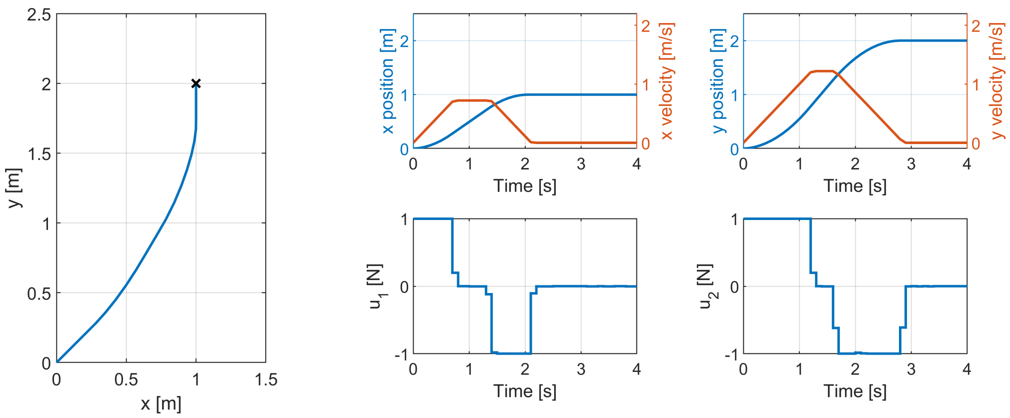
Table 2.3: Controller parameters and computation times for ℓ_1-norm MPC experiments.



(a) Trajectory (b) State and control variables

Figure 2.10: Control of pure inertial system with ℓ_1-norm cost and no constraints

Adding control limits in figure 2.11, the acceleration and deceleration are no longer performed in one sample, due to control saturation, and the system converges asymmetrically to the reference.



(a) Trajectory (b) State and control variables

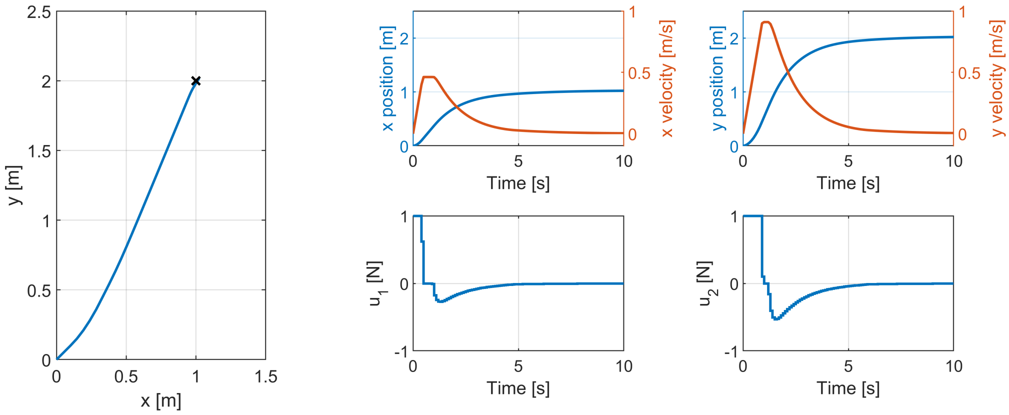
Figure 2.11: Control of pure inertial system with ℓ_1-norm cost and control limits.

LASSO Cost

Using the LASSO cost yields the results in figure 2.12, using the parameters in Table 2.4. At the start, the control is sparse, resembling that obtained with the ℓ_1-norm cost, and afterwards it is more active, like the control obtained with the quadratic cost.

Figure	N	R	R_λ	Q	Q_f	Constraints	t_{avg}
2.12	10	I	I	$50I$	$100I$	control bounds	$4.17\,\text{ms}$
2.13	10	0	I	$50I$	$100I$	control bounds	$3.98\,\text{ms}$

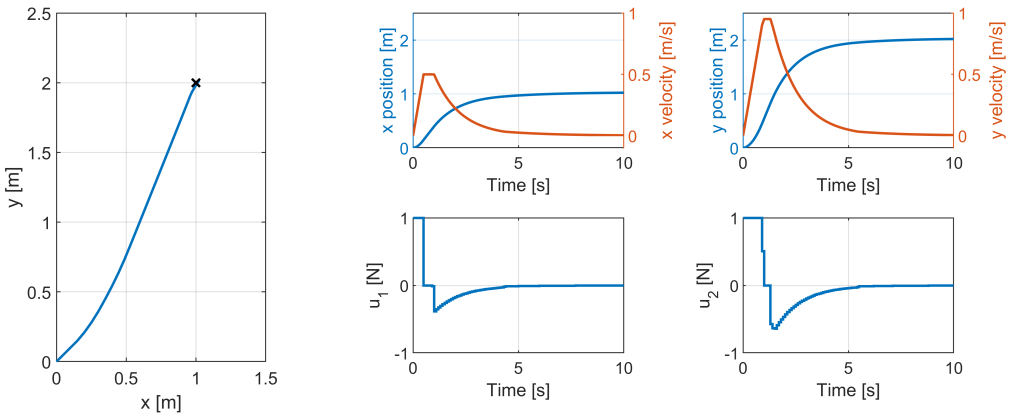
Table 2.4: Controller parameters and computation times for LASSO cost MPC experiments.



(a) Trajectory (b) State and control variables

Figure 2.12: Control of pure inertial system with LASSO cost.

In figure 2.13, the quadratic cost on the control variables is removed entirely, by setting $R = 0$. The control action is now less smooth at the beginning of the simulation.



(a) Trajectory (b) State and control variables

Figure 2.13: Control of pure inertial system with LASSO cost and zero quadratic cost for control variables.

Two Objects Experiment

The following experiment in figure 2.14 features two systems like the one used before, with a fixed-distance constraint and two obstacle constraints. Because the two objects do not fit side-by-side in the space between the objects, they rotate along their midpoint in order to pass through.

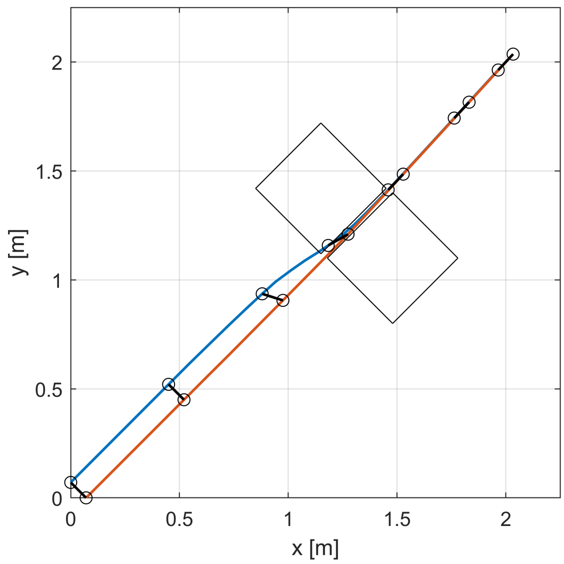


Figure 2.14: Control of two pure inertial objects with a fixed distance and obstacle constraints.

2.5.2 Unicycle

The unicycle is a vehicle with one steerable wheel and thus has restrictions on its movement, since it cannot move in any direction (non-holonomic movement). Figure 2.15 illustrates the unicycle system.

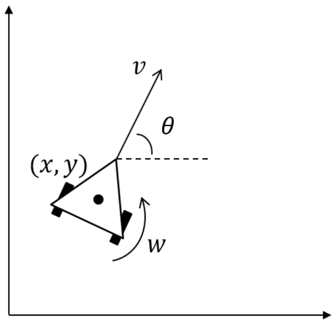


Figure 2.15: Illustration of the unicycle system.

The kinematics of this vehicle are described by a non-linear model, with the differential equations

$$\begin{aligned} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= w. \end{aligned} \tag{2.30}$$

States x and y describe the vehicle position and θ its orientation. The system inputs are the linear velocity v and angular velocity w, and so the vehicle can only move in the direction it faces, but can steer to change its orientation. Performing an Euler discretization, the discrete unicycle model becomes

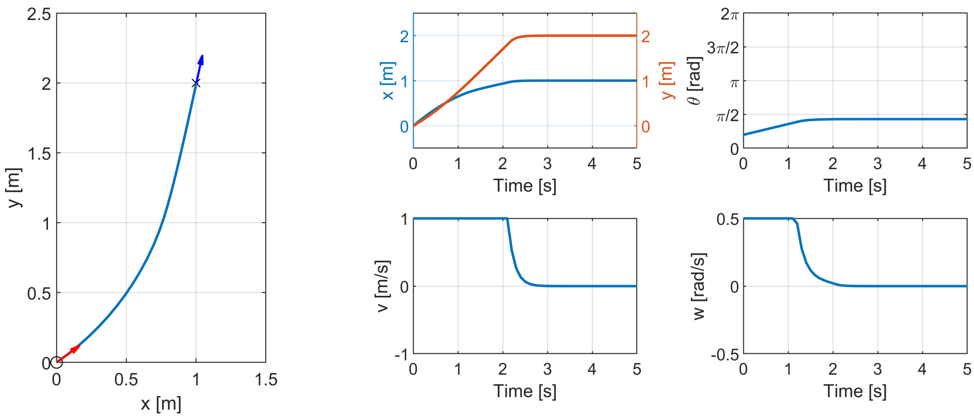
$$\begin{aligned} x(k+1) &= x(k) + T_s v_k \cos(\theta_k) \\ y(k+1) &= y(k) + T_s v_k \sin(\theta_k) \\ \theta(k+1) &= \theta_k + T_s w_k, \end{aligned} \tag{2.31}$$

where T_s is the sampling period. Note that the system model is a non-convex function and so the optimal control problem is also non-convex, meaning that found solutions may not be global minima, and different algorithms may find very different solutions.

In the following experiments a quadratic cost function is used and the controller parameters are presented in table 2.5, with $Q = diag(Q_x, Q_y, Q_\theta)$. A sampling period of 0.1 s is used. In figure 2.16, the cost of θ is disregarded and so there is no reference for the orientation. The vehicle does not travel in a straight line towards the reference due to its initial condition on θ, and so it slowly steers to face the reference point as it moves forward.

Figure	N	R	$Q_{x,y}$	Q_θ	$Q_{f_{x,y}}$	$Q_{f\theta}$	Constraints	t_{avg}
2.16	10	I	50	0	100	0	control bounds	66 ms
2.17	10	I	50	50	100	100	control bounds	70 ms
2.18	10	I	50	50	100	100	control bounds	81 ms
2.19	10	I	0	0	500	50	control bounds	51 ms
2.20	10	I	0	0	500	50	control bounds, obstacle	144 ms
2.21	15	I	0	0	500	50	control bounds, obstacle	180 ms

Table 2.5: Controller parameters and computation times for unicycle MPC experiments.

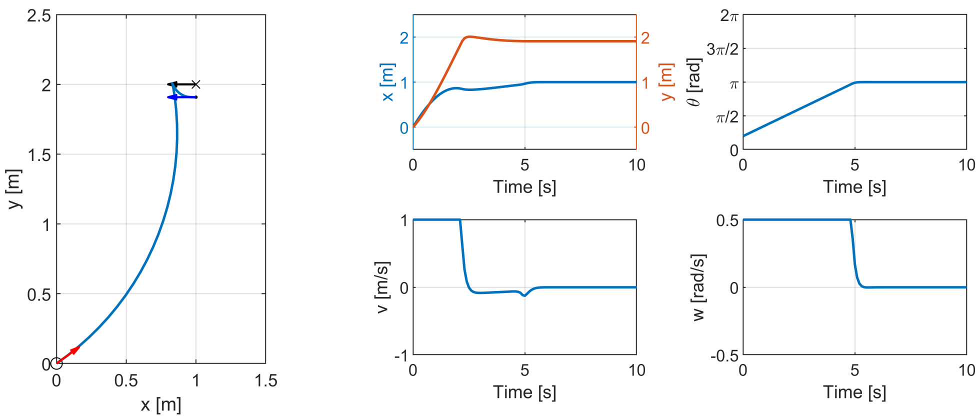


(a) Trajectory (b) State and control variables

Figure 2.16: Control of unicycle system with control constraints. Red and blue arrows represents initial and final orientations, respectively.

Including a cost and reference for θ, the result in figure 2.17 was obtained. In order to stop at the referenced orientation, the vehicle overshoots the set point to the left and then drives in reverse to it. A more intuitive manoeuvre for most drivers would have been to widen the curve and perform the whole manoeuvre going forwards, and although such a manoeuvre could be less costly, the algorithm may converge to local minima since the problem is non-convex. Furthermore, there is a static error regarding the y position, despite the fact that the system has integral action, which again is due to the non-linearity of the model; since the movement is non-holonomic, it cannot simply close the gap without a somewhat complex manoeuvre. Finally, notice that the vehicle is continuously turning in the same direction, which is due to the fact that the direction in which the vehicle must turn to reach the reference position is the same as that to reach the reference orientation.

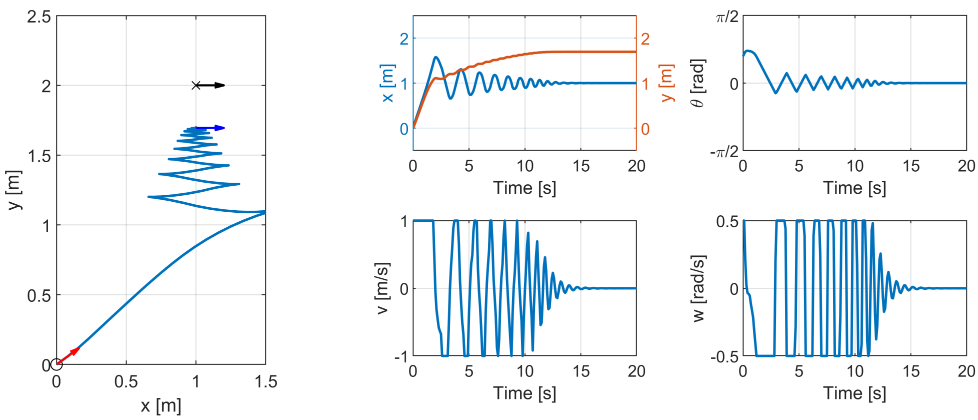
In figure 2.18, the reference orientation is now in the opposite direction, requiring a more complex



(a) Trajectory (b) State and control variables

Figure 2.17: Control of unicycle system with orientation reference and control constraints.

manoeuvre. It can be seen that the controller struggles to reach the reference, continuously going forwards and backwards while getting closer to the reference and eventually stopping with a significant static error.



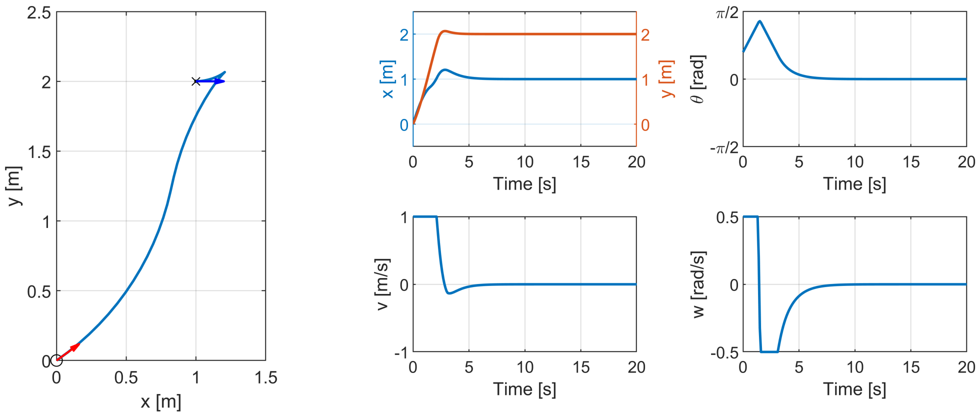
(a) Trajectory (b) State and control variables

Figure 2.18: Control of unicycle system with different orientation reference and control constraints.

To obtain a better performance, the cost of intermediate states can be ignored by setting $Q = 0$, and the terminal cost matrix Q_f increased for position states and decreased for the orientation, yielding the result in figure 2.19. This allows for the controller to better plan ahead, since it only values the tracking error at the end of the prediction horizon; for example, now it is not as penalized for not having the reference orientation while only being halfway through the manoeuvre.

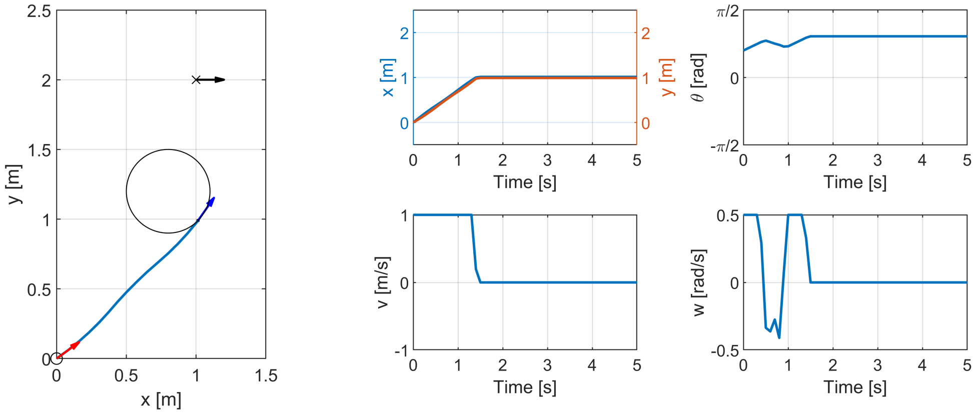
Adding an obstacle avoidance constraint, the controller presents difficulty in dodging the obstacle, and is stuck behind it due to the restrictions in its movement, as shown in figure 2.20. Increasing the prediction horizon to $N = 15$ in figure 2.21, the system can now avoid the obstacle and converge to the reference. Note that this issue may also be due to the solver used.

Notice that the average computation time is comparable to the system sampling period, and sometimes greater. This means that, with this implementation, it is not feasible to apply the controller in real-time, although that was not the objective of these experiments. Reference [27] is a survey on the



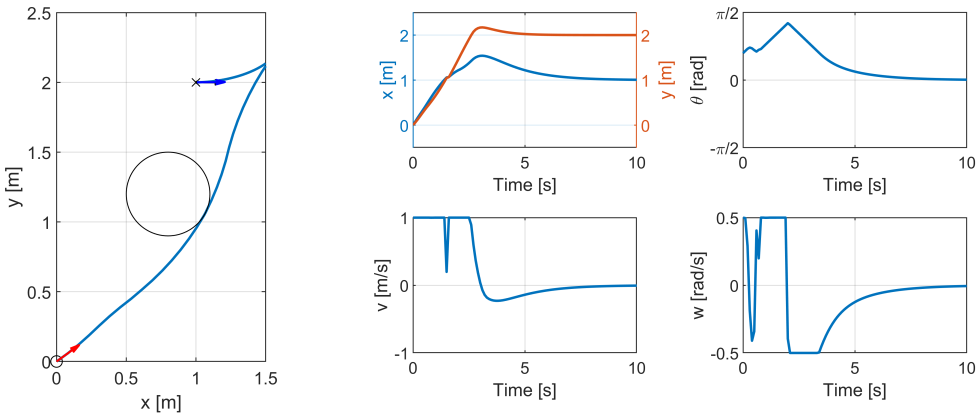
(a) Trajectory (b) State and control variables

Figure 2.19: Control of unicycle system with different orientation reference, control constraints, and tuned cost matrices.



(a) Trajectory (b) State and control variables

Figure 2.20: Control of unicycle system with control limits and obstacle constraints.



(a) Trajectory (b) State and control variables

Figure 2.21: Control of unicycle system with control limits, obstacle constraints and increased prediction horizon.

use of MPC for the control of wheeled vehicles. Typically, the controller reference is not a fixed state, but rather a time-varying trajectory, that can be generated offline or by a guidance system separate from the controller, relieving the latter from some of the computational load.

Chapter 3

Relative Orbital Mechanics

Relative orbital mechanics refers to the relative motion of two satellites orbiting the same body. In an orbital rendezvous context, when the two spacecraft are in close range it is convenient to consider relative positions and velocities, centred at one of the spacecraft, rather than absolute coordinates centred in the central body.

The equations that describe the motion of an orbiting satellite and the relative motion of two satellites are derived from Newton's law of gravitation and from his second law of motion. These result in nonlinear differential equations, which can be infeasible to solve and work with in real time applications, such as the control of a rendezvous trajectory. Hence, it is possible to approximate the nonlinear equations for the relative motion and maintain accuracy if the spacecraft are sufficiently close. For the special case of a circular orbit, these approximations result in the well known Hill equations [11], which describe a dynamical system that is linear and time-invariant. For the general case of an elliptic orbit, the approximation yields the Yamanaka-Ankersen state transition matrix [16], which describes a discrete linear time-variant (LTV) system.

In this chapter, we first introduce the nonlinear dynamics on the inertial frame in Section 3.1. In Section 3.2, a non-inertial frame of reference centred on one of the satellites is presented, and in Section 3.3 the approximated dynamics in this frame of reference are derived. Sections 3.4 and 3.5 present several simulations of free drift motions and thrust manoeuvres for circular and elliptic target orbits, respectively.

3.1 Nonlinear Inertial Dynamics

Consider two satellites, treated as point masses, in motion around a central body, such that their gravitational pull on each other is negligible. In a *rendezvous* context, one of the satellites is in free motion, designated *target* spacecraft, while the other, the *chaser*, performs manoeuvres to close their relative positions. We define the auxiliary function

$$\mathbf{f}_g(\mathbf{r}) = -\frac{\mu}{r^3}\mathbf{r}, \tag{3.1}$$

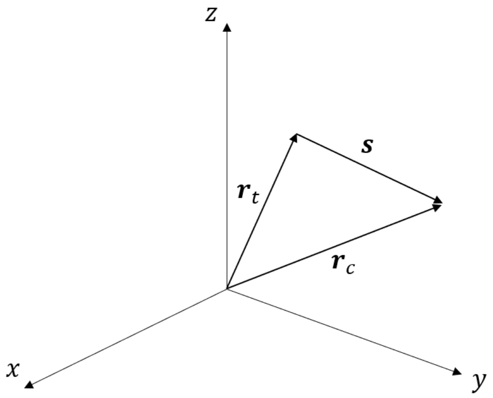


Figure 3.1: Relative position of target and chaser spacecraft.

where $\mathbf{r}$ is a position vector, r its magnitude and μ is the standard gravitational parameter. The motion of the target and chaser spacecraft, with positions $\mathbf{r}_t$ and $\mathbf{r}_c$ relative to the inertial frame of reference, is then given by

$$\ddot{\mathbf{r}}_t = \mathbf{f}_g(\mathbf{r}_t), \tag{3.2}$$

$$\ddot{\mathbf{r}}_c = \mathbf{f}_g(\mathbf{r}_c) + \frac{\mathbf{F}}{m_c}, \tag{3.3}$$

where $\mathbf{F}$ is the force vector applied by the chaser actuators, and m_c is the mass of the chaser.

The relative position between the two satellites, illustrated in figure 3.1, is defined as $\mathbf{s} = \mathbf{r}_c - \mathbf{r}_t$, and so the relative motion is

$$\ddot{\mathbf{s}} = \mathbf{f}_g(\mathbf{r}_c) - \mathbf{f}_g(\mathbf{r}_t) + \frac{\mathbf{F}}{m_c}. \tag{3.4}$$

Unlike in the case of only one unperturbed satellite, this problem has no closed-form solution and must be solved numerically or approximated with a linearisation.

3.2 Target Local Orbital Frame

When the distance between the two spacecraft is short, it is convenient to consider the non-inertial *target local orbital frame*, illustrated in figure 3.2, also known as *local-vertical/local-horizontal frame* (LVLH), centred in the target spacecraft. The axis x_{lo} is in the general direction of the velocity vector, although it is not always aligned with it, and is commonly known as V-bar. Axis y_{lo} is orthogonal to the orbital plane, in the opposite direction of the angular momentum, and is also known as H-bar. The axis z_{lo}, known as R-bar, is always directed at the center of mass of the central body. For this reason, the frame rotates with the orbital angular velocity ω, and thus is non-inertial.

To determine coordinates in this frame (F_{lo}) from ones in the inertial orbital plane frame [1], the target position $\mathbf{r}_t$ is first subtracted, and then a counter-clockwise rotation around z_{op} by the argument of perigee w and the true anomaly θ is performed. Then, another counter-clockwise rotation around the z-axis by $90\,\mathrm{deg}$ is applied, followed by a clockwise rotation around the x-axis by $90\,\mathrm{deg}$. The coordinate

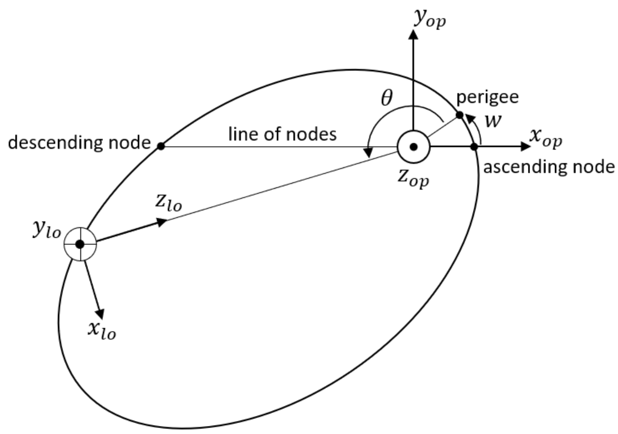


Figure 3.2: The target local orbital frame (F_{lo}).

transformation is then

$$\begin{bmatrix} x_{lo} \\ y_{lo} \\ z_{lo} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{op} - x_t \\ y_{op} - y_t \\ z_{op} - z_t \end{bmatrix} , \tag{3.5}$$

with α = $\theta + w$. The velocity vectors in an inertial frame and a rotating (*) frame with angular velocity vector $\boldsymbol{\omega}$ are related by

$$\frac{\mathrm{d}^*\mathbf{s}^*}{\mathrm{d}t} = -\boldsymbol{\omega} \times \mathbf{s}^* + \frac{\mathrm{d}\mathbf{s}}{\mathrm{d}t} , \tag{3.6}$$

This reference frame is generally used in a rendezvous context in order to represent the chaser spacecraft position and velocity relative to the target.

3.3 Approximate Equations of Relative Motion

As shown in the book by Fehse [1], applying a first order Taylor expansion to $\mathbf{f}_g(\mathbf{r}_c)$ around the target position $\mathbf{r}_t$ yields

$$\mathbf{f}_g(\mathbf{r}_c) \approx \mathbf{f}_g(\mathbf{r}_t) + \frac{\mathrm{d}\mathbf{f}_g(\mathbf{r})}{\mathrm{d}\mathbf{r}}\Big|_{\mathbf{r}=\mathbf{r}_t} (\mathbf{r}_c - \mathbf{r}_t) . \tag{3.7}$$

Applying (3.7) to the equation of the chaser motion in (3.3) and given (3.2), the relative motion is approximated by

$$\ddot{\mathbf{s}} = \frac{\mathrm{d}\mathbf{f}_g(\mathbf{r})}{\mathrm{d}\mathbf{r}}\Big|_{\mathbf{r}=\mathbf{r}_t} \mathbf{s} + \frac{\mathbf{F}}{m_c} . \tag{3.8}$$

Differentiating (3.6), the relation between the acceleration in the inertial and rotating (*) frames is

$$\frac{\mathrm{d}^2\mathbf{s}}{\mathrm{d}t^2} = \frac{\mathrm{d}^{*2}\mathbf{s}^*}{\mathrm{d}t^2} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{s}^*) + 2\boldsymbol{\omega} \times \frac{\mathrm{d}^*\mathbf{s}^*}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{s}^* , \tag{3.9}$$

where the last three terms are the centrifugal, Coriolis and Euler fictitious forces, due to the expression of the acceleration in a rotating frame.

Substituting (3.9) in equation (3.8), we get

$$\frac{\mathrm{d}^{*2}\mathbf{s}^*}{\mathrm{d}t^2} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{s}^*) + 2\boldsymbol{\omega} \times \frac{\mathrm{d}^*\mathbf{s}^*}{\mathrm{d}t} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{s}^* - \left.\frac{\mathrm{d}\mathbf{f}_g(\mathbf{r})}{\mathrm{d}\mathbf{r}}\right|_{\mathbf{r}=\mathbf{r}_t} \mathbf{s}^* = \frac{\mathbf{F}}{m_c}\ . \tag{3.10}$$

As shown in [1], after computing all the cross products and the Jacobian, the simplification of (3.10) for the general case of an elliptical orbit is

$$\ddot{x} - \omega^2 x - 2\omega\dot{z} - \dot{\omega}z + k\omega^{\frac{3}{2}}x = \frac{F_x}{m_c}, \tag{3.11a}$$

$$\ddot{y} + k\omega^{\frac{3}{2}}y = \frac{F_y}{m_c}, \tag{3.11b}$$

$$\ddot{z} - \omega^2 z + 2\omega\dot{x} + \dot{\omega}x - 2k\omega^{\frac{3}{2}}z = \frac{F_z}{m_c}\ , \tag{3.11c}$$

where $\mathbf{s}^* = [x, y, z]^\top$, $\mathbf{F} = [F_x, F_y, F_z]^\top$, and k is the constant $k = \mu/h^{\frac{3}{2}}$, with h as the magnitude of the target orbit specific angular momentum. These are known as the linearised equations of relative motion (LERM).

The set of differential equations in (3.11) is linear with respect to the relative position, velocity and acceleration, although it is not so with respect to the the angular velocity ω, which is not constant in the case of non-circular orbits. This results in the relative motion dynamics being time-variant.

Notice also that the out-of-plane motion (H-bar) in (3.11b) has become detached from the in-plane motion (V-bar and R-bar), which simplifies the problem since the two can be solved separately. This is a result of the linearisation, and for the nonlinear dynamics the two motions are in fact coupled.

These equations are the result of linear approximations, with respect to position, of the real non-linear motion, and are only accurate if the distance between the target and the center of mass of the central body is significantly greater than the distance between the target and chaser spacecraft.

3.3.1 Circular Orbit Case

For the special case of a circular target orbit, the orbital angular velocity is constant $\omega = \mu/r_t^3$, and so $\dot{\omega} = 0$. Since $h = \omega r^2$, we then get $k = w^{1/2}$. Substituting, equations (3.11a-c) simplify to

$$\ddot{x} - 2\omega\dot{z} = \frac{F_x}{m_c}, \tag{3.12a}$$

$$\ddot{y} + \omega^2 y = \frac{F_y}{m_c}, \tag{3.12b}$$

$$\ddot{z} + 2\omega\dot{x} - 3\omega^2 z = \frac{F_z}{m_c}\ . \tag{3.12c}$$

This system of linear differential equations is known as the Hill equations [11], and also sometimes as the Hill-Clohessy-Wiltshire equations [12]. They describe a linear and time invariant dynamical system, which can be represented in state-space with the model

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \ddot{x} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2\omega \\ 0 & 3\omega^2 & -2\omega & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix} \begin{bmatrix} F_x \\ F_z \end{bmatrix} , \tag{3.13}$$

for the in-plane dynamics, and

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_c} \end{bmatrix} \begin{bmatrix} F_y \end{bmatrix} , \tag{3.14}$$

for the out-of-plane motion. The closed-form solution for the Hill equations, known as the Clohessy-Wiltshire equations, is trivial and can be found along with their derivation in [1].

3.3.2 Simplification of the General Equations

The approximate equations of relative motion in (3.11) may be simplified in order to help find a solution, by changing the independent variable from time to the true anomaly, and by applying a coordinate change. With the chain rule, the time derivative of a variable r is related to its derivative in respect to the true anomaly θ by

$$\frac{dr}{dt} = \frac{dr}{d\theta}\frac{d\theta}{dt} . \tag{3.15}$$

Given that the time derivative of the true anomaly θ is the orbital angular velocity ω, and denoting the derivative in respect to θ by r', the above expression simplifies to

$$\dot{r} = \omega r' , \tag{3.16}$$

and the second time derivative is

$$\ddot{r} = \omega^2 r'' + \omega\omega' r' . \tag{3.17}$$

Applying the derivatives (3.16) and (3.17) to the positions and the angular velocity in (3.11) yields

$$\omega^2 x'' + \omega\omega' x' + (k\omega^{\frac{3}{2}} - \omega^2)x - 2\omega^2 z' - \omega\omega' z = \frac{F_x}{m_c}, \tag{3.18a}$$

$$\omega^2 y'' + \omega\omega' y' + k\omega^{\frac{3}{2}} y = \frac{F_y}{m_c}, \tag{3.18b}$$

$$\omega^2 z'' + \omega\omega' z' - (2k\omega^{\frac{3}{2}} + \omega^2)z + 2\omega^2 x' + \omega\omega' x = \frac{F_z}{m_c} . \tag{3.18c}$$

The orbital angular velocity can be rewritten as

$$\omega = (1 + e\cos\theta)^2 \frac{\mu^2}{h^3} . \tag{3.19}$$

Defining the auxiliary parameter

$$\rho(\theta) = 1 + e\cos(\theta)\ , \tag{3.20}$$

and since $k = \mu/h^{3/2}$, the angular velocity becomes

$$\omega = k^2\rho^2\ , \tag{3.21}$$

and its derivative with respect to θ is

$$\omega' = -2k^2\rho e\sin(\theta)\ . \tag{3.22}$$

Substituting with (3.21) and (3.22), (3.18) becomes

$$\rho x'' - 2e\sin\theta x' - e\cos\theta x - 2\rho z' + 2e\sin\theta z = \frac{F_x}{m_c k^4\rho^3}, \tag{3.23a}$$

$$\rho y'' - 2e\sin\theta y' + y = \frac{F_y}{m_c k^4\rho^3}, \tag{3.23b}$$

$$\rho z'' - 2\sin\theta z' - (3 + e\cos\theta)z + 2\rho x' - 2e\sin\theta x = \frac{F_z}{m_c k^4\rho^3}\ . \tag{3.23c}$$

Applying the coordinate transformation

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\end{bmatrix} = \rho(\theta)\begin{bmatrix}x\\ y\\ z\end{bmatrix}\ , \tag{3.24}$$

the derivatives become

$$\begin{bmatrix}\tilde{x}'\\ \tilde{y}'\\ \tilde{z}'\end{bmatrix} = \rho(\theta)\begin{bmatrix}x'\\ y'\\ z'\end{bmatrix} - e\sin\theta\begin{bmatrix}x\\ y\\ z\end{bmatrix}\ , \tag{3.25}$$

and the second derivatives

$$\begin{bmatrix}\tilde{x}''\\ \tilde{y}''\\ \tilde{z}''\end{bmatrix} = \rho(\theta)\begin{bmatrix}x''\\ y''\\ z''\end{bmatrix} - 2e\sin\theta\begin{bmatrix}x'\\ y'\\ z'\end{bmatrix} - e\cos\theta\begin{bmatrix}x\\ y\\ z\end{bmatrix}\ . \tag{3.26}$$

Writing (3.23) as a function of the second derivatives, substituting them in (3.26) and applying the transformations (3.24) and (3.25) yields the simplified equations of relative motion in the domain of θ

$$\tilde{x}'' - 2\tilde{z}' = \frac{F_x}{m_c k^4\rho^3}, \tag{3.27a}$$

$$\tilde{y}'' + \tilde{y} = \frac{F_y}{m_c k^4\rho^3}, \tag{3.27b}$$

$$\tilde{z}'' - \frac{3}{\rho}\tilde{z} + 2\tilde{x}' = \frac{F_x}{m_c k^4\rho^3}\ . \tag{3.27c}$$

These are known as the Tschauner–Hempel equations [14] (also sometimes known as the Lawden equations [28]), and are an easier set of ordinary differential equations to solve than (3.11).

3.3.3 Homogeneous Solution

A simple homogeneous solution ($\mathbf{F} = 0$) to (3.27), in the form of a state transition matrix, was introduced by Yamanaka and Ankersen [16], and further detailed by Ankersen [17]. The transition matrix propagates the state in the domain of the true anomaly from an initial θ_0, at time t_0, to the state at θ_t, for time t.

First, given initial conditions on position and velocity at time t_0, the transformed position and velocities must be determined for $\theta = \theta_0$ with (3.24) for the position, and for the velocity

$$\begin{bmatrix} \tilde{x}' \\ \tilde{y}' \\ \tilde{z}' \end{bmatrix} = -e\sin\theta \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \frac{1}{k^2\rho(\theta)} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} . \tag{3.28}$$

In matrix form, the transformations become

$$\begin{bmatrix} \tilde{x} \\ \tilde{z} \\ \tilde{x}' \\ \tilde{z}' \end{bmatrix} = \underbrace{\begin{bmatrix} \rho(\theta) & 0 & 0 & 0 \\ 0 & \rho(\theta) & 0 & 0 \\ -e\sin\theta & 0 & \frac{1}{k^2\rho(\theta)} & 0 \\ 0 & -e\sin\theta & 0 & \frac{1}{k^2\rho(\theta)} \end{bmatrix}}_{\Lambda_i(\theta)} \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} , \quad \begin{bmatrix} \tilde{y} \\ \tilde{y}' \end{bmatrix} = \underbrace{\begin{bmatrix} \rho(\theta) & 0 \\ -e\sin\theta & \frac{1}{k^2\rho(\theta)} \end{bmatrix}}_{\Lambda_o(\theta)} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} . \tag{3.29}$$

Afterwards, the so-called pseudo-initial conditions must be computed for the in-plane motion with

$$\begin{bmatrix} \bar{x}_0 \\ \bar{z}_0 \\ \bar{x}'_0 \\ \bar{z}'_0 \end{bmatrix} = \underbrace{\frac{1}{1-e^2} \begin{bmatrix} 1-e^2 & 3es(1/\rho + 1/\rho^2) & -es(1+1/\rho) & -ec+2 \\ 0 & -3s(1/\rho + e^2/\rho^2) & s(1+1/\rho) & c-2e \\ 0 & -3(c/p+e) & c(1+1/\rho)+e & -s \\ 0 & 3\rho + e^2 - 1 & -\rho^2 & es \end{bmatrix}_{\theta_0}}_{\phi_i^{-1}(\theta_0)} \begin{bmatrix} \tilde{x}_0 \\ \tilde{z}_0 \\ \tilde{x}'_0 \\ \tilde{z}'_0 \end{bmatrix} , \tag{3.30}$$

with $s(\theta) = \rho\sin\theta$ and $c(\theta) = \rho\cos\theta$. Note that, for simplification, dependencies on θ for ρ, s and c were omitted, however these parameters must be computed for $\theta = \theta_0$. For the out-of-plane motion, no pseudo-initial conditions are computed.

The state at a time t with true anomaly θ_t can be computed from the state at time t_0 with a transition matrix, and so we have for the in-plane motion

$$\begin{bmatrix} \tilde{x}_t \\ \tilde{z}_t \\ \tilde{x}'_t \\ \tilde{z}'_t \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -c(1+1/\rho) & s(1+1/\rho) & 3\rho^2 J \\ 0 & s & c & (2-3esJ) \\ 0 & 2s & 2c-e & 3(1-2esJ) \\ 0 & s' & c' & -3e(s'J + s/\rho^2) \end{bmatrix}_{\theta_t}}_{\phi_i(\theta_t)} \begin{bmatrix} \bar{x}_0 \\ \bar{z}_0 \\ \bar{x}'_0 \\ \bar{z}'_0 \end{bmatrix} , \tag{3.31}$$

and for the out-of-plane motion

$$\begin{bmatrix} \tilde{y}_t \\ \tilde{y}'_t \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta_t - \theta_0) & \sin(\theta_t - \theta_0) \\ -\sin(\theta_t - \theta_0) & \cos(\theta_t - \theta_0) \end{bmatrix}}_{\phi_o(\theta_0, \theta_t)} \begin{bmatrix} \tilde{y}_0 \\ \tilde{y}'_0 \end{bmatrix} \tag{3.32}$$

with $s' = \cos\theta + e\cos 2\theta$, $c' = -(\sin\theta + e\sin 2\theta)$ and $J = k^2(t - t_0)$.

The transformed position and velocity at time t must then be reverted. The inverse transformation for the position is then

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\rho(\theta)} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} , \tag{3.33}$$

and for the velocity

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = k^2 e \sin\theta \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} + k^2 \rho(\theta) \begin{bmatrix} \tilde{x}' \\ \tilde{y}' \\ \tilde{z}' \end{bmatrix} . \tag{3.34}$$

In matrix form, the transformation is

$$\begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{\rho(\theta)} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho(\theta)} & 0 & 0 \\ k^2 e \sin\theta & 0 & k^2\rho(\theta) & 0 \\ 0 & k^2 e \sin\theta & 0 & k^2\rho(\theta) \end{bmatrix}}_{\Lambda_i^{-1}(\theta)} \begin{bmatrix} \tilde{x} \\ \tilde{z} \\ \tilde{x}' \\ \tilde{z}' \end{bmatrix} , \quad \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{\rho(\theta)} & 0 \\ k^2 e \sin\theta & k^2\rho(\theta) \end{bmatrix}}_{\Lambda_o^{-1}(\theta)} \begin{bmatrix} \tilde{y} \\ \tilde{y}' \end{bmatrix} . \tag{3.35}$$

The true anomaly at time t can be computed from t_0 and θ_0 by first computing the eccentric anomaly E at time t_0 with

$$E = \operatorname{arctan2}\left(\sqrt{1 - e^2}\sin\theta,\ e + \cos\theta\right), \tag{3.36}$$

and then calculating the mean anomaly M at time t_0 with Kepler's equation

$$M = E - e\sin E. \tag{3.37}$$

The mean anomaly at time t can be determined with

$$M_t = M_0 + \frac{2\pi}{T}(t - t_0) , \tag{3.38}$$

where T is the target orbital period, and the eccentric anomaly E_t can then be obtained by solving Kepler's equation (3.37) w.r.t the eccentric anomaly E. Finally, the true anomaly θ_t is computed with

$$\theta = 2\arctan\left(\sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}\right). \tag{3.39}$$

Summarizing, to determine the state at time t from the state at time t_0 one must

- Compute the transformed position and velocity with (3.24) and (3.28) for $\theta = \theta_0$;
- Compute the pseudo-initial conditions for the in-plane motion with (3.30);
- Compute true anomaly θ_t at time t;
- Apply the transition matrices for in-plane and out-of-plane motions with (3.31) and (3.32);
- Revert coordinate transformations with (3.33) and (3.34) for $\theta = \theta_t$.

These transition matrices can be used as the dynamic matrix of a discrete linear time-variant system, setting the sampling period as $T_s = t - t_0$, and so they allow to easily simulate the relative motion between the target and chaser spacecraft in the LVLH frame for an elliptical orbit. These, however, only model free drift motions, that is, in the absence of input forces. The particular solution for this problem in the domain of the true anomaly θ is presented in the next section.

3.3.4 Particular Solution

Rewriting the system of equations (3.27) in state space form as $x' = Ax + Bu$, the particular solution is obtained with

$$x_p(\theta_t) = \int_{\theta_0}^{\theta_t} \Phi(\theta)B(\theta)u(\theta)d\theta \; , \tag{3.40}$$

where Φ represents the transition matrices presented in the previous section; for the in-plane motion $\Phi_i(\theta_0, \theta_t) = \phi_i(\theta_t)\phi_i^{-1}(\theta_0)$, and for the out-of-plane motion $\Phi_o(\theta_0, \theta_t) = \phi_o(\theta_0, \theta_t)$. Also, for the in-plane motion we have

$$B_i(\theta) = \frac{1}{m_c k^4 \rho^3(\theta)} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad u_i(\theta) = \begin{bmatrix} F_x(\theta) \\ F_z(\theta) \end{bmatrix} \; , \tag{3.41}$$

and for the out-of-plane motion

$$B_o(\theta) = \frac{1}{m_c k^4 \rho^3(\theta)} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and} \quad u_o(\theta) = F_y(\theta) \; . \tag{3.42}$$

The solution requires computing several non-trivial integrals, which has already been done in [17] for the case where the force is constant along the propagation interval ($\mathbf{F}(\theta) = \mathbf{F}$). The solution presented there for the in-plane motion is

$$\begin{bmatrix} \tilde{x}_p \\ \tilde{z}_p \\ \tilde{x}'_p \\ \tilde{z}'_p \end{bmatrix} = \underbrace{\Phi_i(\theta_0, \theta_t) \frac{1}{k^4(1-e^2)} \begin{bmatrix} 3I_{1J} - e(I_{s_3} + I_{s_2}) & 2I_3 - e(I_{c_2} + 3I_{s_2 J}) \\ I_{s_3} + I_{s_2} - 3eI_{1J} & I_{c_2} - e(2I_3 - 3eI_{s_2 J}) \\ I_{c_3} + I_{c_2} + eI_3 & -I_{s_2} \\ -I_1 & eI_{s_2} \end{bmatrix} \frac{1}{m_c}}_{\Gamma_i(\theta_0, \theta_t)} \begin{bmatrix} F_x \\ F_z \end{bmatrix} \; , \tag{3.43}$$

and for the out-of-plane motion is

$$\begin{bmatrix} \tilde{y}_p \\ \tilde{y}'_p \end{bmatrix} = \underbrace{\frac{1}{k^4} \begin{bmatrix} \sin(\theta_t) & \cos(\theta_t) \\ \cos(\theta_t) & -\sin(\theta_t) \end{bmatrix} \begin{bmatrix} I_{c_3} \\ -I_{s_3} \end{bmatrix} \frac{1}{m_c}}_{\Gamma_o(\theta_0, \theta_t)} F_y \ , \tag{3.44}$$

where the variables I_i are integrals, the values of which are presented in Appendix A.

To obtain the full solution, the particular solution is added to the homogeneous solution. Merging the in-plane and out-of-plane motions yields the full solution

$$\begin{bmatrix} \tilde{x}_t \\ \tilde{y}_t \\ \tilde{z}_t \\ \tilde{x}'_t \\ \tilde{y}'_t \\ \tilde{z}'_t \end{bmatrix} = \Phi(\theta_0, \theta_t) \begin{bmatrix} \tilde{x}_0 \\ \tilde{y}_0 \\ \tilde{z}_0 \\ \tilde{x}'_0 \\ \tilde{y}'_0 \\ \tilde{z}'_0 \end{bmatrix} + \Gamma(\theta_0, \theta_t) \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} (\theta_0), \tag{3.45}$$

where $\Phi(\cdot\, , \cdot)$ and $\Gamma(\cdot\, , \cdot)$ are appropriately generated from the entries of Φ_i, Φ_o, Γ_i and Γ_o. Inverting the coordinate transformation to obtain the solution in the time-domain yields

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{bmatrix} = \Lambda^{-1}(\theta_t)\Phi(\theta_0, \theta_t)\Lambda(\theta_0) \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} + \Lambda^{-1}(\theta_t)\Gamma(\theta_0, \theta_t) \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} . \tag{3.46}$$

State-Space Model

Equation (3.46) can be used as a discrete space-state model of a linear and time-variant system

$$\mathbf{x}_{k+1} = A_k^{k+1}\mathbf{x}_k + B_k^{k+1}\mathbf{u}_k, \tag{3.47}$$

where the state vector is $\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^\top$ and the input vector $\mathbf{u} = [F_x, F_y, F_z]^\top$, and such that the system at time k has true anomaly θ_0 and at time $k+1$ the true anomaly θ_t. Matrix A_k^{k+1} is the state transition matrix from time k to $k+1$, and from (3.46) is defined as

$$A_k^{k+1} = \Lambda^{-1}(\theta_t)\Phi(\theta_0, \theta_t)\Lambda(\theta_0), \tag{3.48}$$

while B_k^{k+1} is the input matrix, which becomes

$$B_k^{k+1} = \Lambda^{-1}(\theta_t)\Gamma(\theta_0, \theta_t). \tag{3.49}$$

Note that, since for the particular solution a constant force between sampling intervals is assumed, this discretized model constitutes a ZOH discretization of the relative dynamics.

3.4 Relative Motion in a Circular Target Orbit

In this section and the next, the relative motion between two satellites is simulated, considering circular and elliptical target orbits, respectively. Both free drift motions and impulsive thrust manoeuvres are presented.

The relative motion for a circular target orbit can be simulated with the Hill linearised dynamics, and can either be numerically simulated in Simulink, with the continuous state-space model in (3.13) and (3.14), or analytically simulated with its solution, the Clohessy-Wiltshire equations. Because numerical simulation introduces a discretization error, it is preferable to use the analytical solution. All simulations are performed for Earth satellites with a target orbit height of 600 km from the surface, simulated for two orbital periods and with a sampling period of 10 s. The orbit orientation parameters are disregarded, since a uniform gravitational field is assumed, and the Hill equations describe the relative motion on the orbital plane.

Despite all satellite trajectories being circles or ellipses on the inertial frame, the relative trajectories on this frame may be non-intuitive, due to it being non-inertial. On the R-bar/V-bar plots, the target spacecraft is always on the origin; hence, if the chaser travels along the negative direction of R-bar (up) then it gains altitude relative to the target, and if it travels along the positive direction of V-bar (left) then it gets ahead of the target in its orbit. Because the in-plane and out-of-plane motions are decoupled in the linearised model, they will be shown separately. The chaser initial position is marked with '$\times$', while the target position (origin) is marked with 'O'.

3.4.1 Free Drift Motions

To begin with, the motion with no action from the chaser actuators is considered, being simulated with different initial conditions for position and velocity.

If the chaser is on V-bar (R-bar = 0) with the same velocity as the target, it will not move relative to it, as shown in figure 3.3, because in these conditions the spacecraft are simply on a different phase of approximately the same circular orbit.

If the chaser is at a different altitude and with zero relative velocity then it is necessarily on an elliptic orbit, since higher/lower circular orbits have lower/higher orbital velocity. Hence, as seen in figure 3.4, if the target starts at a higher altitude, it will gain altitude until it reaches apogee, and then come back to perigee. On the other hand, if it starts lower, it will lose altitude and gain velocity until it reaches perigee, and then climb back up to apogee. Moreover, the chaser at a higher orbit will have a greater orbital period, and so will fall behind the target, while the one with the lower orbit will get ahead.

If the horizontal velocity is compensated as to ensure the chaser is on a circular orbit, it will now just drift along a constant altitude, as shown in figure 3.5. For a difference in orbit heights of z_0, the chaser

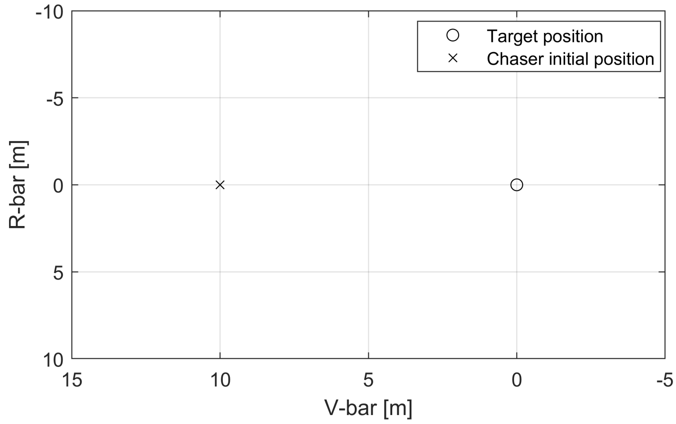


Figure 3.3: Relative in-plane motion on a circular target orbit with V-bar start. Initial conditions $s_0 = [10, 0, 0]$ m, $\dot{s}_0 = [0, 0, 0]$ m/s.

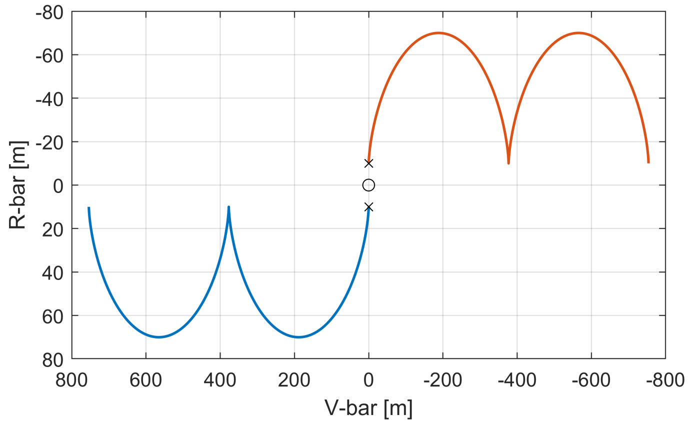


Figure 3.4: Relative in-plane motion on a circular target orbit with R-bar start. Initial conditions $s_0 = [0, 0, \pm 10]$ m, $\dot{s}_0 = [0, 0, 0]$ m/s.

relative horizontal velocity that generates a circular chaser orbit is $V_x = \frac{3}{2}\omega z_0$.

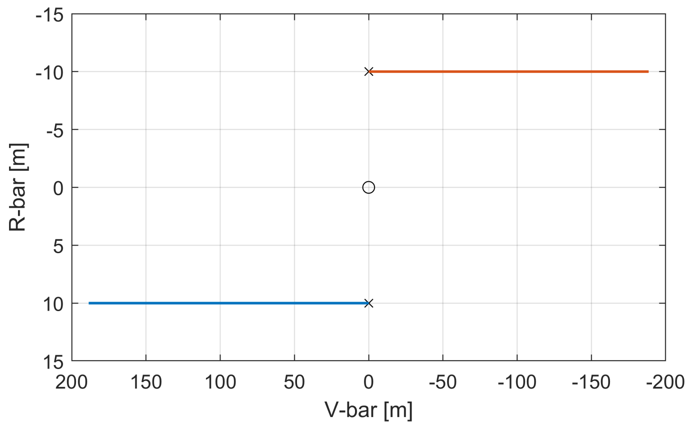


Figure 3.5: Relative in-plane motion in a circular target orbit with R-bar start and circular chaser orbit. Initial conditions $s_0 = [0, 0, \pm 10]$ m, $\dot{s}_0 = [\pm 10\frac{3}{2}\omega, 0, 0]$ m/s.

In figure 3.6, the chaser starts at V-bar with a lower radial velocity, which causes it to be on an elliptic orbit, however with the same orbital period as the target. This causes the chaser to drop down and get ahead of the target, and then looping around to the the same initial relative position.

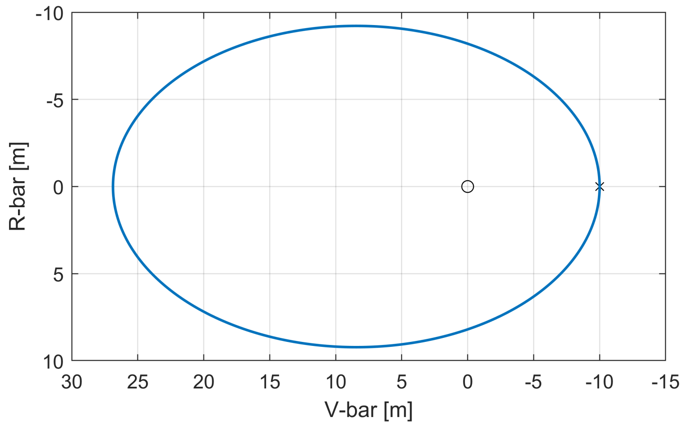


Figure 3.6: Relative in-plane motion in a circular target orbit with V-bar start and radial relative velocity. Initial conditions $s_0 = [-10, 0, 0]$ m, $\dot{s}_0 = [0, 0, 0.01]$ m/s.

If the chaser starts on V-bar with a higher horizontal velocity, it will get ahead of the target and gain altitude. As it does, the chaser loses velocity and starts falling behind, until it reaches apogee. It will then lose altitude and gain velocity, and then start catching up again as it reaches perigee. This causes the loops seen in figure 3.7 around perigee. If the chaser starts with a lower horizontal velocity, the trajectory will be mirrored.

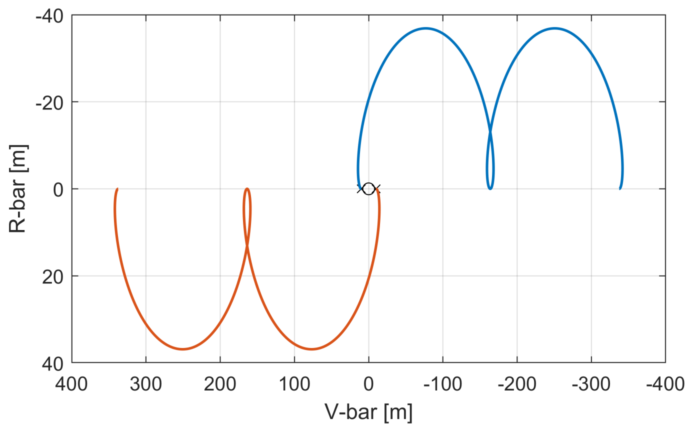


Figure 3.7: Relative in-plane motion in a circular target orbit with V-bar start and horizontal relative velocity. Initial conditions $s_0 = [\pm 10, 0, 0]$ m, $\dot{s}_0 = [\pm 0.01, 0, 0]$ m/s.

The out-of-plane motion without any actuation will always be sinusoidal with respect to time. In figure 3.8, the chaser starts above the target orbital plane (H-bar=0), with the same normal velocity. It will then decrease along H-bar and cross the target orbital plane at the ascending node, until it reaches the opposite distance with which it started. It will then increase and intersect the orbital plane at the descending node, until it reaches the initial position.

3.4.2 Impulsive Thrust Manoeuvres

We will now consider manoeuvres in which instant changes in velocity (ΔV) are applied by the chaser actuators along the trajectory. Note that in reality these instant ΔV's are impossible, since any spacecraft thrusters can only generate a gradual increase in velocity. Nevertheless, it is useful to consider this type

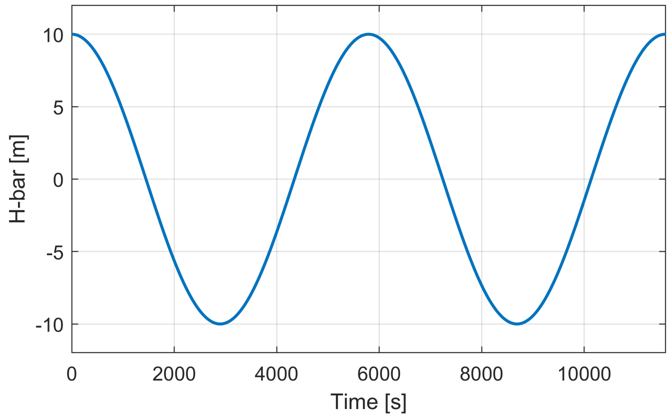


Figure 3.8: Relative out-of-plane motion in a circular target orbit. Initial conditions $s_0 = [0, 10, 0]$ m, $\dot{s}_0 = [0, 0, 0]$ m/s.

of manoeuvres for planning rendezvous trajectories. The deduction of the expressions for the ΔV's can be found in [1].

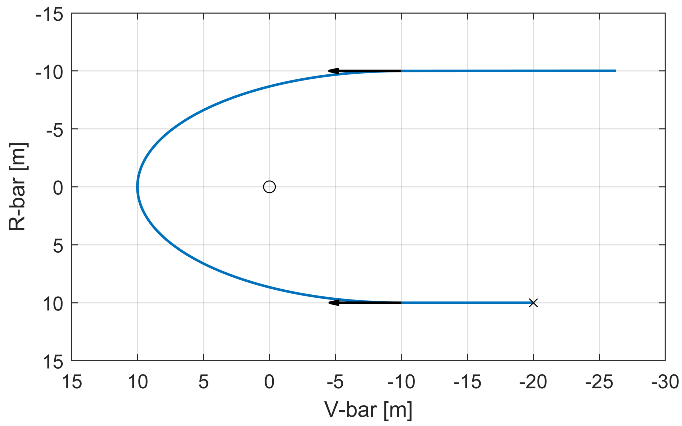


Figure 3.9: Hohmann transfer manoeuvre in a circular target orbit.

In figure 3.9 a Hohmann transfer manoeuvre is shown relative to the target, in which two ΔV's (presented as black arrows) are applied in order to change the altitude of the chaser by Δz. The chaser starts on a circular orbit below the target and then increases its horizontal velocity by

$$\Delta V_x = \Delta z \omega / 4, \tag{3.50}$$

which changes its orbit to an eccentric transfer trajectory with the apogee at the desired altitude. At apogee, the same ΔV is applied in order to circularize the orbit. Thus, it takes half an orbital period to complete this manoeuvre.

One way to perform a V-bar transfer, in which the chaser advances or retreats by Δx, is with two radial ΔV's with magnitude

$$\Delta V_z = \Delta x \omega / 4, \tag{3.51}$$

exploiting a trajectory such as the one seen in figure 3.6. In figure 3.10, the chaser is on the same orbit but behind the target, which causes it to move below and ahead of the target. As it crosses V-bar, it executes the same ΔV to circularize the orbit and remain stationary relative to the target.

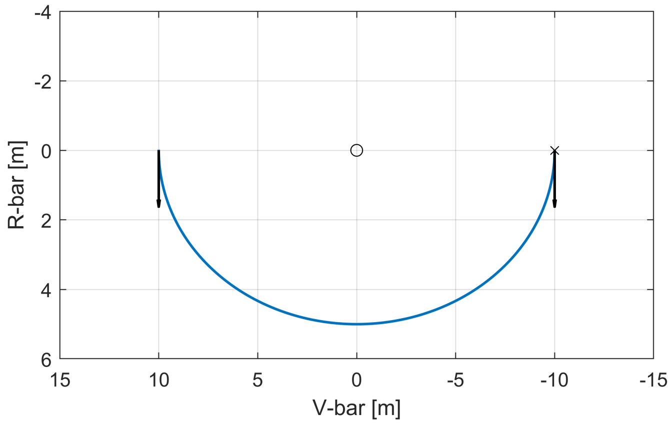


Figure 3.10: Radial V-bar transfer manoeuvre with radial impulses in a circular target orbit.

Another possible V-bar transfer manoeuvre is with the use of horizontal ΔV's (instead of radial) with magnitude

$$\Delta V_x = -\Delta x \frac{\omega}{6\pi}, \tag{3.52}$$

exploiting a trajectory like the one in figure 3.7, and resulting in the manoeuvre in figure 3.11. This manoeuvre takes one full orbital period to complete, as opposed to half a period with the radial impulses, but costs $3\pi/2$ times less, which is very significant.

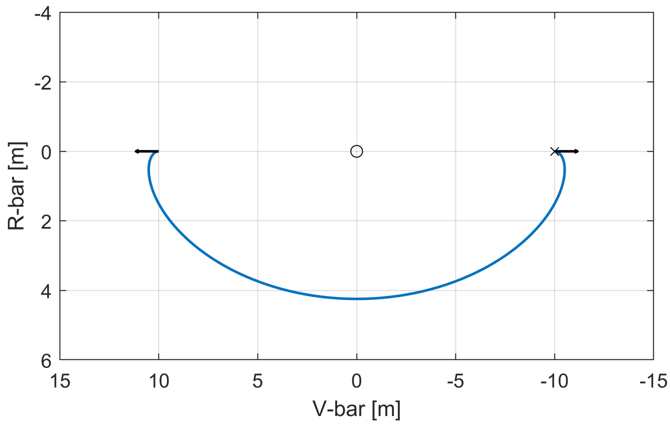


Figure 3.11: Radial V-bar transfer manoeuvre with horizontal impulses in a circular target orbit.

To fully correct a chaser inclination difference in respect to the target orbit requires a normal ΔV at either the ascending or descending node. If the amplitude of the out-of-plane motion is Δy, then the required ΔV is

$$\Delta V_y = -\omega \Delta y. \tag{3.53}$$

Figure 3.12 shows an example of this manoeuvre.

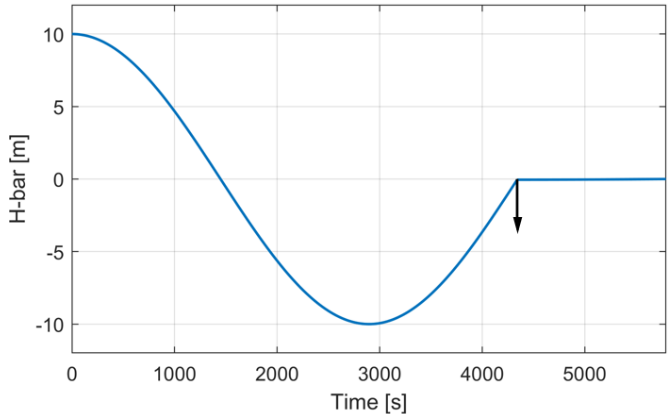


Figure 3.12: Inclination correction manoeuvre in a circular target orbit.

3.5 Relative Motion in an Elliptic Target Orbit

The relative motion in an elliptic target orbit was simulated using the discrete state-space model in (3.47). Due to the fact that the orbit is elliptic, the initial true anomaly θ_0 now has to be defined as an initial condition. As before, the simulations are performed for an Earth satellite, with a perigee height of 600 km from the surface, and simulated for two orbital periods and a sampling anomaly Θ_s of $0.5\,\mathrm{deg}$. The eccentricity of the orbit will vary between experiments. As before, the orbit orientation parameters have been disregarded.

The relative motion in an elliptic orbit is significantly more complicated and harder to comprehend. One of the reasons for this is that for elliptic orbits the LVLH frame rate of rotation varies along the orbit, while for circular orbits it rotates uniformly. Also note that, for elliptic orbits, the V-bar axis is not always aligned with the target velocity vector, unlike in the case of circular orbits.

3.5.1 Free Drift Motions

As for the circular case, the motions in the absence of thrust will first be explored. With a V-bar start and zero initial relative velocity, it can be seen in figure 3.13 that the chaser drifts away from the target, unlike what happens for the same conditions in a circular orbit, where it would stay stationary (figure 3.3). This happens because the magnitude of the orbital velocity is not constant along an elliptic orbit, which means that if the chaser is ahead and with the same velocity, then it is not exactly on the same orbit as the target, and so the spacecraft present some relative motion. Also note that, since V-bar is not necessarily aligned with the target velocity vector, being on the same orbit may require the chaser to be on a different R-bar position. It may also be observed that the drift trajectory changes with the position of the target along its orbit, since the dynamics change with the true anomaly θ; if the target starts at perigee ($\theta_0 = 0\,\mathrm{deg}$) or apogee ($\theta_0 = 180\,\mathrm{deg}$), the drift happens only along V-bar, otherwise the chaser also drifts in R-bar. Increasing the eccentricity, the drift motions now have greater amplitudes, as shown in figure 3.14.

If the initial relative velocity is compensated as to ensure the spacecraft are on the same orbit, but

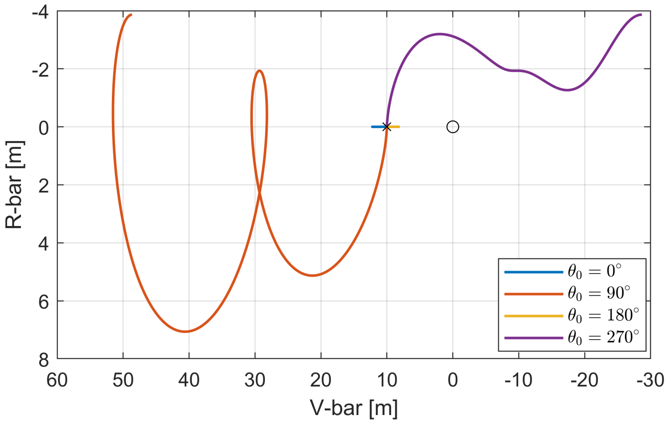


Figure 3.13: Relative in-plane motion in an elliptic target orbit with V-bar start and various initial true anomalies. Initial conditions $s_0 = [10, 0, 0]\,\mathrm{m}$ and $\dot{s}_0 = [0, 0, 0]\,\mathrm{m/s}$, with eccentricity $e = 0.1$.

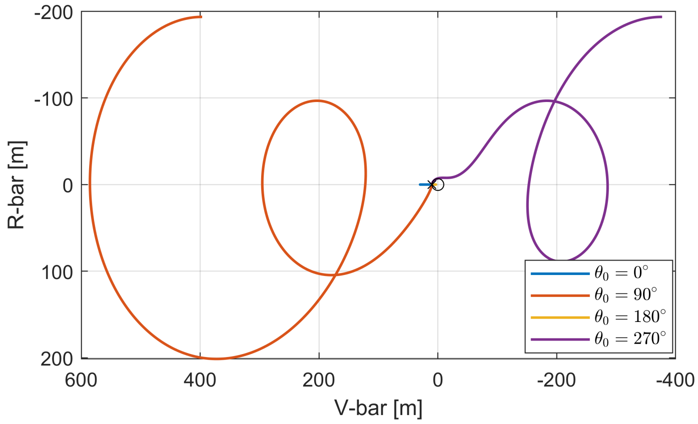


Figure 3.14: Relative in-plane motion in an elliptic target orbit with V-bar start and different initial true anomalies. Initial conditions $s_0 = [10, 0, 0]\,\mathrm{m}$ and $\dot{s}_0 = [0, 0, 0]\,\mathrm{m/s}$, with eccentricity $e = 0.5$.

with different true anomalies, the results from figure 3.15 are obtained, where two different eccentricities are simulated. To generate these initial conditions, the chaser and target positions were defined on the orbital plane frame with the orbit elements, on the same orbit but with different phases, and then transformed to the target local orbital frame with the transformation (3.5), as well as (3.6) for the velocity. Despite being on the same orbit, the spacecraft still move relative to each other, although it can be observed that, due to the fact that the spacecraft are on the same orbit and thus have the the same orbital period, the chaser returns to the initial relative position after one orbit. It can be again observed that, a higher eccentricity leads to a greater amplitude of the relative motion.

The motion along R-bar is due to the fact that, as mentioned before, the V-bar axis is not always aligned with the velocity vector, as can be observed in figure 3.2, and so it moves relative to it along the orbit: at perigee they are aligned, and then V-bar lowers toward the inside of the ellipse until it reaches perigee and they become aligned again, after which V-bar raises toward the outside of the ellipse. Since in figure 3.15 the target starts at perigee, the chaser first decreases in R-bar, and then it comes back up until it reaches perigee and increases R-bar. On the other hand, the motion along V-bar is due to the varying orbital velocity. As the spacecraft move toward apogee they lose velocity, and because the chaser is ahead the target catches up with it and closes their relative distance. When moving toward perigee, the spacecraft speed up and the chaser gets ahead of the target, increasing along V-bar.

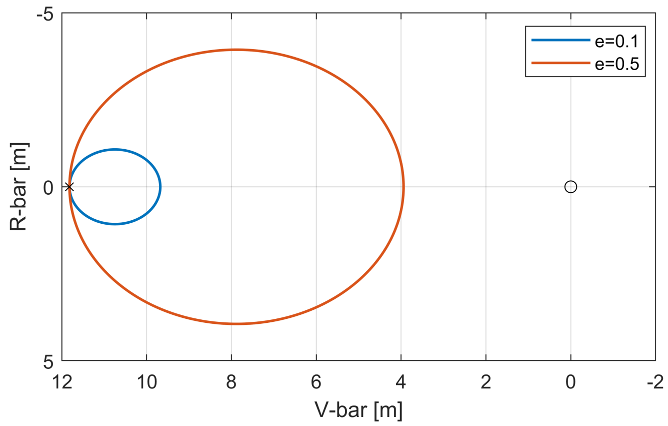


Figure 3.15: Relative in-plane motion in an elliptic target orbit with chaser on target orbit. Initial true anomaly $\theta_0 = 0\,\mathrm{deg}$ and initial orbital phase of $\Delta\theta = 0.0001\,\mathrm{deg}$.

With and R-bar perigee start with zero relative velocity, the result in figure 3.16 is obtained. The drift motion is similar to that of a circular target orbit in the same conditions (figure 3.4), but now the amplitude of the trajectory increases with each orbit. If the chaser starts at $\theta_0 = 90\,\mathrm{deg}$ instead, the motion is different, as shown in figure 3.17.

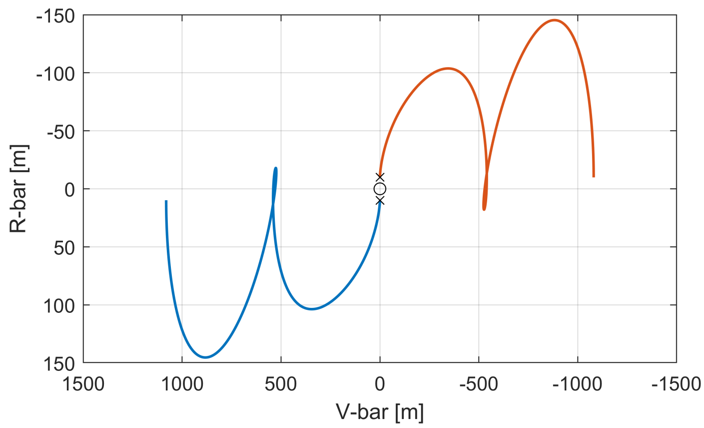


Figure 3.16: Relative in-plane motion in an elliptic target orbit with R-bar start. Initial conditions $s_0 = [0, 0, \pm 10]\,\mathrm{m}$, $\dot{s}_0 = [0, 0, 0]\,\mathrm{m/s}$, $\theta_0 = 0\,\mathrm{deg}$, eccentricity $e = 0.1$.

If the chaser starts on R-bar on a higher or lower orbit, but such that it has the same eccentricity as the target orbit, the result in figure 3.18 is obtained. Unlike in the case of a circular orbit with the same initial conditions (figure 3.5), where the chaser simply drifts along V-bar, now it also moves along R-bar. This is in part due to the motion of V-bar relative to the target velocity vector, but also because two orbits with the same eccentricity and a different perigee height will not have the same difference in apogee heights. Also, because the satellites have different orbital periods, the amplitude of the motion increases as they phase along the orbit. The initial conditions for this simulation were again defined in the orbital plane frame, placing the chaser on an orbit with the same eccentricity but with a different semi-major axis and on the same true anomaly, and then converting to the LVLH frame.

For the out-of-plane motion, the chaser trajectory is still approximately a sine wave, although the resemblance fades with the increase of the eccentricity, as can be observed in figure 3.19. It can also be

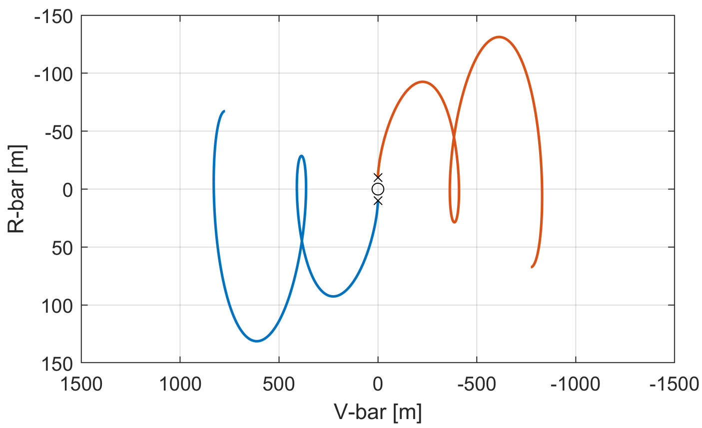


Figure 3.17: Relative in-plane motion in an elliptic target orbit with R-bar start and different initial true anomaly. Initial conditions $s_0 = [0, 0, \pm 10]\,\mathrm{m}$, $\dot{s}_0 = [0, 0, 0]\,\mathrm{m/s}$, $\theta_0 = 90\,\mathrm{deg}$, eccentricity $e = 0.1$.

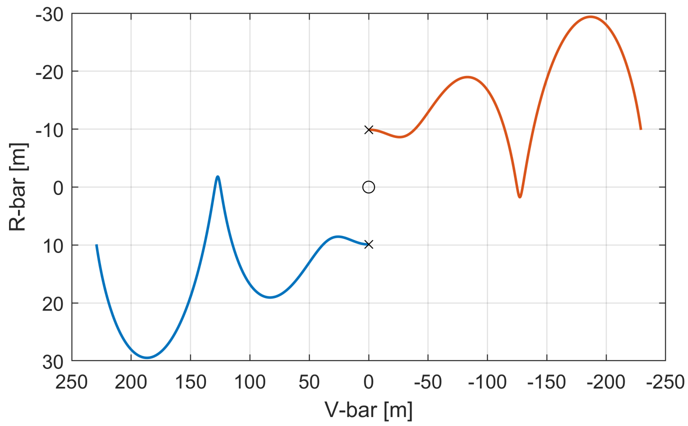


Figure 3.18: Relative in-plane motion in an elliptic target orbit with chaser on lower/higher with target eccentricity. Initial true anomaly $\theta_0 = 0\,\mathrm{deg}$, eccentricity $e = 0.1$ and difference in semi-major axis of $\Delta a = \pm 11\,\mathrm{m}$.

concluded from the results that the trajectory depends on the initial position of the target along its orbit, *i.e.* with the initial θ.

At perigee ($\theta = 0\,\mathrm{deg}$), the acceleration in H-bar is greater than at apogee ($\theta = 180\,\mathrm{deg}$), and so if the spacecraft start at perigee (blue trajectory) the chaser will drop further down along the normal direction than if they start at apogee (yellow trajectory). Furthermore, because of the difference in H-bar acceleration, the spacecraft will spend more time at apogee than at perigee, which causes the peaks of the blue trajectory to be more narrow than the valleys, while the opposite happens for the yellow trajectory. This effect is greater for a bigger eccentricity. If the chaser does not start at perigee or apogee (orange trajectory), the peaks and valleys do not match with apogee and perigee, and so the varying dynamics cause the wave to be asymmetrical.

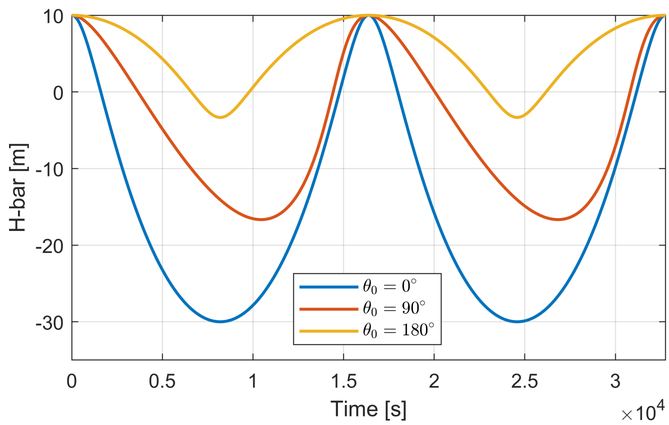


Figure 3.19: Relative out-of-plane motion in an elliptic target orbit with increased eccentricity. Initial conditions $s_0 = [0, 10, 0]\,\mathrm{m}$, $\dot{s}_0 = [0, 0, 0]\,\mathrm{m/s}$ and different initial true anomalies, eccentricity $e = 0.5$.

3.5.2 Impulsive Thrust Manoeuvres

The ΔV for an arbitrary impulsive manoeuvre given specific initial and final conditions can be determined with the Yamanaka-Ankersen transition matrix. Using the coordinate transformations we have for the in-plane motion

$$\begin{bmatrix} x_f \\ z_f \\ \dot{x}_f \\ \dot{z}_f \end{bmatrix} = \Lambda_i^{-1}(\theta_f)\Phi_i(\theta_f)\Lambda_i(\theta_0) \begin{bmatrix} x_0 \\ z_0 \\ \dot{x}_0 \\ \dot{z}_0 \end{bmatrix} , \tag{3.54}$$

where the index f represents the final conditions, and the transition time between the initial and final positions is specified with θ_f. Given an initial position, the required initial velocity to achieve the final position at the specified time can be obtained by solving the above equation for $\dot{x}_0$ and $\dot{z}_0$. Denoting the transformed transition matrix as D, the first two equations yield

$$\begin{bmatrix} x_f \\ z_f \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \end{bmatrix} \begin{bmatrix} x_0 \\ z_0 \\ \dot{x}_0 \\ \dot{z}_0 \end{bmatrix} , \tag{3.55}$$

and so the required ΔVs are

$$\begin{aligned} \dot{x}_0 &= \frac{d_{14} z_f + (d_{24} d_{11} - d_{21} d_{14}) x_0 + (d_{24} d_{12} - d_{22} d_{14}) z_0 - d_{24} x_f}{d_{23} d_{14} - d_{24} d_{13}}, \\ \dot{z}_0 &= \frac{d_{23} x_f + (d_{13} d_{21} - d_{11} d_{23}) x_0 + (d_{13} d_{22} - d_{12} d_{23}) z_0 - d_{13} z_f}{d_{23} d_{14} - d_{24} d_{13}}. \end{aligned} \tag{3.56}$$

Note that, due to the denominator in the above expressions, a transition time equal to an orbital period generates a singularity, and so is not possible. To eliminate the relative velocity at the end of the transfer, the final ΔV can be determined by computing the negative of $\dot{x}_f$ and $\dot{z}_f$ from equation (3.54).

Figure 3.20 shows an example of a V-bar transfer manoeuvre, similar to the one seen in figure 3.9

for a circular target orbit. It can be observed that the chaser achieves the final position in the specified transfer time. Figure 3.21 shows an in-plane transfer between arbitrary points.

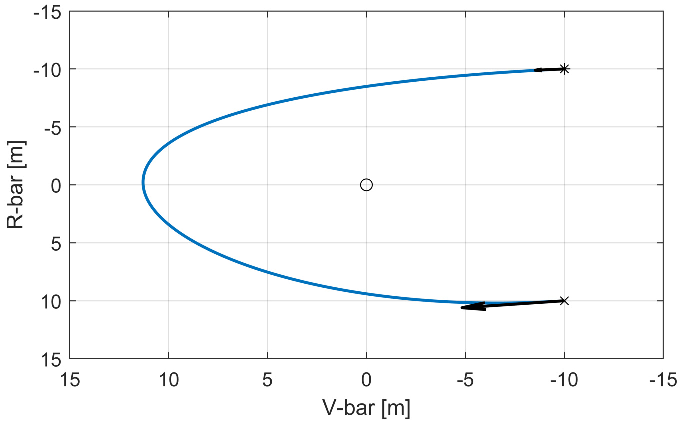


Figure 3.20: R-bar transfer manoeuvre in an elliptic target orbit. Initial position $s_0 = [-10, 0, 10]\,\mathrm{m}$ at $\theta_0 = 0\,\mathrm{deg}$, final position $s_f = [-10, 0, -10]\,\mathrm{m}$ at $\theta_f = 180\,\mathrm{deg}$, with eccentricity $e = 0.4$.

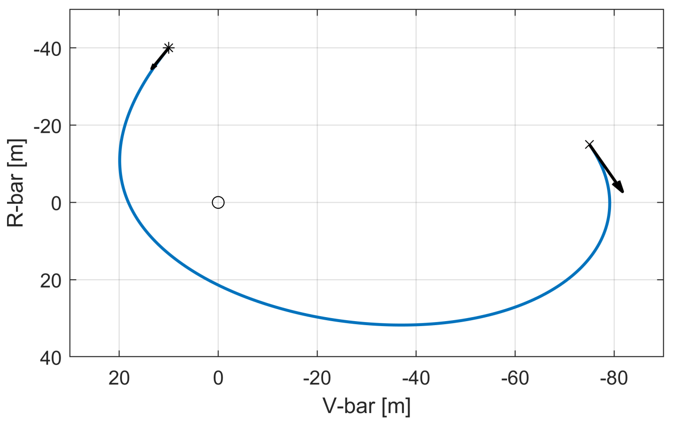


Figure 3.21: Arbitrary in-plane transfer manoeuvre in an elliptic target orbit. Initial position $s_0 = [-75, 0, -15]\,\mathrm{m}$ at $\theta_0 = 0\,\mathrm{deg}$, final position $s_f = [10, 0, -40]\,\mathrm{m}$ at $\theta_f = 180\,\mathrm{deg}$, with eccentricity $e = 0.4$.

For the out-of-plane dynamics, equation (3.54) becomes

$$\begin{bmatrix} y_f \\ \dot{y}_f \end{bmatrix} = \Lambda_o^{-1}(\theta_f)\Phi_o(\theta_f)\Lambda_o(\theta_0) \begin{bmatrix} y_0 \\ \dot{y}_0 \end{bmatrix} , \tag{3.57}$$

which yields the ΔV

$$\dot{y}_0 = \frac{k^2 \rho(\theta_0)}{\sin(\theta_f - \theta_0)} \left[\rho(\theta_f) y_f - (\cos(\theta_f - \theta_0) + e\cos(\theta_f)) y_0\right] . \tag{3.58}$$

This expression is singular for $\theta_f - \theta_0 = n\pi,\ n \in \mathbb{N}$, and so transfers of a half or full orbit are not possible.

Notice how for all these manoeuvres the first ΔV is greater than the last. This is due to the fact that these manoeuvres end at/near apogee, where the dynamics are slower than at perigee.

Chapter 4

Rendezvous With Model Predictive Control

In traditional rendezvous mission design, guidance trajectories are designed offline, and so manoeuvres are performed in open-loop, often times with punctual mid-course correction boosts determined online from the trajectory deviation [1]. In this context, an increasing amount of research has been dedicated to applying Model Predictive Control (MPC) to the rendezvous problem [6, 19, 29–35], in order to perform these thrust manoeuvres online and in full closed-loop. This is desirable since it increases the autonomy of the spacecraft and allows for more precise manoeuvres. Furthermore, MPC can handle crucial operational constraints present in a rendezvous mission, such as:

- minimization of propellant consumption,
- limited thruster authority,
- spacecraft collision safety and passive safety.

Most of the MPC literature for rendezvous is dedicated solely to the translational control of the spacecraft. In fact, in rendezvous processes the attitude and position control are typically separated, since there is a weak coupling between translational and rotational motion [1]. Docking and berthing operations require the two controllers to be coupled, though these manoeuvres are outside the scope of this work. Furthermore, the spacecraft attitude may be subject to more operational constraints, such as the exposure of the solar panels to the sun, or the orientation of antennae towards ground stations, which significantly complicate the MPC optimization problem. Thus, this work only deals with translational control. Nevertheless, MPC has been applied to spacecraft attitude control [36], and to coupled control [37–39].

MPC can perform both guidance and control functions. The manoeuvre terminal state can be input as the controller reference, generating a trajectory much like a guidance system. It also generates a sequence of control decisions, which can be discarded in favour of a different low-level controller. MPC can also perform solely controller functions if a reference trajectory is provided, instead of a reference terminal state, although that is not explored in this work. The navigation function is not considered in this work, and the state is always assumed to be known, although Section 4.6 considers the presence of measurement noise.

The computational requirement of MPC is its greatest limitation. In a rendezvous context, however, it can be computationally feasible to implement MPC online, due to the fact that orbital dynamics are fairly slow, and that relative dynamics can be accurately linearised, as shown in Chapter 3, which allows for the use of Linear MPC. The MPC computation time will be a major consideration in this chapter.

In Section 4.1 we present a method for sampling the dynamics for the prediction model, which deals with the fact that orbital dynamics are highly time-varying for highly elliptical orbits. Section 4.2 applies the standard MPC approach with the receding horizon strategy to the rendezvous problem, showing how it is not appropriate for use in this applications. In Section 4.3, the *finite-horizon* strategy is presented and shown to be more appropriate for rendezvous, as opposed to the receding-horizon strategy, since it allows for fuel-optimal manoeuvres. Section 4.4 presents the alternative *variable-horizon* MPC formulation, which allows for optimizing manoeuvre duration simultaneously with fuel. In Section 4.5 we approach the passive safety problem, presenting two new techniques for an efficient implementation of these constraints for real-time optimization. Section 4.6 considers the presence of disturbances, and presents a several robustness techniques, of which some are first proposed in this work. Finally Section 4.7 presents several simulations with the methods presented in this Chapter, some of them in the conditions of the PROBA-3 RVX.

4.1 Relative Dynamics Sampling

The prediction model to be used in the MPC formulation is the Yamanaka-Ankersen state transition matrix [16] together with the Ankersen ZOH particular solution [17], presented and derived in Chapter 3, which provides a linear model of the relative dynamics between two spacecraft in an elliptic orbit. A difficulty arises, however, related to the fact that orbital dynamics are time-varying in an elliptic orbit. For example, in the conditions of the Proba-3 mission, where the perigee height is 600 km and the orbit is highly elliptical with an eccentricity of 0.8111, the orbital period is approximately 19.6 hours. Despite this very long period, in just 100 seconds the spacecraft will change their true anomaly by 8.3° from perigee, while in the same amount of time they cover only 0.091° from apogee, as illustrated in figure 4.1. Thus, in these conditions the dynamics are about 100 times faster at perigee than at apogee, which then translates to the velocity of the relative motion.

Because in MPC there is a limited amount of samples, associated with the length of the prediction horizon, these must be allocated appropriately along the orbit in order to get the best performance, since the point at which the thrust is applied is important for the optimality of the trajectory. If the dynamics are sampled with constant time intervals, as is often the norm, with 100 samples and in the conditions of the PROBA-3 mission we get the result in figure 4.2. The samples concentrate on apogee because the orbital velocity is lower there, which is opposite to what is desired, since the dynamics are faster on perigee. This results in the true anomaly interval between samples being over 50° at perigee, and less

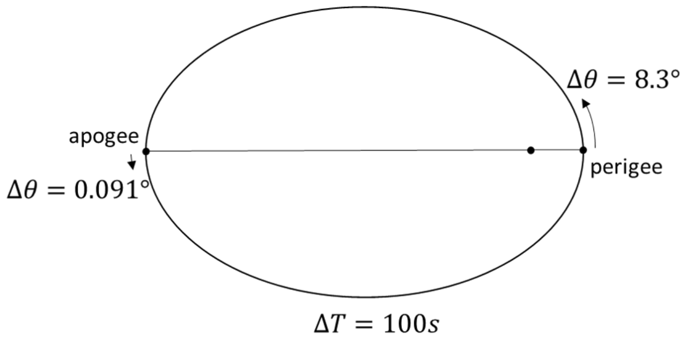


Figure 4.1: Illustration of the time-variation of the relative dynamics, in the conditions of the PROBA-3 mission.

than 1^{o} at apogee.

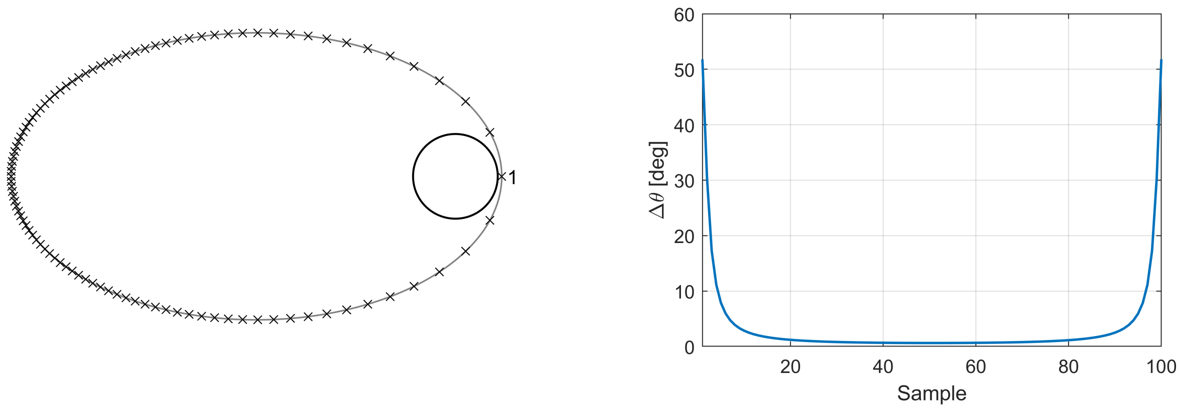


Figure 4.2: Time sampling of relative dynamics with 100 samples in conditions of PROBA-3 mission.

An alternative is to sample the dynamics with a constant true anomaly. This has the opposite effect, as seen in figure 4.3, where now the samples are concentrated on perigee, which is more desirable. One further advantage is that Kepler's equation no longer needs to be solved numerically in order to determine the true anomaly and compute the transition matrix. However, the samples at apogee may now be too spread apart, and the time interval is almost 4000 seconds at apogee, and 40 seconds at perigee.

A final possibility is to sample the system with constant eccentric anomaly intervals. As seen in figure 4.4, this results in the samples being evenly spread in space. Furthermore, the time intervals are still greater at apogee than at perigee, but the difference is not as considerable as before. Thus, this approach might be the most appropriate, and will be used throughout this chapter. Note that sampling the system without time-constant intervals has an effect on the actuation profile, since a ZOH discretization is utilized, which results in a constant thrust along each sampling interval and so each thrust action will have a different duration. In the case of a circular target orbit, the dynamics are time-invariant, and so time sampling will be used. Since manoeuvres can last up to several hours, sampling periods are often very large, and thus the ZOH discretization results in very long constant burns being commanded, which may be undesirable. However, the burn time in the particular solution in (3.40) can be set to less than the sampling period, yielding a partial zero-order-hold discretization.

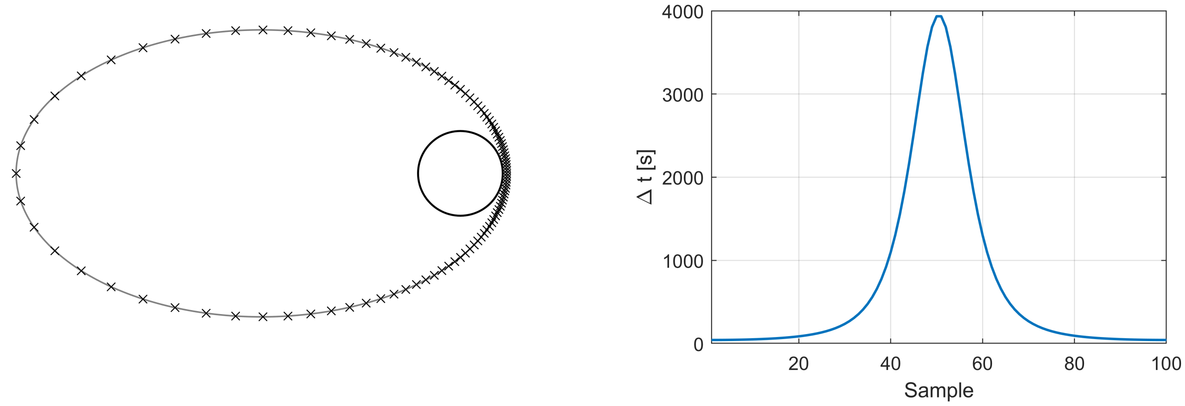


Figure 4.3: True anomaly sampling of relative dynamics with 100 samples in conditions of PROBA-3 mission.

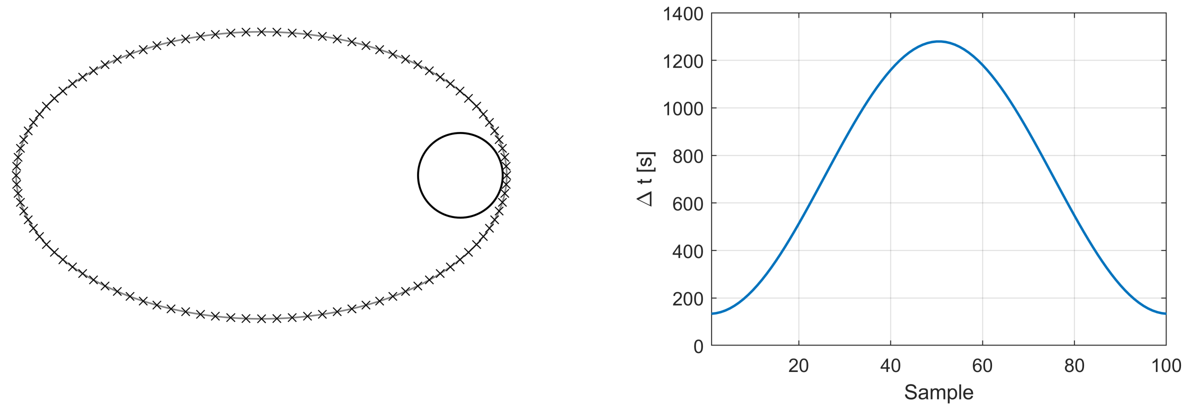


Figure 4.4: Eccentric anomaly sampling of relative dynamics with 100 samples in conditions of PROBA-3 mission.

4.2 Rendezvous with Receding-Horizon MPC

The most common and naive approach for the MPC formulation is to use a quadratic cost function, as presented in equation (2.13), together with the standard receding horizon strategy. In the absence of state and control constraints it becomes

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad (\bar{x}_N - x_{ref})^\top Q_f(\bar{x}_N - x_{ref}) + \sum_{i=0}^{N-1} (\bar{x}_i - x_{ref})^\top Q(\bar{x}_i - x_{ref}) + \bar{u}_i^\top R\bar{u}_i \tag{4.1a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{4.1b}$$

$$\bar{x}_{k+1} = A_k^{k+1}\bar{x}_k + B_k^{k+1}\bar{u}_k,\ k = 0,\ldots,N-1 \tag{4.1c}$$

where the prediction model is that presented in (3.47).

This section will show through simulation that the receding horizon strategy is not appropriate for the rendezvous problem, as well as show the progression toward an optimal formulation. To simplify, a circular target orbit will be considered, with the height of the perigee of the PROBA-3 mission (600 km), and the chaser mass considered is that of the PROBA-3 Occulter Spacecraft, with 211 kg (launch mass).

A simple V-bar transfer manoeuvre of 30 metres is considered, in order to easily evaluate the controller performance by comparing it with the ideal impulsive manoeuvres of this type, shown in figures 3.10 and 3.11. The controller parameters used in the following simulations, as well as the ΔV spent performing the manoeuvre and the average computation time, are presented in table 4.1. In the absence of any state or control constraints, the QP can easily be solved analytically.

Figure	T_s	N	R	Q	Q_f	ΔV	t_{avg}
4.5	$1\,\mathrm{s}$	10	I	$100I$	$100I$	$6.77\,\mathrm{m/s}$	$40\,\mu\mathrm{s}$
4.6	$1\,\mathrm{s}$	10	I	I	I	$1.41\,\mathrm{m/s}$	$39\,\mu\mathrm{s}$
4.7	$100\,\mathrm{s}$	10	$10^6 I$	I	I	$111\,\mathrm{mm/s}$	$50\,\mu\mathrm{s}$
4.8	$290\,\mathrm{s}$	20	I	0	$100I$	$5.27\,\mathrm{mm/s}$	$70\,\mu\mathrm{s}$

Table 4.1: Controller parameters and results for V-bar transfer manoeuvre simulations with receding-horizon quadratic MPC.

Also, because the in-plane and out-of-plane motions are decoupled, the two can be solved in separate MPC problems, where each problem becomes smaller and easier to solve. Although solving separately was found to improve the computation time when solving analytically, it often became worse when solving numerically. Furthermore, this separation is not possible in the presence of constraints that relate both motions, such as collision avoidance and passive safety constraints.

Short Horizon

In a first simulation, a short sampling period of 1 s and a prediction horizon of 10 samples are used, granting only 10 seconds of prediction, which is a much shorter time frame than that of the orbital motion, given that the orbital period is approximately 96 minutes. Furthermore, a high state cost is used, which results in the whole manoeuvre being performed in a straight line in approximately 60 seconds, as shown in figure 4.5. The generated manoeuvre requires a ΔV of 6.77 m/s and a thrust that would greatly exceed the capabilities of the small spacecraft. In comparison, the ideal V-bar transfer manoeuvre with two horizontal impulses requires a total ΔV of 3.45 mm/s, as determined with equation (3.52), which is almost 2000 times less spent fuel. On the other hand, the ideal manoeuvre requires one orbital period to complete, but because the spacecraft fuel is extremely limited, and in order to take advantage of the relative dynamics, this is the standard time frame for rendezvous operations.

To decrease the manoeuvre ΔV, the state cost is reduced, which makes the controller slower and decreases the fuel spent. However, as observed in figure 4.6, the system is now more affected by the natural drift between the satellites and is not able to converge on the reference. This happens because the controller only predicts 10 seconds ahead, and so is not able to predict much of the natural drift, and it does not have the authority to react to it.

Long Horizon

To counteract this, the prediction window is increased to the order of magnitude of the orbital period. However, since increasing the prediction horizon significantly worsens the computation time, the sampling period is instead increased to 100 s, granting the controller a prediction of 1000 seconds. The

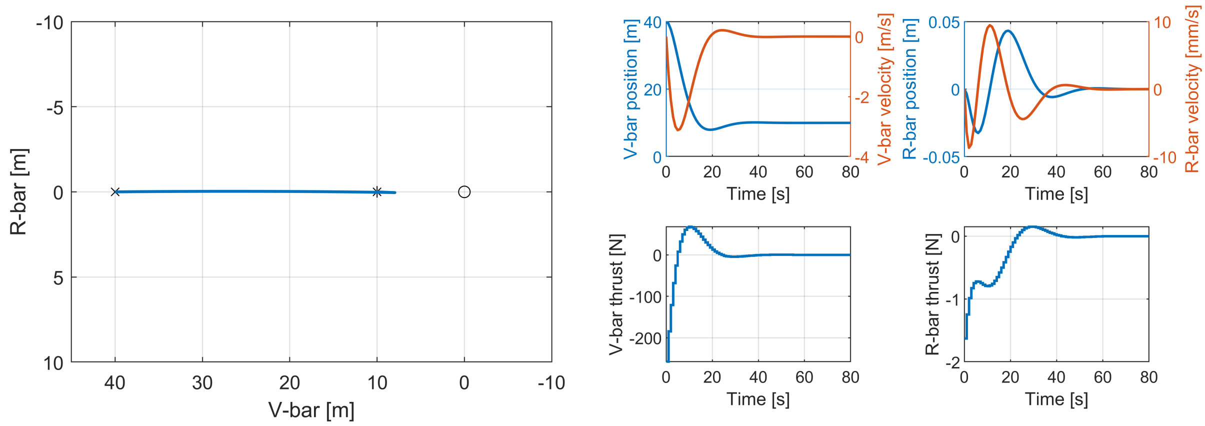


Figure 4.5: V-bar transfer manoeuvre with receding-horizon quadratic MPC, short prediction window and high state cost.

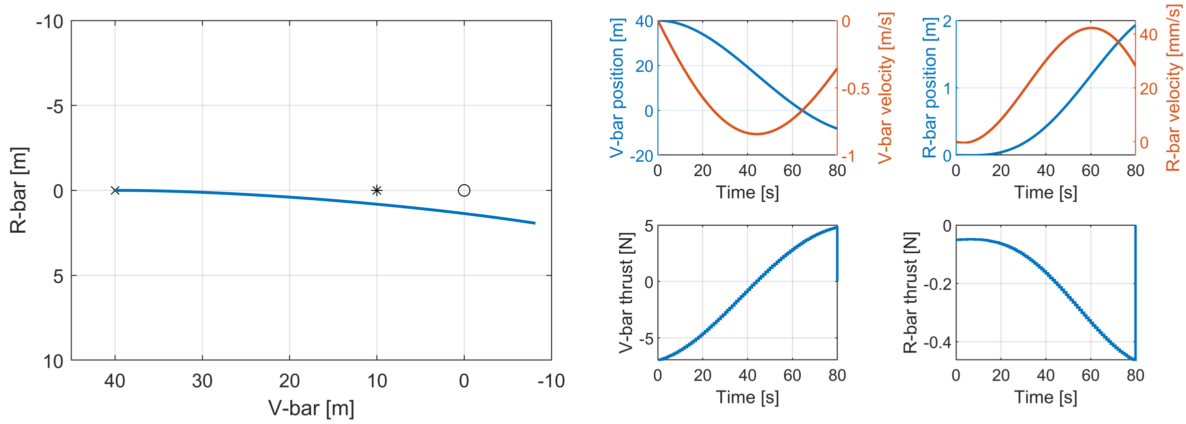


Figure 4.6: V-bar transfer manoeuvre with receding-horizon quadratic MPC, short prediction window and lower state cost.

input cost is also increased, to compensate for the fact that thrust intervals are now longer and so have a greater effect on the system. As observed in figure 4.7, the chaser actuation now has lesser amplitude, but because its prediction is improved it can use the natural relative motion to better converge on the reference. This is also evidenced by the fact that now there is extensive R-bar actuation, despite the fact that the reference is only offset in V-bar from the initial position. The manoeuvre ΔV is now 111 mm/s, which is a significant improvement, but still far from the ideal figure.

Intermediate State Cost

To improve the performance, the intermediate state cost can be disregarded, by setting $Q = 0$, which allows the controller to better plan ahead, since now it is not penalized for not being on the reference state while halfway through the manoeuvre. Furthermore, in an attempt to reproduce the ideal V-bar transfer manoeuvre with horizontal impulses, the sampling time is chosen such that the prediction window is exactly one orbital period. The prediction horizon N is also increased in order to avoid the sampling time becoming too great. As shown in figure 4.8, the trajectory resembles that of the ideal manoeuvre, although the spacecraft cannot reach the reference in one orbital period, and in the final approach it

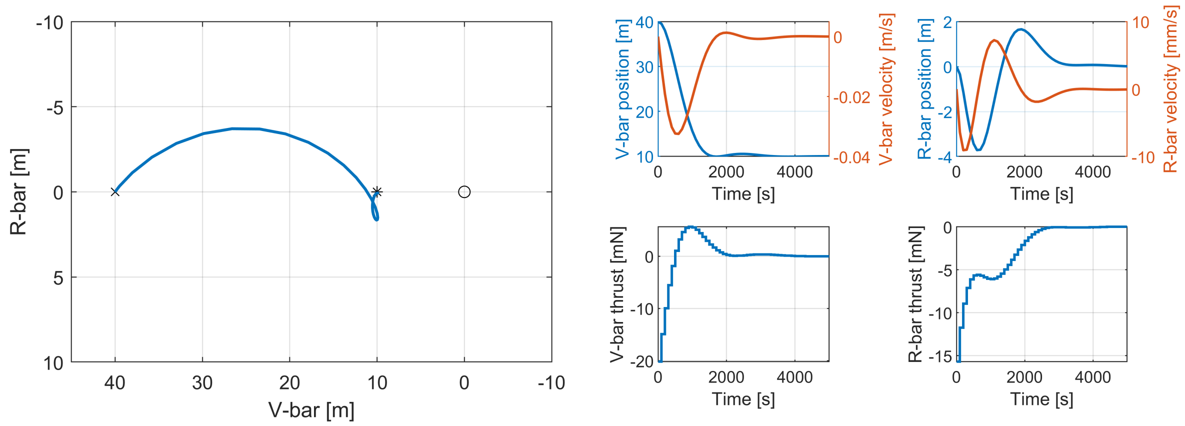


Figure 4.7: V-bar transfer manoeuvre with receding-horizon quadratic MPC and long prediction horizon.

circles the reference, getting closer and closer while never reaching it. This is due to the receding horizon strategy; since the prediction horizon slides forward every sample, the state that is being tracked is always one orbital period away, and so the controller never makes the final effort to reach the reference. Therefore, a different strategy is required.

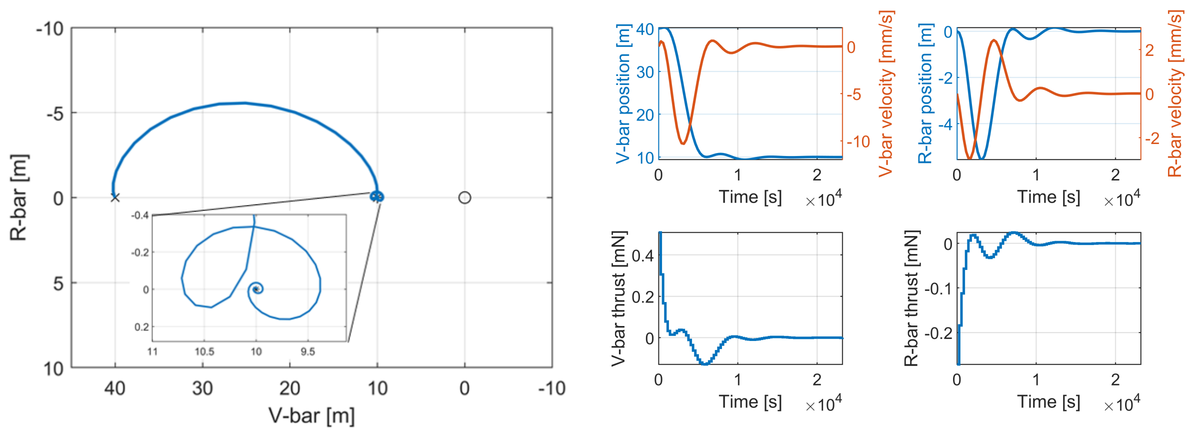


Figure 4.8: V-bar transfer manoeuvre with receding-horizon quadratic MPC, long prediction horizon and no intermediate state cost.

4.3 Fixed-Horizon MPC

An alternative to the receding-horizon strategy is to decrement the prediction horizon every sample, such that its edge is always at the same time-instant, and thus known as Fixed-Horizon MPC (FH-MPC) [19]. This, together with the use of a terminal state cost and no intermediate state cost, allows for the manoeuvre to be completed in a specified amount of time, which in turn allows for the controller to generate the ideal optimal manoeuvres. The parameters for the following experiments are presented in table 4.2. Because the prediction horizon is decremented and thus the computational complexity decreases every sample, the worst case is now shown instead of the average.

With the same conditions as in figure 4.8 and the FH strategy yields the result in figure 4.9. The tra-

Figure	T_s	N	R	Q	Q_f	u_{max}	ΔV	t_{max}
4.9	$290\,\mathrm{s}$	20	I	0	$100I$	-	$7.61\,\mathrm{mm/s}$	$121\,\mu\mathrm{s}$
4.10	$290\,\mathrm{s}$	21	-	-	-	-	$3.45\,\mathrm{mm/s}$	$7.97\,\mathrm{ms}$
4.11	$58.0\,\mathrm{s}$	100	-	-	-	$1\,\mathrm{mN}$	$4.68\,\mathrm{mm/s}$	$9.40\,\mathrm{ms}$

Table 4.2: Controller parameters and results for V-bar transfer manoeuvre simulations with FH-MPC.

jectory closely resembles that of the ideal manoeuvre and the system reaches the reference in exactly one orbital period, although the ΔV applied is still over two times greater. This is due to the quadratic cost used for the input variable, meaning that the cost function is not directly proportional to the manoeuvre ΔV, since this parameter is linearly proportional to the absolute value of the input force. This also results in a continuous control action instead of sparse like the ideal manoeuvre, with just two thruster burns.

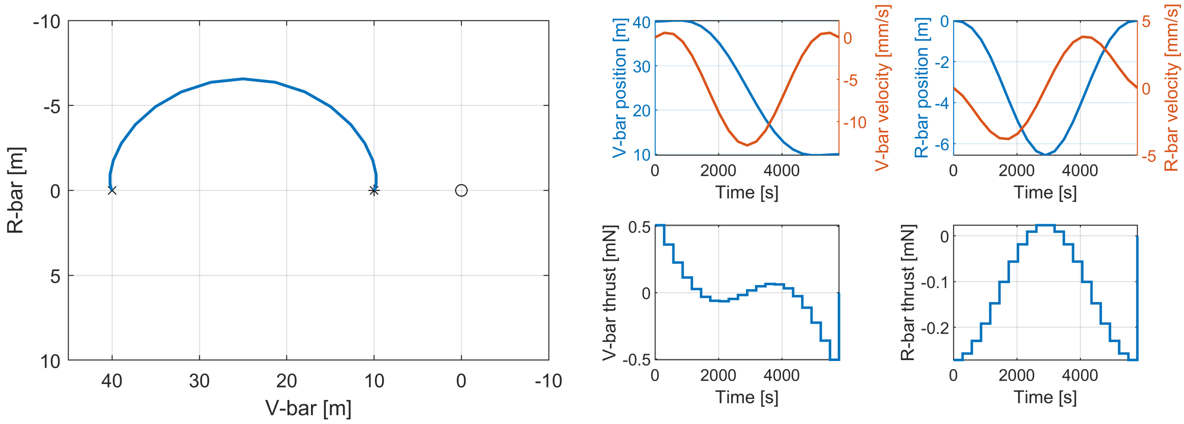


Figure 4.9: V-bar transfer manoeuvre with FH-MPC and quadratic input cost.

Fuel-Optimal LP Formulation

To obtain a fuel-optimal trajectory, the ℓ_1-norm must be used for the input cost instead of the quadratic function, while the terminal state cost remains quadratic. This resembles the Lasso cost function discussed in section 2.2.4, with $R = 0$ and $R_\lambda = I$. However, the inclusion of the ℓ_1-norm term significantly complicates the optimization problem and makes it more difficult to solve, since it is no longer a QP.

One possible way to simplify the optimization problem is to modify the prediction model such that the input forces are split into its positive and negative components

$$F = F^+ - F^- \tag{4.2}$$

by extending the B matrix, and thus extending the input vector to $u = \left[F_x^+, F_x^-, F_y^+, F_y^-, F_z^+, F_z^-\right]^\top$. This increases the number of optimization variables, which is disadvantageous, but now each input variable can only take positive values, which makes its absolute value equal to itself. Thus, the ℓ_1-norm can be discarded and the formulation becomes

$$\begin{aligned}
\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad & (\bar{x}_N - x_{ref})^\top Q_f (\bar{x}_N - x_{ref}) + \sum_{i=0}^{N-1} \Delta t_i \mathbf{1}^\top \bar{u}_i && \text{(4.3a)}\\
\text{s.t.} \quad & \bar{x}_0 = x_t, && \text{(4.3b)}\\
& \bar{x}_{k+1} = A_k^{k+1} \bar{x}_k + B_k^{k+1} \bar{u}_k, \quad k = 0, \ldots, N, && \text{(4.3c)}\\
& \bar{u}_k \geq 0, \quad k = 0, \ldots, N-1, && \text{(4.3d)}
\end{aligned}$$

which is again a QP, where $\mathbf{1}$ is a column vector of 1's. Note that the input variables are weighted by the time interval between samples Δt, which is crucial in obtaining a fuel-optimal formulation in case the dynamics are not sampled with constant time intervals, since the ΔV is proportional to the duration the thrusters are fired. Furthermore, the constraint (4.3d) ensures that this formulation is equivalent to that with the ℓ_1-norm, since it constrains each of the input force components to be equal to or greater than zero.

The formulation can be further simplified by using a terminal state constraint, instead of a terminal state cost, and thus the problem becomes

$$\begin{aligned}
\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N-1}\\ \bar{x}_0,\ldots,\bar{x}_N}} \quad & \sum_{i=0}^{N-1} \Delta t_i \mathbf{1}^\top \bar{u}_i && \text{(4.4a)}\\
\text{s.t.} \quad & \bar{x}_0 = x_t, && \text{(4.4b)}\\
& \bar{x}_{k+1} = A_k^{k+1} \bar{x}_k + B_k^{k+1} \bar{u}_k, \quad k = 0, \ldots, N, && \text{(4.4c)}\\
& \bar{u}_k \geq 0, \quad k = 0, \ldots, N-1, && \text{(4.4d)}\\
& \bar{x}_N = x_{ref}, && \text{(4.4e)}
\end{aligned}$$

which is now an LP and can be solved very efficiently. Another advantage of this formulation is that, once the manoeuvre duration is defined, there are no controller parameters that need to be tuned, and, in the absence of disturbances, the controller will always reach the reference state without static error. Thus, since the cost function only contains one term that is linearly proportional to the ΔV and since the optimization problem is convex, this formulation is guaranteed to always generate the fuel-optimal trajectory in the specified transfer duration. The disadvantage of using a hard terminal constraint is that the optimization problem may become infeasible, given the constraints and the length of prediction horizon, especially since in the FH-MPC strategy the prediction horizon is decremented until it is only one sample. However, in the absence of disturbances the problem will never become infeasible if previous iterations are feasible, and so this issue will only be addressed in Section 4.6.

Figure 4.10 shows the result of the fuel-optimal linear FH-MPC formulation applied to the one-orbit V-bar transfer. It can be observed that the manoeuvre is performed solely with two thruster actions in the horizontal direction, much like the ideal impulsive manoeuvre in figure 3.11. Furthermore, the total ΔV applied is approximately the same as in the ideal manoeuvre, validating the fact that this formulation is fuel-optimal. Note that, to obtain this exact ΔV value, the manoeuvre duration has to be one sample more than the orbital period, to account for the time it takes to perform the last braking input action.

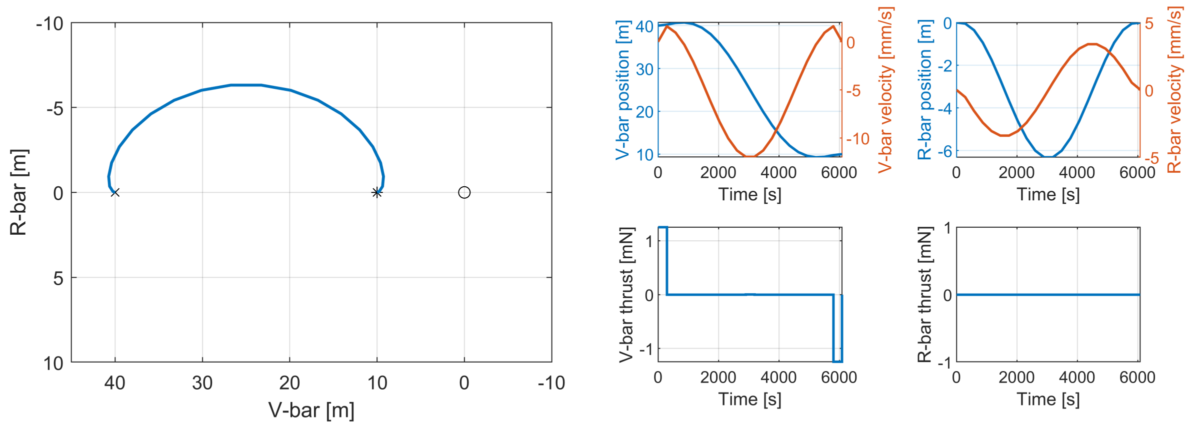


Figure 4.10: V-bar transfer manoeuvre with fuel-optimal linear FH-MPC.

Control Saturation

Limiting the maximum thrust that can be applied by each of the spacecraft thrusters, constraint (4.4d) becomes

$$u_{max} \geq \bar{u}_k \geq 0, \tag{4.5}$$

where $u_{max} \in \mathbb{R}^6$ is the maximum thrust for each of the input components. Notice that with this constraint the maximum thrust is independent for each direction. If the spacecraft does not have omnidirectional thrust, however, it is more appropriate to constrain the magnitude of the total thrust vector, although this cannot be performed with linear constraints. In [40], this more realistic constraint is utilized, formulating the optimization problem as a second-order cone program. However, if the simpler constraint is used, as it will be in this work in order to maintain the optimization problem as an LP, u_{max} must be lesser than the physical maximum thrust possible, which is a suboptimal approach since the full capacity of the thrusters is not utilized [29]. In the case of the PROBA-3 Occulter Spacecraft, there is omnidirectional thrust and thus the constraint (4.5) realistically models the spacecraft thruster limitations.

In figure 4.11, the prediction horizon is increased and control limits of 1 mN for each component are added, and thus it can be observed that the initial and final control actions now happen over more than one sample. Also, the ΔV has slightly increased from the previous experiment, which is due to the fact that the thrusters are now limited. Finally, from table 4.2 it can be seen that, despite having increased the prediction horizon by a factor of five and adding more constraints, the computation time only increased by approximately 18%, which is due to the fact that the optimization problem is an LP and can be solved very efficiently. Hence, we showed that this formulation can generate fuel-optimal trajectories, similar to the ones already used to plan rendezvous missions, and that it can generate them very efficiently and feasibly in real-time.

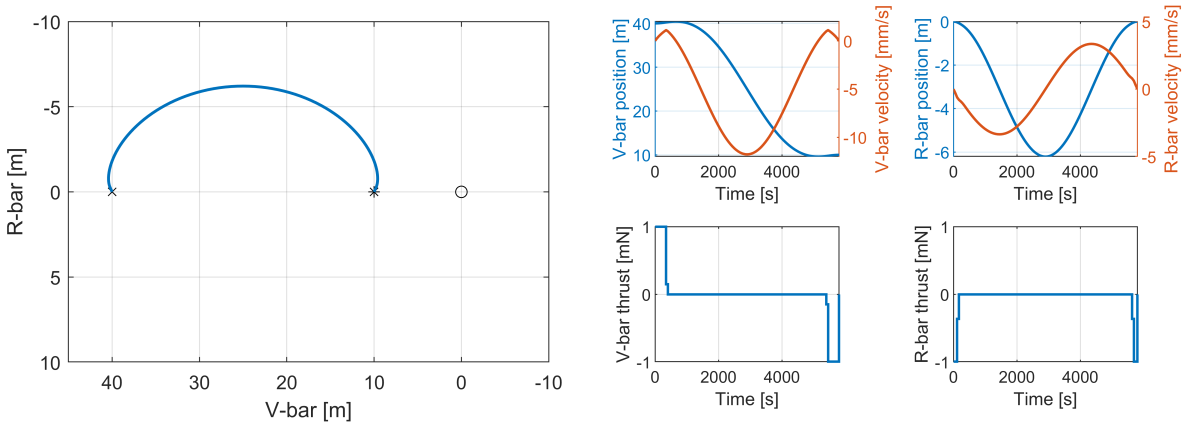


Figure 4.11: V-bar transfer manoeuvre with fuel-optimal linear FH-MPC, control limits and increased prediction horizon.

4.4 Variable-Horizon MPC

The previous formulation optimizes the fuel required for the manoeuvre given a pre-specified transfer duration. However, it is also desirable to optimize the manoeuvre duration as well as fuel expenditure. This requires adding the prediction horizon N as an integer optimization variable, and so the formulation known as Variable-Horizon MPC (VH-MPC) becomes

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N_{max}-1}\\ \bar{x}_0,\ldots,\bar{x}_{N_{max}}\\ N\in\mathbb{N}}} \quad \gamma N + \sum_{i=0}^{N_{max}-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \tag{4.6a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{4.6b}$$

$$\bar{x}_{k+1} = A_k^{k+1}\bar{x}_k + B_k^{k+1}\bar{u}_k, \tag{4.6c}$$

$$0 \le \bar{u}_k \le u_{max},\ k = 0,\ldots,N_{max}-1, \tag{4.6d}$$

$$\bar{x}_N = x_{ref}, \tag{4.6e}$$

$$1 \le N \le N_{max}, \tag{4.6f}$$

where N_{max} is the bound for the prediction horizon. The prediction horizon is also added to the cost function, and thus the parameter γ is used to tune the trade-off between transfer time and fuel consumption. If $\gamma = 0$, the solution is the manoeuvre duration that minimizes the fuel within the bounds of the prediction horizon; for simple manoeuvres in a circular target orbit, the solution will typically be at N_{max}, while for more complex manoeuvres and in elliptic target orbits that may not be the case. Constraint (4.6e) now becomes nonlinear, since it indexes an optimization variable with another, and thus the problem is a mixed-integer nonlinear program, which is computationally expensive to solve.

A method of transforming this problem into a mixed-integer linear program (MILP) was first presented by Richards and How [41] and applied to rendezvous in [19], requiring the substitution of the prediction horizon by two vectors of binary optimization variables. Variable v_k is 1 if the manoeuvre is completed exactly at instant k, and 0 otherwise, and thus

$$\sum_{k=1}^{N_{max}} v_k = 1. \tag{4.7}$$

The variable p_k is 1 while the manoeuvre is not completed, and 0 afterwards, and so conceptually we have

$$\sum_{k=1}^{N_{max}} p_k = N. \tag{4.8}$$

The two binary variables are related by the dynamic equation

$$p_{k+1} = p_k - v_{k+1}, \tag{4.9}$$

which maintains their integrity: if the manoeuvre is completed at $k+1$, then $v_{k+1} = 1$ which forces p_{k+1} to flip values, otherwise it is maintained. The VH-MPC MILP formulation then becomes

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N_{max}-1} \\ \bar{x}_0,\ldots,\bar{x}_{N_{max}} \\ p_0,\ldots,p_{N_{max}} \in \{0,1\} \\ v_1,\ldots,v_{N_{max}} \in \{0,1\}}} \quad \gamma \sum_{i=0}^{N_{max}} p_i + \sum_{i=0}^{N_{max}-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \tag{4.10a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{4.10b}$$

$$\bar{x}_{k+1} = A_k^{k+1} \bar{x}_k + B_k^{k+1} \bar{u}_k, \tag{4.10c}$$

$$0 \leq \bar{u}_k \leq u_{max},\ k = 0, \ldots, N_{max} - 1, \tag{4.10d}$$

$$-(1 - v_k)h \leq x_k - x_{ref} \leq (1 - v_k)h, \tag{4.10e}$$

$$p_{k+1} = p_k - v_{k+1},\ k = 0, \ldots, N_{max} - 1, \tag{4.10f}$$

$$p_{N_{max}} = 0, \tag{4.10g}$$

$$\sum_{k=1}^{N_{max}} v_k = 1. \tag{4.10h}$$

Notice that the optimal prediction horizon is no longer implicitly included in the cost function, but rather the sum of the p variables, since we have the relation in (4.8). Also, the properties in equations (4.7) and (4.9) are included as optimization constraints in (4.10h) and (4.10f), respectively, while the constraint (4.10g) forces the manoeuvre to be completed at least by the end of the maximum prediction horizon. Lastly, the terminal state constraint in (4.10e) is now an inequality constraint, where parameter h is a sufficiently large number. Thus, the term $(1 - v_k)$ allows to trigger the terminal constraint: at the moment the manoeuvre is completed, v_k is 1 and this term is 0, such that the bounds of the linear inequality are tight and the terminal state constraint becomes active; otherwise it is 0, and so the bounds are very wide and thus the constraint becomes inactive.

The computational load for this formulation is greater than for the FH-MPC, since MILP problems are harder to solve. Thus, in a real-time scenario, it may be preferable to predetermine offline the manoeuvre duration and use the FH-MPC formulation instead. However, as proposed in [35], determining the manoeuvre duration offline can be performed in an optimal way by using the VH-MPC formulation. The optimal transfer time may change slightly along the way due to disturbances, but the time determined offline can still be expected to remain approximately optimal. As will be discussed in Section 4.6, however, using the VH-MPC formulation online is advantageous for its feasible robustness property. When used online, if the maximum manoeuvre duration counting from the initial instant is to be maintained for subsequent iterations then N_{max} should be decremented every instant, otherwise the controller might deviate from its initial trajectory and extend the manoeuvre beyond the initial maximum final instant in order to spend less fuel.

In [35] the authors present an extension to the VH-MPC framework that allows for multi-step manoeuvres to be considered, where the durations of the sub-manoeuvres are optimized simultaneously and offline, also via integer linear programming. The resulting multi-step manoeuvre is then performed online as a sequence of several FH-MPC manoeuvres.

4.5 Passive Safety

It is a requirement in rendezvous missions that, besides ensuring that the nominal trajectory does not cause a collision between the spacecraft, the free-drift motion from any point in the trajectory also remain collision-free within a specified time-horizon. Designing trajectories in this way ensures that, in case a thruster fails to fire, the two spacecraft will not collide due to the natural drift, and thus this is known as passive safety. Furthermore, in the event of any other fault that warrants an abort in the approach, it becomes a safe strategy to simply shut-down the spacecraft thrusters. Figure 4.12 illustrates the need for designing passively safe rendezvous trajectories, where a V-bar transfer manoeuvre is shown such that a collision occurs after one orbit in case the final thrust fails.

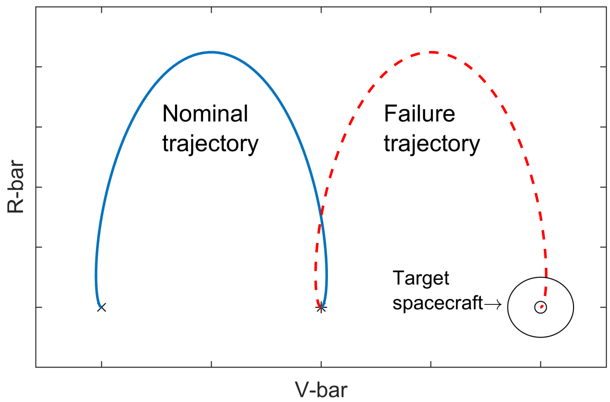


Figure 4.12: Illustration of the passive safety problem in a V-bar transfer manoeuvre.

Typically, passively safe trajectory design is performed by choosing specific classes of manoeuvres with good passive safety properties. For example, in the example in figure 4.12, if radial pulses are applied instead of horizontal, the nominal trajectory will be similar to the one in figure 3.10, and the free-drift failure trajectory would be similar to figure 3.6, which returns to the initial position after half an orbit, thus avoiding collision in an infinite horizon. Ensuring passively safe trajectories usually comes at the cost of increased fuel expenditure; in the previous example, the radial-pulse manoeuvre requires a ΔV over four times greater.

For rendezvous with MPC, passive safety design has to be included in the optimization problem as a constraint. Typically, this is performed by constraining the discrete states in the nominal and failure trajectories to be outside of the target spacecraft or its safety region. The following sections present different methods of formulating these obstacle avoidance constraints, which will then be extended to passive obstacle avoidance constraints in Section 4.5.4.

4.5.1 Obstacle Avoidance with Nonlinear Optimization

The most straightforward approach to formulating an obstacle avoidance constraint is to simply constrain the states to be outside the region defined by the obstacle. Since this constrains a connected region of the state-space, the feasible set becomes non-convex. Thus, these obstacle avoidance constraints result in a optimization problem with nonlinear constraints, which makes it difficult to solve and introduces different local minima.

For example, for the circular obstacle illustrated in figure 4.13, the obstacle avoidance constraint becomes

$$\|x_k - c\|^2 \geq r^2,\ k = 1, \ldots, N, \tag{4.11}$$

requiring N nonlinear optimization constraints. With the passive safety constraints as well, the number of nonlinear optimization constraints increases dramatically, greatly lowering computational performance and making it infeasible to use this approach in real-time.

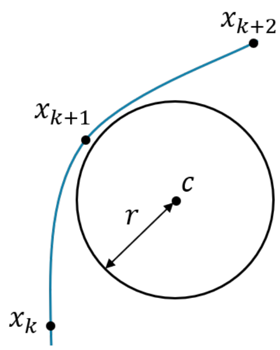


Figure 4.13: Illustration of obstacle avoidance of circular object with nonlinear constraints.

4.5.2 Obstacle Avoidance with Mixed-Integer Linear Optimization

An alternative method of formulating an obstacle avoidance constraint is with the use of linear constraints and auxiliary binary optimization variables, thus turning the problem into a MILP [42]. The binary variables are used to activate and deactivate the different constraints that define the obstacle, allowing these constraints to be linear inequalities. In the example in figure 4.14, the state at time k is only subject to constraints x_{min} and y_{min}, while the other constraints are not active, while for the state at time $k+2$, only the constraint y_{max} is active.

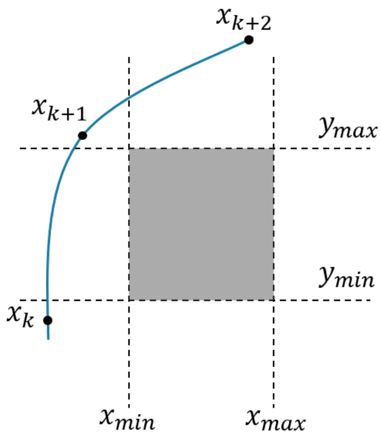


Figure 4.14: Illustration of obstacle avoidance with mixed-integer linear optimization.

The obstacle avoidance constraint for the state at time k then becomes

$$x_k \leq x_{min} + Ma_{k,1} \tag{4.12a}$$

$$-x_k \leq -x_{max} + Ma_{k,2} \tag{4.12b}$$

$$y_k \leq y_{min} + Ma_{k,3} \tag{4.12c}$$

$$-y_k \leq -y_{max} + Ma_{k,4} \tag{4.12d}$$

$$\sum_{i=1}^{4} a_{k,i} \leq 3, \tag{4.12e}$$

where $a_{k,i}$ are binary optimization variables, and M is a large positive number. If $a_{k,i} = 0$, the linear inequality is tight and thus constraint i is active for the state at time k. Otherwise, the linear inequality is relaxed and the constraint inactive. Constraint (4.12e) is crucial, as it forces at least one constraint to be active. With this method, only polytope obstacle shapes can be modelled, since it relies on linear inequalities.

In a three-dimensional space and with a cubic obstacle, this method requires six optimization variables and seven optimization constraints per time instant, which again can make it infeasible to operate in real-time with the passive-safety constraint, since integer programming is NP-complete. An advantage of this method, however, is that the global minimum can more easily be found, since there is a finite number of active constraint configurations.

4.5.3 Obstacle Avoidance with Linear Optimization

Another method to perform obstacle avoidance is with pure linear optimization. In [43] the obstacle constraint is replaced with a convex set that excludes the obstacle, but is static along the the prediction. This approach does not allow for trajectories that bend around the obstacle, and otherwise results in a significant ΔV increase, since the trajectory must lie within a much more conservative region.

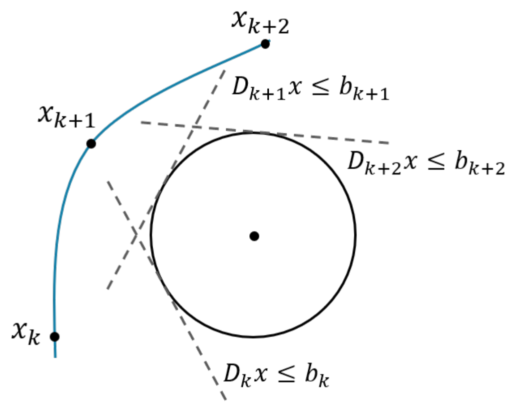


Figure 4.15: Illustration of obstacle avoidance with linear constraints.

Another approach is to subject each state in the trajectory to a different linear inequality constraint

$$D_k x_k \leq b_k, \; k = 1, \dots, N, \tag{4.13}$$

which is tangent to the original obstacle, as illustrated in figure 4.15. The linear constraints have to be determined *a priori* to the optimization, which can be performed offline. Furthermore, the state at each time is now subject to a more conservative constraint, which can affect the optimality of the trajectory, although the optimization problem becomes an LP again, allowing it to be solved much more efficiently.

The method used to determine the linear constraints $D_k x \leq b_k$ currently found in the literature is to rotate the constraint around the obstacle with time [29, 30, 33, 44], where the rate at which it rotates is a controller parameter that must be optimized offline. This method is not appropriate to use in the passive safety problem, as each failure trajectory would require different rates of rotation that must be optimized simultaneously.

Offline Nonlinear Optimization

In this work we propose a different method for determining the linear inequality constraints which is simple, does not require any parameter tuning, and relies on offline optimization. First, the problem is solved once and offline with the original obstacle constraints, formulated with either of the methods from Sections 4.5.1 and 4.5.2. Figure 4.16a) illustrates this step, with a circular obstacle and for a V-bar transfer manoeuvre. Then, as demonstrated in figure 4.16b), the planes tangent to the obstacle and facing each of the states are determined. These tangent planes are then used to define the linear obstacle constraints of the online LP.

In the absence of disturbances, the result of the linear optimization will exactly match that with the nonlinear constraints. Otherwise, there might be some loss of optimality since the linear constraints are more conservative than the original one, given that they restrict a greater region of the state-space. On the other hand, the linear constraints completely cover the obstacle, meaning that there is no possibility of the original constraint being violated in the optimization. It can, however, be violated after a disturbance acts on the system, an issue which will be addressed in Section 4.6.

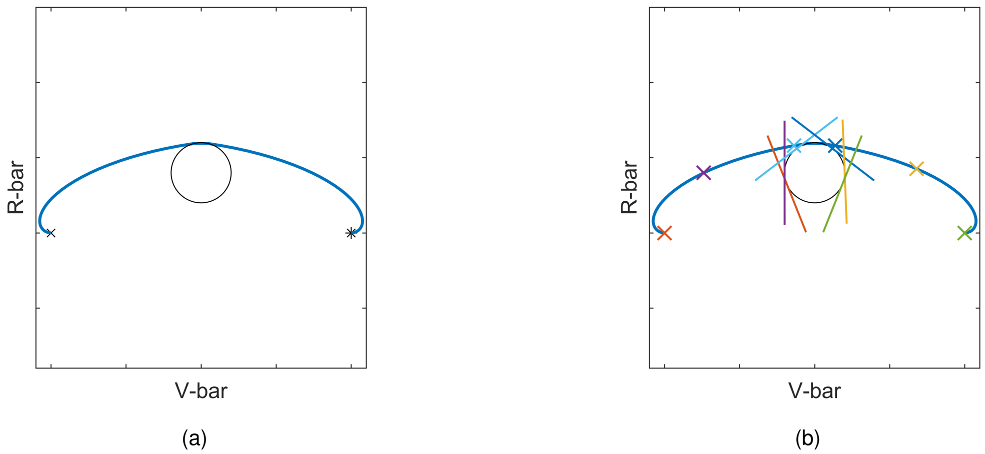


Figure 4.16: Illustration of the method for determining the linear obstacle avoidance constraints with offline nonlinear programming.

Finally, note that the optimization problem with the nonlinear obstacle constraints is non-convex, while the problem resulting from this technique is an LP and thus convex. Therefore, the LP will always find a solution that is close to the local minimum found in the NLP, which may not be a global minimum. However, because the NLP is solved offline, it is feasible to find the global minimum, which will correspond approximately to the minimum found by the online LP. This technique will be referred to as *obstacle avoidance with offline nonlinear programming* (OAONP) for the rest of this work.

Iterative Linear Optimization

Another possible way of determining the online linear constraints first proposed here is to use the unconstrained LP instead of the optimization problem with nonlinear constraints. Figure 4.17a) is the result of the unconstrained problem, where, naturally, violations of the obstacle avoidance constraint can occur. The tangent planes for this trajectory are determined, just like in the previous approach, and these are then used as linear obstacle constraints for LP optimization. This yields the result in figure 4.17b), where the trajectory now avoids the collision, but is more conservative than necessary since it flies-by the obstacle at a distance. The process can then be repeated, where this trajectory is used to again determine the linear constraints for another LP optimization. This yields the trajectory in figure 4.17c), which is very similar to that obtained previously with the nonlinear optimization in figure 4.16b).

It is then possible to achieve obstacle avoidance with a sequence of purely linear optimizations. However, note that in some situations the linear constraints determined from the first unconstrained optimization can make the feasible region empty, as will be shown in Section 4.7.3, which limits the use of this technique. Moreover, there is currently no guarantee that the trajectory converges to a local minimum of the problem with the nonlinear obstacle constraints, although that optimization problem may be warmstarted with the result from this technique for faster convergence.

This technique will hereby be referred to as *obstacle avoidance with iterative linear programming* (OAILP) and Algorithm 1 summarizes the strategy, where the stopping criteria may be the discrete states in the trajectory between iterations being equal up to a specified tolerance. It can be feasible to

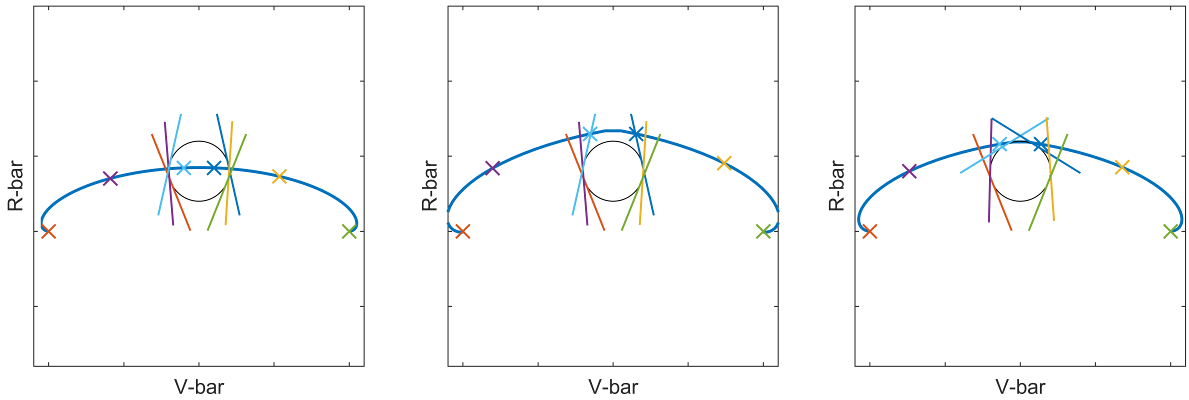


Figure 4.17: Illustration of the method for determining the linear obstacle avoidance constraints with iterative optimization with linear constraints.

use the OAILP technique online, since it relies purely on linear optimization. Alternatively, this algorithm may also be applied only once offline to determine the linear constraints, which are then used for online optimization, much like the OAONP strategy. This latter approach, however, will suffer from the same problem as the OAONP strategy, where the disturbances can make the linear constraints determined offline less optimal. In the presence of disturbances, it is then preferable to use the OAILP algorithm online, where the number of iterations is limited.

Algorithm 1: Obstacle Avoidance with Iterative Linear Programming

```
Solve optimization problem without obstacle constraints
repeat
    Determine planes tangent to obstacle facing each point in the trajectory
    Solve optimization problem with tangent planes as linear constraints
until trajectory convergence
```

4.5.4 Passive Safety Constraint

To formulate the passive safety constraint, the failure trajectories must be propagated with the prediction model, as first proposed by Breger and How [43], and then constrained with the methods presented in the previous sections. If a total thruster failure occurs at time k, the resulting free-drift failure trajectory x_{F_k} is described by

$$x_{F_{k,t}} = A_k^t x_k, \; t > k, \tag{4.14}$$

where A_k^t is the dynamic matrix that transitions the state from instant k to t, as illustrated in figure 4.18.

The passive safety constraint then becomes

$$\begin{aligned} x_{F_{k,t}} \notin Obstacle, \quad & k \in \{1, \dots, N\} \\ & t \in \{k+1, \dots, k+S\}, \end{aligned} \tag{4.15}$$

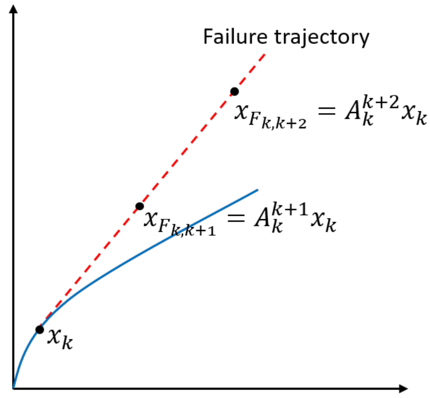


Figure 4.18: Propagation of the free-drift failure trajectories.

where failures at all discrete instants in the trajectory are considered and tracked for S samples, where S is the safety horizon. This can be seen as a type of move blocking strategy, as described in Section 2.4, where the control actions past the control horizon are zero.

Notice that, while at instant $k = 1$, the failure trajectory for instant $k = N$ is considered and constrained, which might seem conservative; this, however, allows the controller to generate a more accurate trajectory right away and avoid corrections later. However, this requires $N \times S$ optimization constraints, which have a great computational burden and thus require an efficient implementation, such as the OAONP or OAILP methods presented in Section 4.5.3. If those methods are utilized, each failure trajectory will have its own set of linear constraints. Also, an additional N constraints are required for the nominal trajectory collision avoidance. One way to reduce the online complexity is to check in the offline computation if some failure trajectories are very far from violating the constraint or superimposed with other trajectories, and remove the constraints associated to those, although this strategy is not employed in this work.

Note that this method does not consider thruster failures during a ΔV, considering only that thrusters fail to fire in the first place. Often those types of failures will also be covered as a consequence of applying the method, although this cannot be guaranteed. Because MPC operates in discrete time, it is not possible to consider mid-thrust failures at every continuous time-instant, although some additional discrete intermediate points can be considered, at the cost of greater computational complexity.

Another disadvantage of working in discrete time is that only the discrete states are constrained to be outside the obstacle, and not the whole continuous trajectory, which can lead to a collision if the time between samples is too great. This can be minimized by decreasing the sampling time or by including extra intermediate samples, both at the cost of a greater computation time. An alternative approach presented in [26] allows to constrain trajectories in continuous time, by transforming the optimization problem into a semidefinite program. This also comes at the cost of increased computation time, but completely eliminates the issue.

Finally, another limitation of this method is that it only guarantees passive safety within a finite horizon S. For most operations this is sufficient, since it gives ground operators enough time to react accordingly in face of a fault. Sometimes, however, it might be desirable to achieve passive safety with an infinite horizon. In [43] this is achieved by forcing all failure orbits to be invariant with respect to the target, via the constraint

$$x_{F_{k,k}} = A_k^{k+N_o} x_{F_{k,k}}, \quad k \in \{1, \dots, N\}, \tag{4.16}$$

where N_o is the number of samples in an orbit. This, however, can constrain the problem too much and easily make it infeasible, and in [43] it is only tested for very close-range operations and in a circular target orbit.

4.6 Robust Rendezvous

There are many sources of disturbances in a real rendezvous mission scenario, to which the controller must be robust. Firstly, there are modelling errors, since the prediction model used is a linearization of the real dynamics. Furthermore, this approximation emerged from a nonlinear model that assumed the gravitational field of the central body is uniform, which is never the case and is another significant perturbation on the model, of which the most significant for Earth satellites is the J_2 effect, due to the planet's oblateness.

There are also navigation errors on the position of the chaser relative to the target, especially when relative navigation is based on vision, such as in the PROBA-3 mission. Navigation uncertainty in the absolute position of the spacecraft is also present, which generates errors in the orbital parameters used for the prediction model. Another considerable perturbation is actuator errors, which are in magnitude, due to imprecise thruster action, in direction, due to error in the spacecraft orientation and thrusters mounting misalignment, and in burn duration, due to imprecise timing. Often, spacecraft thrusters can only be turned on or off, with no intermediate thrust possible. Thus, intermediate thrust commands are performed via *pulse width modulation* (PWM), as it is for the PROBA-3 spacecraft, which is another source of mismatch between prediction and reality. Furthermore, there is also a minimum thrust value that can be generated via PWM, due the delay in opening the thruster valves, and thus this is another perturbation on the system.

Finally, there are undesirable external forces acting on the spacecraft which are another source of disturbance, such as atmospheric drag, solar radiation pressure, or the gravitational effect of other massive bodies such as the Moon. These, however, affect both spacecraft and thus only the difference in forces generate a disturbance in the relative position, although absolute forces will change the orbit over time.

Robustness to these disturbances means not only that the controller remains stable, but also that it can still accurately converge to the reference, in approximately the specified manoeuvre duration and without a very significant increase in ΔV, in what we address here as *robust performance*. Furthermore, it is necessary that state constraints, such as passive safety, are not violated due to disturbances, which is known as *robust constraint satisfaction*. Finally, the disturbances can often push the system to a state that renders the optimal control problem infeasible, and thus the controller must have *robust feasibility*.

There is some inherent stability robustness for MPC [2], resulting from the fact that it is a closed-loop control strategy. Sometimes this is enough, though often robust strategies must be employed to increase performance and also to ensure state constraints are not violated due to a disturbance. Several robust strategies exist for general Robust MPC, such as Min-Max Feedback MPC [45] and Tube MPC [46], although it is often difficult to feasibly implement these in real-time. More on Robust MPC can be found in [2]. We will first review some of the robustness strategies presented in the literature specifically for MPC for rendezvous, before presenting our own contributions.

4.6.1 Robustness Techniques Review

In an early robustness technique for rendezvous, the authors How and Tillerson [47] extend the LP formulation and consider uncertainty in the initial condition, optimizing the trajectory simultaneously for multiple initial states. The technique is tested with nonlinear simulations that include differential drag, and although it is an improvement upon the non-robust controller, it is overly conservative and requires a ΔV much greater than in the unperturbed case. Also, no constraints at all are considered.

A different approach is taken by Richards and How [19], extending the VH-MPC formulation to yield robustness in face of an arbitrary unknown but bounded disturbance. Using the VH strategy solves the terminal constraint infeasibility that can easily occur for FH-MPC at the end of the manoeuvre, when the prediction horizon is short. Furthermore, the VH formulation is extended with correction control decisions in the next two steps which are limited given the disturbance bounds. This two-step correction of the disturbance guarantees that if the problem is feasible at the current step, it will also be at the next, thus achieving robust feasibility. Furthermore, the correction steps ensure that, given any disturbance, the state can converge to the nominal trajectory in two steps, hence granting robust stability. The downsides for this approach are that only bounded additive disturbances are considered, and a MILP formulation is required which can be difficult to implement in real-time. Also, the robust strategy required almost twice the nominal fuel for one simulation, and it does not address robust constraint satisfaction.

In [48], Deaconu et al. consider a bounded but unknown navigation error, and a technique based on the feedback-MPC strategy from the Tube-MPC framework is presented, meaning that the decision variables are feedback policies, parametrized as affine control laws, instead of control actions. The state uncertainty at each step is propagated as elliptical sets, and the optimal control problem becomes the minimization of the terminal uncertainty set, turning it into a convex conic optimization problem. The advantage in the use of feedback-MPC is that the full plan can be determined offline, while the online work becomes the computation of small disturbance correction terms, which is performed with simple algebraic computations. However, the fuel-optimality of the resulting controller is not evaluated, and more complex disturbances for which the computation of bounds is not possible are not considered. Furthermore, only control saturation constraints are included, and thus robust state constraint satisfaction is not addressed.

In [49], the same authors consider bounded execution errors in firing time and orientation. The terminal state constraint is substituted by a convex polytopic set, where its dimensions are minimized as decision variables such that it is guaranteed to contain the terminal state, taking into account the propagation of the uncertainty due to the bounded errors. By assuming the terminal polytopes are parallelotopes, linearizing the effect of orientation errors, and by introducing several auxiliary optimization variables, the problem can be formulated as an LP. A fuel budget constraint is included to limit propellant consumption, but because the cost function contains only the dimensions of the terminal set, the formulation is not necessarily fuel-optimal. Furthermore, magnitude thruster errors are not considered, although simulations are performed with the nonlinear dynamics including the J_2 effect and atmospheric drag. Navigation errors are also not considered, and this method is incompatible with the one in [48] due to necessarily very different formulations.

An approach that simply relies on the intrinsic robustness of MPC is presented by Di Cairano et al. [29] for in-plane proximity manoeuvring, using a quadratic cost and the Receding-Horizon strategy. The controller is shown to be robust for large actuation disturbances of up to $\pm$25% magnitude and $\pm$45 deg in orientation, and to solar pressure and atmospheric drag. However, the authors do not analyse fuel-consumption, although it can be seen that it is significantly greater than the ideal sparse manoeuvres, since the resulting trajectories are approximately straight-line approaches with significant non-sparse actuation. Furthermore, navigation errors are not considered and robust constraint satisfaction is not addressed. This work is extended by Weiss et al. [33], by including the out-of-plane motion, obstacle constraints, and formulating two different controllers for the rendezvous and docking phases of the mission, for which the first is complemented with a reference governor. The manoeuvres generated appear to require a ΔV orders of magnitude higher than is typically expected for these medium distance manoeuvres in a circular orbit. We strongly believe that a robust MPC controller for rendezvous should be based on the Finite-Horizon strategy with a cost function proportional to the ΔV, as was presented in Section 4.3.

An effective method for robust constraint satisfaction is *constraint tightening* [50], in which the bounds of state constraints are constricted along the horizon, in order to account for disturbances that could push the system to the infeasible region. The greatest advantage for this approach is that the complexity of the original problem is retained. However the tightened constraints can becomes too conservative, affecting performance and causing infeasibility. This technique is used in a rendezvous application by Breger et al. [31], where the tightened constraints are determined by propagating uncertainty sets with a preselected feedback nilpotent control law and given a bounded navigation disturbance, and computing the Pontryagin difference between the original constraint region and the uncertainty region. However, this method is not applicable when dynamics are time-varying, and is applicable with unbounded disturbances.

Gavilan et al. [32] present a probabilistic constraint tightening approach, labelled chance-constrained MPC, where all disturbances are assumed to be Gaussian and are estimated in real-time. The constraint bounds are then adjusted taking into account the uncertainty due to disturbances, guaranteeing that the original constraints are satisfied with a specified probability. Although many disturbances are not additive in nature, this approach shows good performance when simulated with the nonlinear dynamics, thrust magnitude and orientation errors, and unmodeled eccentricity, where the state constraint considered is a line-of-sight cone. However, only circular orbit dynamics are considered, and the issue of infeasibility when constraints become too conservative is not addressed. This technique is extended for a general elliptic orbit in [35], and also improved by exploiting the fact that navigation error Gaussian parameters are often known and don't need to be estimated. Furthermore, infeasibility in the terminal state constraint is dealt with a technique first presented by Tillerson et al. [51], where the terminal constraint is allowed to be relaxed into a terminal box in face of infeasibility. The technique presents good performance in the presence of nonlinear dynamics, J_2 effect, atmospheric drag, navigation errors and thruster errors, although the parameters chosen for these last two are quite small and not representative of real conditions, such as those in the PROBA-3 mission.

As can be seen, no single robustness strategy has yet emerged as the definite standard approach. These often neglect one or more of the core requirements for a robust MPC controller for rendezvous, such as fuel-optimality, being computationally feasible for real-time operation, manoeuvre accuracy, guaranteeing robust constraint satisfaction and robust feasibility, or being robust to all possible types of disturbances and uncertainty. The chance-constrained approach in [32, 35] comes close, since it takes into account all disturbances via online estimation in order to ensure constraint satisfaction without increasing the complexity of the optimization problem, uses the fuel-optimal FH-MPC formulation, and deals with infeasibility in the terminal state constraint. However, as will be shown later on, in the presence of navigation and thruster errors in the magnitudes expected for the PROBA-3 mission, the FH-MPC strategy faces some difficulty related to fuel expenditure and reference convergence. Furthermore, the constriant-tightening approach has not been tested with passive safety constraints, although with the obstacle avoidance techniques presented in 4.5.3 these can be written as linear constraints, to which the chance-constrained technique can easily be applied.

4.6.2 Feasible Terminal Box

In the presence of disturbances, the optimal control problem can often become infeasible, which should be avoided at all costs in a real-time application. The terminal state constraint in the FH-MPC strategy in particular is prone to this issue, since its prediction horizon is decremented until it is only one sample. Also, the system is not one-step controllable, meaning that it is not possible t o transfer between any two arbitrary states in just one control action. Therefore, although the controller initially plans for the terminal state to be reachable from the penultimate state, disturbances can push it to a position from which the terminal state cannot be reached in one step, rendering the optimization problem infeasible. This limitation can be further increased by other control and state constraints, and even more so when constraint tightening techniques are used.

One possible solution to the infeasibility problem is to use a quadratic terminal state cost as in (4.3), instead of the terminal constraint. This is undesirable, however, since it turns the optimization problem into a QP, and introduces more parameters for tuning. Another solution that is commonly used is to relax the equality constraint into an inequality, thus introducing the concept of *terminal box* [19, 26, 30, 43]. Constraint (4.4e) then becomes

$$-\delta_{box} \leq \bar{x}_N - x_{ref} \leq \delta_{box}, \tag{4.17}$$

where $\delta_{box} \in \mathbb{R}^6$ defines the bounds for the box. The issue with this approach is that to improve the guarantee of feasibility, the size of the terminal box has to be increased, which in turn worsens the accuracy of the controller since it will tend to aim for the edges of the terminal box. Furthermore, there is no guarantee that the chosen dimensions for the terminal box will always ensure the existence of a solution for all possible scenarios.

This approach can be improved upon by introducing the dimensions of the terminal box as optimization variables and including them in the cost function for minimization [51]. Introducing the optimization variables $\delta_1, \ldots, \delta_6$, the terminal constraint (4.17) becomes

$$-\begin{bmatrix} \delta_1 \\ \vdots \\ \delta_6 \end{bmatrix} \leq \bar{x}_N - x_{ref} \leq \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_6 \end{bmatrix}, \tag{4.18}$$

which remains a linear constraint with respect to all optimization variables. Furthermore, the cost function (4.4a) now includes the new variables

$$V(\cdot) \;=\; \sum_{i=0}^{N-1} \Delta t_i \mathbf{1}^\top \bar{u}_i \;+\; \sum_{j=1}^{6} h_j \delta_j, \tag{4.19}$$

where h_j is a large enough number as to ensure the controller only relaxes the terminal constraint to ensure feasibility, and not to save fuel. Thus, the terminal box will always have the minimum size that guarantees feasibility, and thus we designate this technique as *feasible terminal box*. Note that, although the box dimensions can be upper-bounded as to avoid the terminal box becoming too large, this should be avoided as it no longer guarantees the terminal constraint will always be feasible. It is also not necessary to lower-bound the box dimensions with 0, since negative values are already impossible. The obvious drawback for the feasible terminal box approach is that it requires the addition of six new optimization variables, although this is a very small number when compared to the usual dimensions of the control variable.

4.6.3 Dynamic Terminal Box

In the presence of stochastic disturbances, such as navigation or actuation errors, the terminal state constraint will cause the controller to perform frequent trajectory corrections, in an attempt to maintain the predicted terminal state exactly on the reference. However, because the controller acts on imperfect information and because its commands are not perfectly executed, this results in an overcorrection that results in the waste of fuel. Such an effect may be minimized by considering a terminal box constraint instead of a terminal equality constraint, thus loosening the terminal constraint and resulting in fewer trajectory corrections. This is similar to the feasible terminal box technique, although that was for preventing infeasibility whereas this is for improving the fuel consumption, and the two are not mutually exclusive. However, loosening the terminal constraint results in reduced manoeuvre accuracy. Thus, we propose here that the terminal box is reduced as the the manoeuvre is performed in what we designate as *dynamic terminal box*, thus achieving reduced fuel expenditure and maintaining accuracy.

One idea for achieving this is to dynamically change the weights h_j of the feasible terminal box technique, increasing them with time such that the box is increasingly tightened. However, because of the sparsity of the linear cost function, changing the box cost only results in either the terminal box having the minimum size that allows the problem to be feasible, or the the terminal box being completely loosened such that no control input is required. An alternative is to make the box cost quadratic, although this is undesirable since it makes the optimization problem a QP. Thus, the box margins have to tuned directly.

We propose here a modification on the feasible terminal box constraint to include a time-varying margin ε_t, where t is the time instant at which the optimization problem is being solved. Thus, constraint (4.18) becomes

$$-\begin{bmatrix}\delta_1\\ \vdots\\ \delta_6\end{bmatrix} - \varepsilon_t \leq \bar{x}_N - x_{ref} \leq \begin{bmatrix}\delta_1\\ \vdots\\ \delta_6\end{bmatrix} + \varepsilon_t. \tag{4.20}$$

Although this approach can achieve increased performance, it requires significant tuning of the margins for each time instant. In Section 4.7.4 we experiment with an initial box ε_0 that decreases linearly with time until it is zero at the final iteration. This may be improved, for example by determining the box dimensions as a function of the uncertainty, and thus may warrant further research.

4.6.4 Terminal Quadratic Controller

The sparse thrust profile of the FH-MPC formulation is not appropriate for executing accurate manoeuvres in the presence of stochastic disturbances such as navigation and execution errors. This is due to the fact that crucial ΔV's tend to be performed in one sample only, while planning under imperfect state information and with imperfect execution of the ΔV. One of these crucial ΔV's is the final braking thrust that is usually performed at the end of the manoeuvre in order to cancel all relative velocity, which under sparse actuation and in the presence of the mentioned disturbances tends to not be very effective.

Thus, we propose here the use of the following terminal linear-quadratic MPC controller to substitute the last sample of the FH-MPC manoeuvre

$$\min_{\substack{\bar{u}_0,\ldots,\bar{u}_{N_T-1}\\ \bar{x}_0,\ldots,\bar{x}_{N_T}}} \quad (\bar{x}_{N_T} - x_{ref})^\top Q_f (\bar{x}_{N_T} - x_{ref}) \; + \sum_{i=0}^{N_T-1} \bar{u}_i^\top R \bar{u}_i \tag{4.21a}$$

$$\text{s.t.} \quad \bar{x}_0 = x_t, \tag{4.21b}$$

$$\bar{x}_{k+1} = A_k^{k+1}\bar{x}_k + B_k^{k+1}\bar{u}_k, \tag{4.21c}$$

$$-u_{max} \leq \bar{u}_k \leq u_{max},\; k = 0,\ldots,N_T-1, \tag{4.21d}$$

where x_t is the state measurement/estimate at the penultimate iteration of the manoeuvre, and N_T is the prediction horizon of the terminal controller. The use of the quadratic cost for the input variable results in a less sparse actuation that, combined with the fact that one control decision is substituted with N_T decisions, allows for a more precise execution of the final braking ΔV. To maintain the manoeuvre duration specified initially, the prediction horizon N_T covers the same time window as the last sample of the FH-MPC controller. Furthermore, the FH strategy is also utilized, where the prediction horizon is decremented every sample to ensure the manoeuvre is completed in the specified time, although the terminal state constraint is substituted by a terminal quadratic cost. Finally, no intermediate state cost is used, since it decreases the controller performance, as shows in Section 4.2.

Since the prediction horizon for the terminal controller is not required to be very long, it is feasible to implement it with Explicit MPC, granting computational advantages which are desirable given that the braking manoeuvre is critical. Furthermore, because the terminal controller only has to cancel the relative velocity at the end of a manoeuvre that was planned with the FH-MPC formulation with passive safety constraints, the inclusion of these constraints for this controller may be unnecessary, further increasing the feasibility of implementing it with Explicit MPC.

4.7 Tests and Results

This Section features several simulations and experiments with the methods presented along this Chapter. The MPC optimization problems are solved with the MATLAB Optimization Toolbox, where function *linprog* with the dual-simplex algorithm is used for linear programming for the FH-MPC formulation, function *intlinprog* is used to solve the MILP in the VH-MPC formulation, and *fmincon* with the sequential quadratic programming algorithm is used for solving the nonlinear program that arises in the OAONP technique for the passive safety problem. Because these algorithms do not take advantage of the MPC problem structure, the state-substitution technique presented in Section 2.2.2 is always utilized. The computation times for solving these optimization problems are always presented, solved with a 4th Generation 2.4GHz Intel-i7 Processor.

4.7.1 FH-MPC

In this section, several rendezvous experiments are performed with the FH-MPC formulation, recreating several of the thrust manoeuvres presented in Chapter 3 and thus proving this formulation is indeed fuel-optimal. The V-bar transfer manoeuvre in one orbital period has already been validated in figure 4.10. The parameters and results for the following experiments are contained in table 4.3.

Figure	T_s	E_s	e	θ_0	N	ΔV	t_{max}	t_{avg}
4.19	$59.2\,\mathrm{s}$	-	0	-	50	$16.25\,\mathrm{mm/s}$	$8.84\,\mathrm{ms}$	$7.37\,\mathrm{ms}$
4.20	$58.3\,\mathrm{s}$	-	0	-	200	$3.45\,\mathrm{mm/s}$	$12.1\,\mathrm{ms}$	$7.79\,\mathrm{ms}$
4.21	$58.6\,\mathrm{s}$	-	0	-	100	$5.47\,\mathrm{mm/s}$	$11.6\,\mathrm{ms}$	$7.68\,\mathrm{ms}$
4.22	$59.2\,\mathrm{s}$	-	0	-	50	$10.83\,\mathrm{mm/s}$	$8.51\,\mathrm{ms}$	$7.38\,\mathrm{ms}$
4.23	-	$3.6\,\mathrm{deg}$	0.4	$0\,\mathrm{deg}$	100	$119.0\,\mathrm{mm/s}$	$9.26\,\mathrm{ms}$	$7.41\,\mathrm{ms}$
4.24	-	$1.8\,\mathrm{deg}$	0.8111	$180\,\mathrm{deg}$	200	$52.1\,\mathrm{mm/s}$	$10.3\,\mathrm{ms}$	$8.03\,\mathrm{ms}$
4.25	-	$1.70\,\mathrm{deg}$	0.8111	$179\,\mathrm{deg}$	100	$407.4\,\mathrm{mm/s}$	$9.52\,\mathrm{ms}$	$7.89\,\mathrm{ms}$

Table 4.3: Controller parameters and results for V-bar transfer manoeuvre simulations with FH-MPC.

In figure 4.19, a V-bar transfer manoeuvre with a duration of half an orbital period is presented. The trajectory and thrust profile generated by FH-MPC exactly matches the well-known V-bar transfer with two radial impulses, such as the one presented in figure 3.10, and the obtained ΔV is exactly the same as that computed with (3.51).

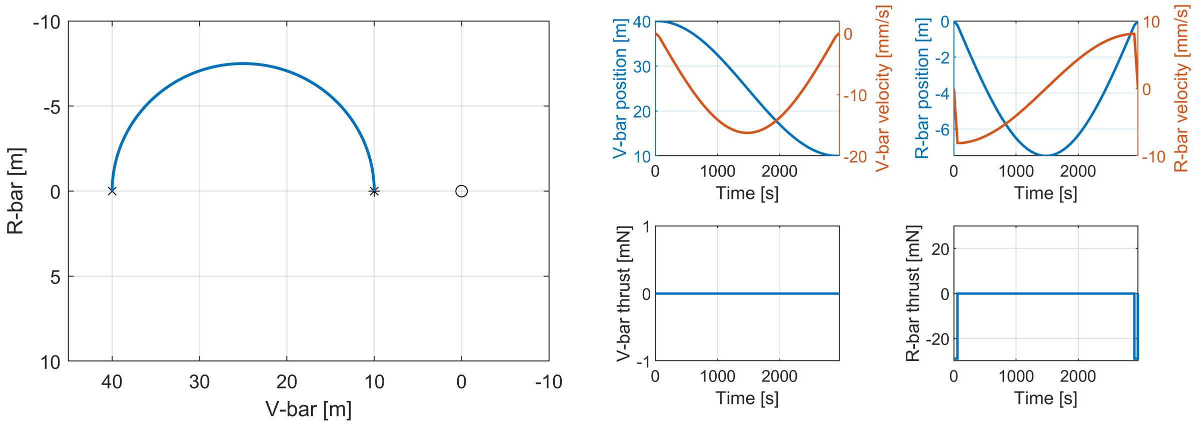


Figure 4.19: V-bar transfer manoeuvre in half an orbital period with FH-MPC.

Figure 4.20 presents another V-bar transfer, but now with a two orbit manoeuvre duration. It can be seen that the generated trajectory is much like the V-bar transfer manoeuvre with horizontal thrust, such as presented in figures 3.11 and 4.10, but with half the thrust such that it takes two orbits to reach the final state. The obtained ΔV is also the same as that obtained with (3.52) where Δx is half the total transfer distance.

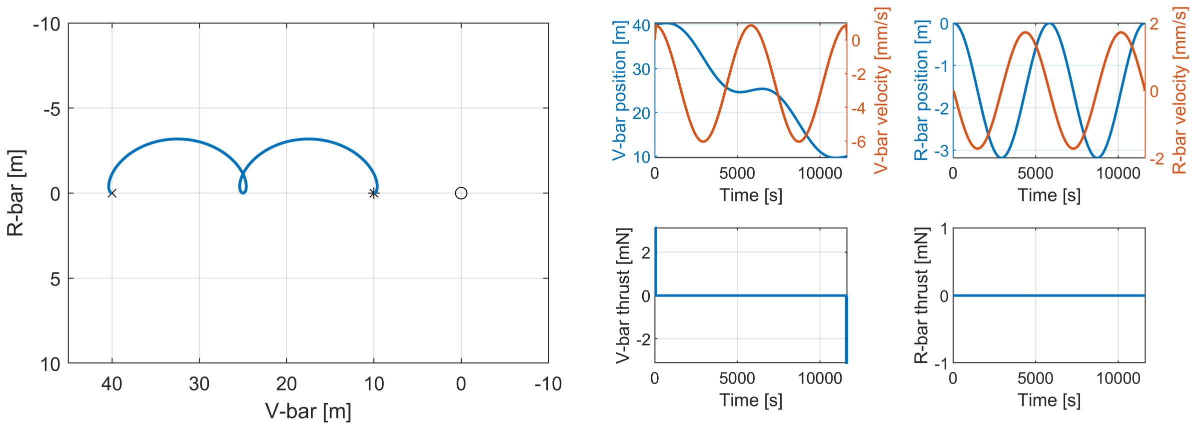


Figure 4.20: V-bar transfer manoeuvre with in two orbital periods with FH-MPC.

A one-orbit R-bar transfer manoeuvre is presented in figure 4.21. It can be observed that the resulting trajectory resembles the Hohmann transfer, such as the one presented in figure 3.9, and thus the manoeuvre is actually completed in only half of one orbit. The ΔV is also the same as that obtained with (3.50), although note that the FH-MPC manoeuvre is performed with asymmetric thrusts, generating a slightly different trajectory but equivalent regarding fuel consumption which suggests that the optimization problem is not strictly convex.

Next, in figure 4.22 an H-bar correction manoeuvre is performed in one orbital period. Because the ascending node is crossed after only half an orbit, the reference is reached after this period of time. The ΔV obtained is the same as that obtained for the inclination correction manoeuvre in figure 3.12 and

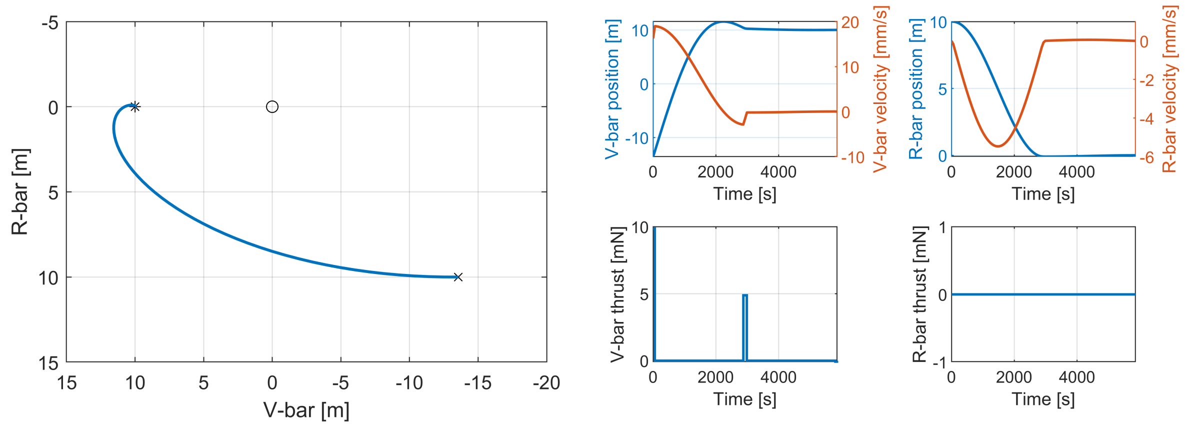


Figure 4.21: R-bar transfer manoeuvre in half an orbital period with FH-MPC.

computed with (3.53).

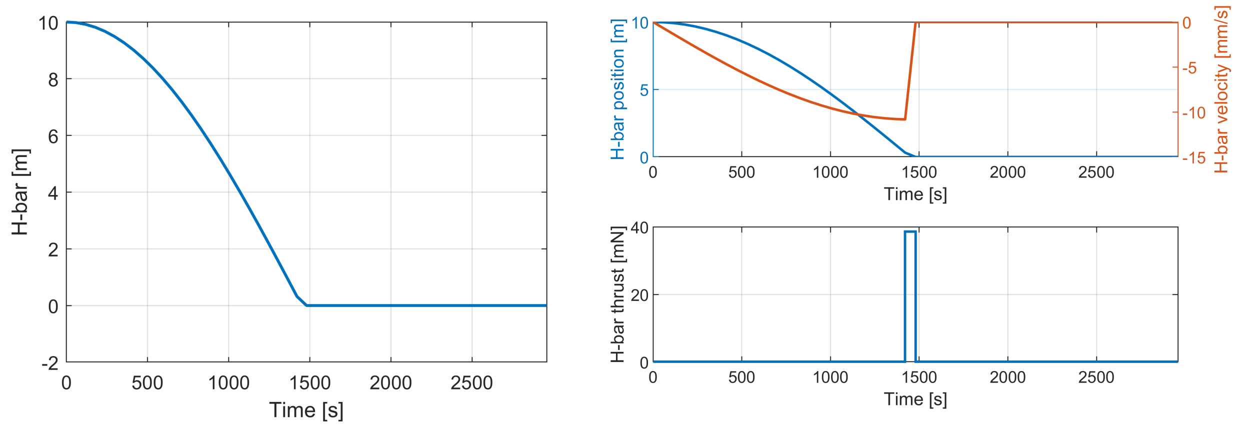


Figure 4.22: H-bar transfer manoeuvre in half an orbital period with FH-MPC.

Introducing now some eccenctricity, the manoeuvre presented in figure 4.23 is in the same conditions as that in figure 3.21. The trajectory obtained with the FH-MPC formulation is the same as the ideal one with the two impulses, and the ΔV required is the same as that computed with (3.54) and (3.56), since the generated manoeuvre also only has two thrust actions. Also note that, because the target orbit is now elliptical, the eccentric anomaly sampling technique presented in Section 4.1 is used, where E_s is the sampling eccentric anomaly, meaning that the initial and final thrust actions have different durations.

A manoeuvre with the eccentricity of the PROBA-3 RVX is presented in figure 4.24. It can be seen from table 4.3 that the ΔV obtained is 52.1 mm/s. On the other hand, the ΔV obtained for the ideal two impulse manoeuvre, which can be computed with (3.54) and (3.56), is 72.7 mm/s, which is significantly higher. The more efficient manoeuvre obtained with the FH-MPC formulation is due to the use of an intermediate thrust in V-bar around 30000 s, while the ideal manoeuvre is constrained to only an initial and final thrusts. This greater degree of freedom from the use of MPC thus allows it to generate more fuel-efficient trajectories than the traditional two-impulse manoeuvres.

Finally, a manoeuvre that includes H-bar and is an actual manoeuvre planned for the PROBA-3 RVX

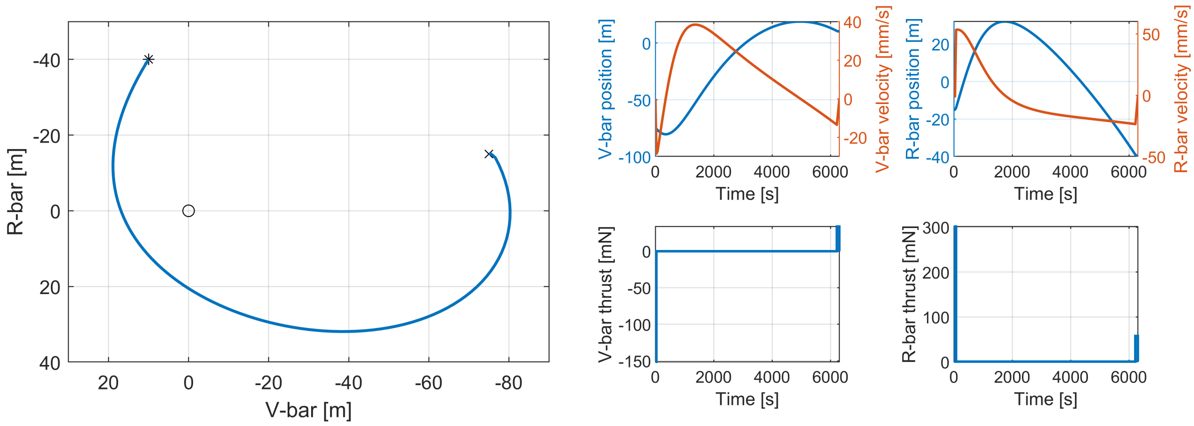


Figure 4.23: Arbitrary in-plane transfer manoeuvre in half an orbital period in elliptic orbit with FH-MPC.

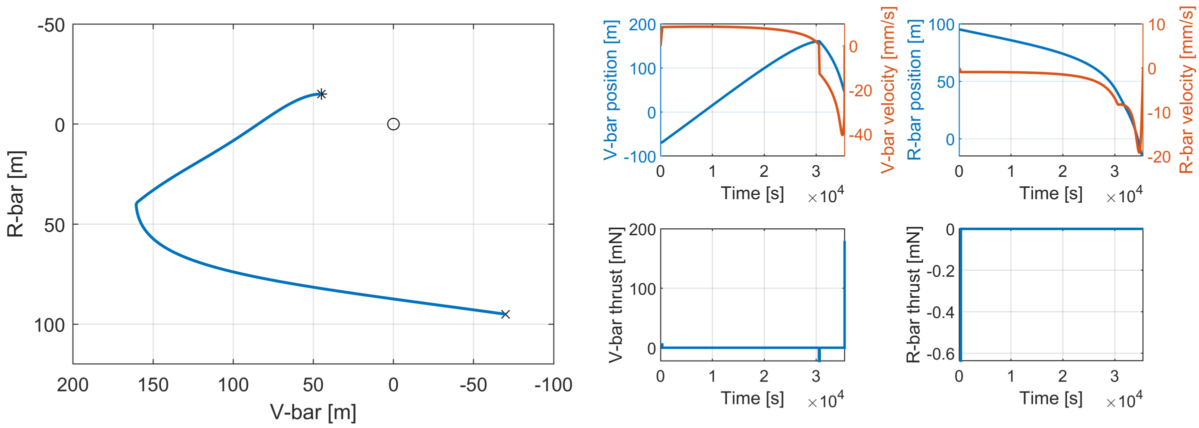


Figure 4.24: Arbitrary in-plane transfer manoeuvre in half an orbital period in PROBA-3 orbit with FH-MPC.

is presented in figure 4.25. It can be observed from table 4.3 that the required ΔV is 407.4 mm/s. On the other hand, the ΔV required for the two-impulse manoeuvre computed with (3.54) and (3.56) is 481 mm/s. Therefore, the FH-MPC formulation requires only 85% of the fuel that the manoeuvre that is typically employed requires, which, once again, is achieved via intermediate thrust action, this time seen in H-bar. We have thus showed in this section through simulation that the FH-MPC formulation is fuel-optimal.

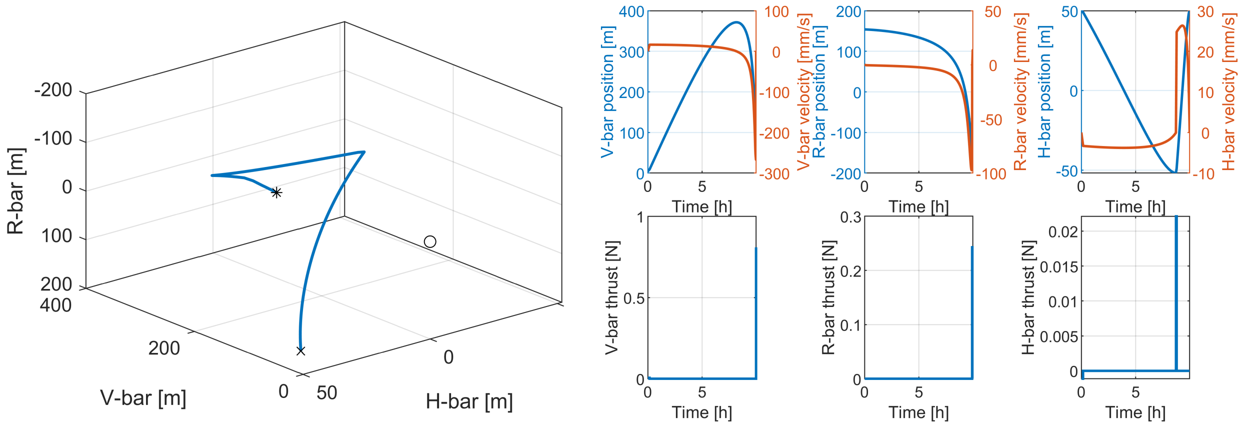


Figure 4.25: Manoeuvre from the PROBA-3 RVX with FH-MPC.

Regarding computational load, notice from table 4.3 that the increase of the prediction horizon has a relatively small impact on the execution time. For example, for an horizon of $N = 50$ the worst-case computation took 8.84 ms, while increasing the horizon by four times to $N = 200$ resulted in this value being 12.1 ms, which is only a approximately 37% increase. This is due to the fact that the optimization problem is formulated as a linear program, which can be solved very efficiently, in this case with the dual simplex algorithm. Furthermore, in these simulations there aren't any additional control or state constraints, which would increase the computational load. In the case of control constraints, however, this increase would not be very significant.

Figure 4.26 shows the effect of the prediction horizon on the execution time, in the conditions of the PROBA-3 manoeuvre presented in figure 4.25, but with control saturation constraints of 1/3 N in each component and maintaining the manoeuvre duration constant. For each prediction horizon the simulation was run ten times, and the execution times were averaged. It can be seen execution time grows approximately linearly, although the worst-case time oscillates unpredictably. This shows the computational advantage of formulating the optimization problem as an LP, where increasing the prediction horizon by a factor of 50 only increases the computation time by less than two and half times.

Figure 4.26: Computation time of the PROBA-3 RVX manoeuvre as a function of the prediction horizon.

4.7.2 VH-MPC

The Variable-Horizon formulation is useful to optimize both the manoeuvre duration and required fuel, where the trade-off between the two can be tuned with the parameter γ. If γ is zero, the solution will be the manoeuvre duration that minimizes the fuel only, where the duration is bounded by the maximum prediction horizon N_{max}. In simpler manoeuvres, however, the required fuel may be strictly decreasing with the manoeuvre duration.

Figure 4.27 shows the ΔV required for a V-bar transfer manoeuvre in a circular orbit as a function of the duration of the transfer. For a half-orbit transfer the result corresponds to that in figure 4.19, with one orbit we get the result from figure 4.10, and a two-orbit transfer corresponds to the result from figure 4.20. It can be then seen that the required the ΔV strictly decreases with the increase of the manoeuvre duration.

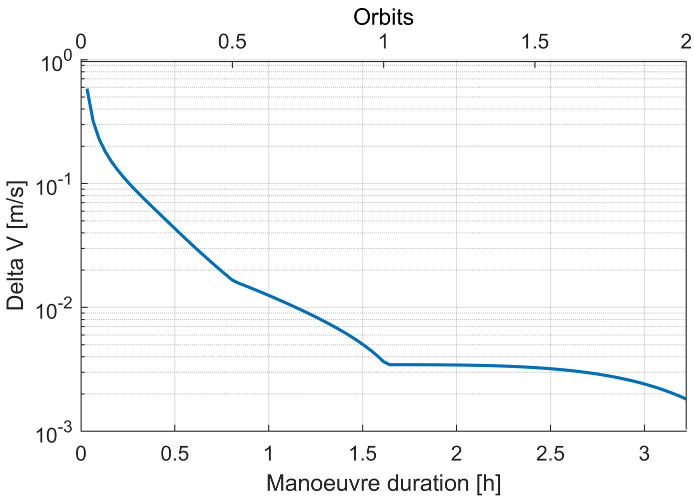


Figure 4.27: V-bar transfer manoeuvre in circular orbit ΔV as a function of its duration.

Applying the VH-MPC formulation in the conditions from figure 4.27 with a maximum manoeuvre duration of one orbit and $\gamma = 0$ yields the result from figure 4.28, where indeed the maximum transfer time is the optimum. The parameters and results for these experiments are presented in table 4.4.

Figure	T_s	E_s	e	N_{max}	γ	N	ΔV	t_{max}	t_{avg}
4.28	$98.1\,\mathrm{s}$	-	0	60	0	60	$3.45\,\mathrm{mm/s}$	$138\,\mathrm{ms}$	$54.9\,\mathrm{ms}$
4.29	$98.1\,\mathrm{s}$	-	0	60	0.01	30	$16.90\,\mathrm{mm/s}$	$184\,\mathrm{ms}$	$121\,\mathrm{ms}$
4.31	$57.9\,\mathrm{s}$	-	0	100	0	32	$130.9\,\mathrm{mm/s}$	$412\,\mathrm{ms}$	$277\,\mathrm{ms}$
4.33	-	$3.6\,\mathrm{deg}$	0.8111	100	0	41	$48.16\,\mathrm{mm/s}$	$710\,\mathrm{ms}$	$595\,\mathrm{ms}$

Table 4.4: Controller parameters and results for VH-MPC experiments.

If a manoeuvre duration cost is included ($\gamma > 0$), the optimal solution will have a shorter transfer time. In figure 4.29, the optimal transfer time becomes approximately half an orbital period, and the trajectory is similar to the ideal V-bar transfer with radial impulses presented in figure 3.10, although there is some actuation in V-bar. The total ΔV for that manoeuvre, determined with equation (3.52), is 16.3 mm/s, which is slightly lower than that obtained in this experiment due to the small V-bar actuation present there.

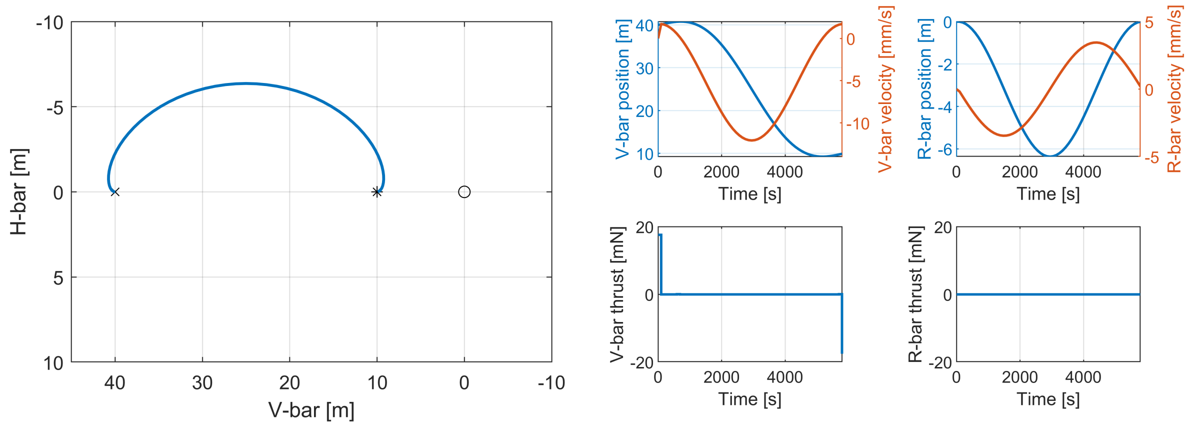


Figure 4.28: V-bar transfer manoeuvre with VH-MPC, with a maximum transfer of one orbit and no manoeuvre duration cost

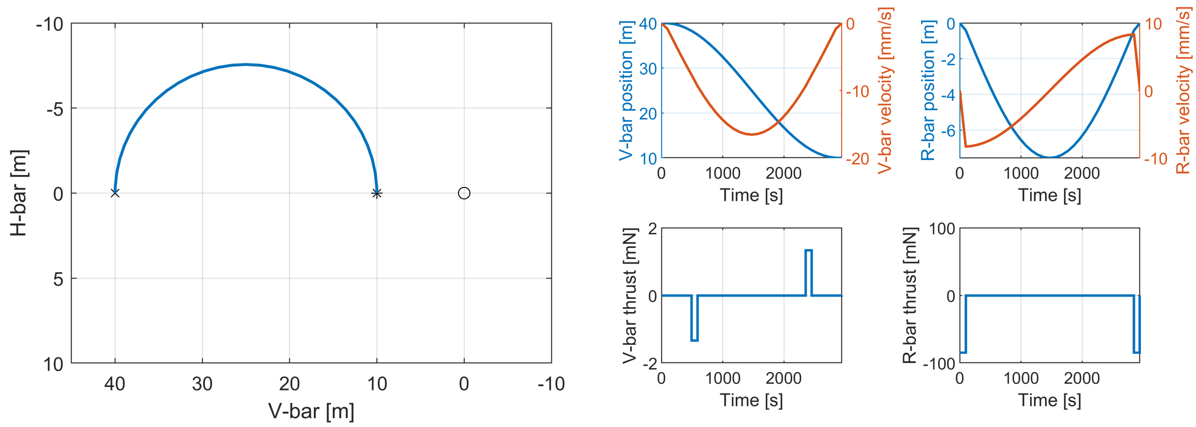


Figure 4.29: V-bar transfer manoeuvre with VH-MPC, with a maximum transfer of one orbit and a manoeuvre duration cost.

When the manoeuvre initial and reference points are not invariant, meaning that natural drift occurs if no thrust action is applied, then the manoeuvre ΔV may no longer be strictly decreasing with the duration. Figure 4.30 represents the ΔV as a function of the duration for a manoeuvre in such conditions, where the initial and final states are not on V-bar, and for a circular target orbit. It can be seen that the minimum transfer time within the two-orbit interval plotted is just over half an hour, at about 32% of an orbit. This happens because the chaser is drifting relative to the target, and thus there is an optimal time for performing the manoeuvre that is not the maximum time. When VH-MPC is applied in this scenario with a maximum duration of one orbit, it yields a manoeuvre with a duration that is approximately 32% of an orbit, as shown in figure 4.31, thus validating the VH-MPC formulation.

We will now analyse the PROBA-3 manoeuvre from figure 4.25. Figure 4.32 shows the ΔV required for that manoeuvre as a function of the transfer time. It can again be seen that the ΔV is not strictly decreasing with the manoeuvre duration, and that the peaks of this plot are much more pronounced than for the previous one. This happens because now the orbit is elliptical and the dynamics time-varying, meaning that there is another factor regarding the optimal time for performing the manoeuvre. For the PROBA-3 manoeuvre, there is a local minimum at around 40% of an orbit, but the globally optimum

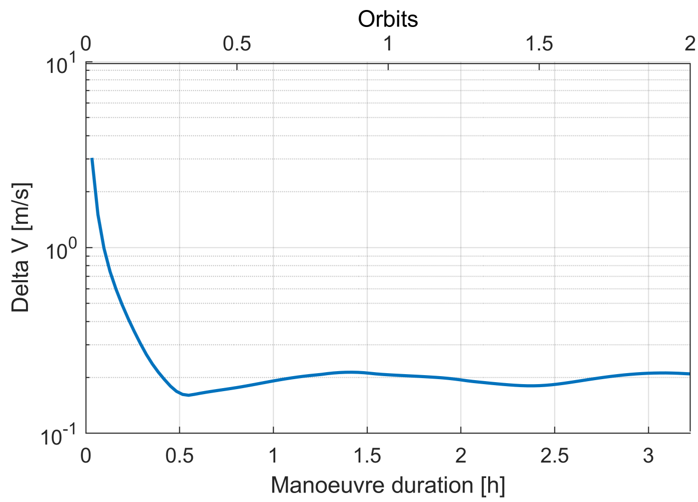


Figure 4.30: Arbitrary in-plane transfer manoeuvre in circular orbit ΔV as a function of its duration.

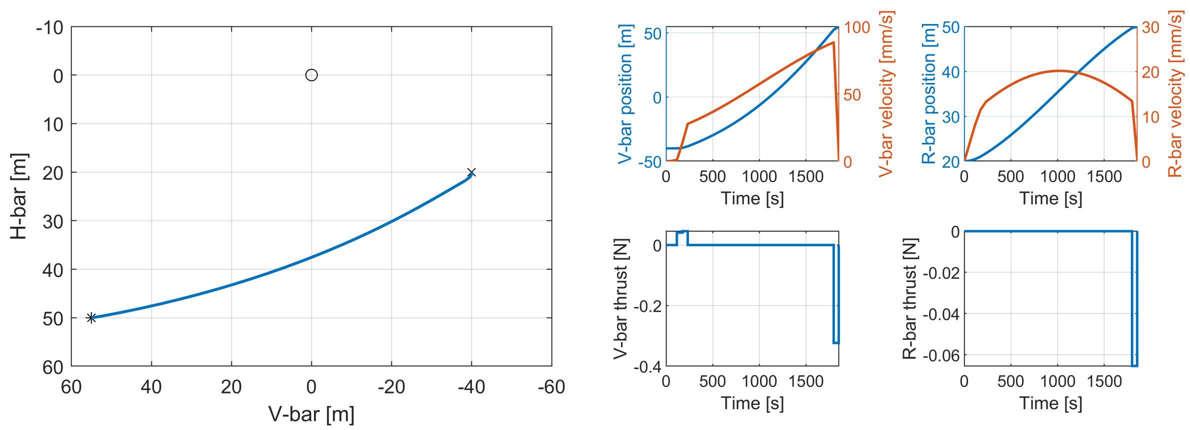


Figure 4.31: Arbitrary in-plane transfer manoeuvre with VH-MPC, with a maximum transfer of one orbit and no manoeuvre duration cost.

transfer time is one orbit, while there are peaks in the ΔV for half-orbit and one-and-a-half orbit transfers, the first of which corresponds to the manoeuvre duration from figure 4.25.

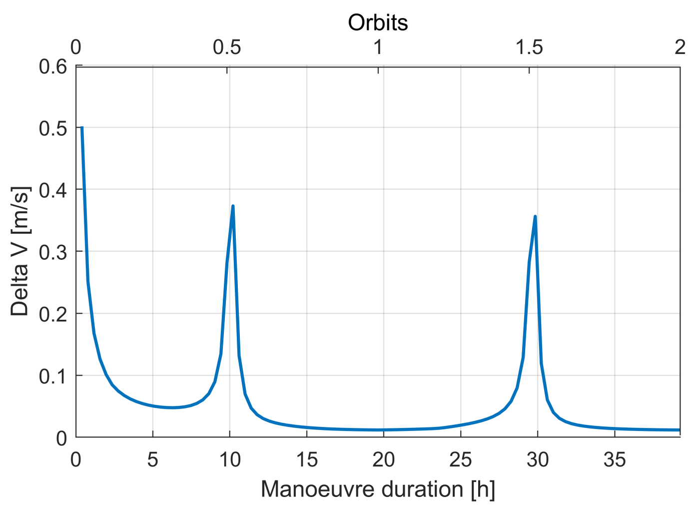


Figure 4.32: PROBA-3 manoeuvre ΔV as a function of its duration.

Applying VH-MPC to this manoeuvre with a maximum transfer time of one orbit yields the result from figure 4.33, where the optimal duration is 41% of an orbit. It can be seen from table 4.4 that the ΔV required is about eight times less than that obtained in figure 4.25. Although, this value does correspond to a minimum from the plot in figure 4.32, it is not the global minimum, which would be a duration of one orbital period. This is due to the fact that the algorithm used – branch and bound – does not optimize globally, and thus it converges to this local minimum. For online VH-MPC this can be a problem as the optimization may converge to a different minimum than previous iterations, which could affect performance; this can be avoided with the use of warm start. For offline VH-MPC it its feasible to optimize the problem globally.

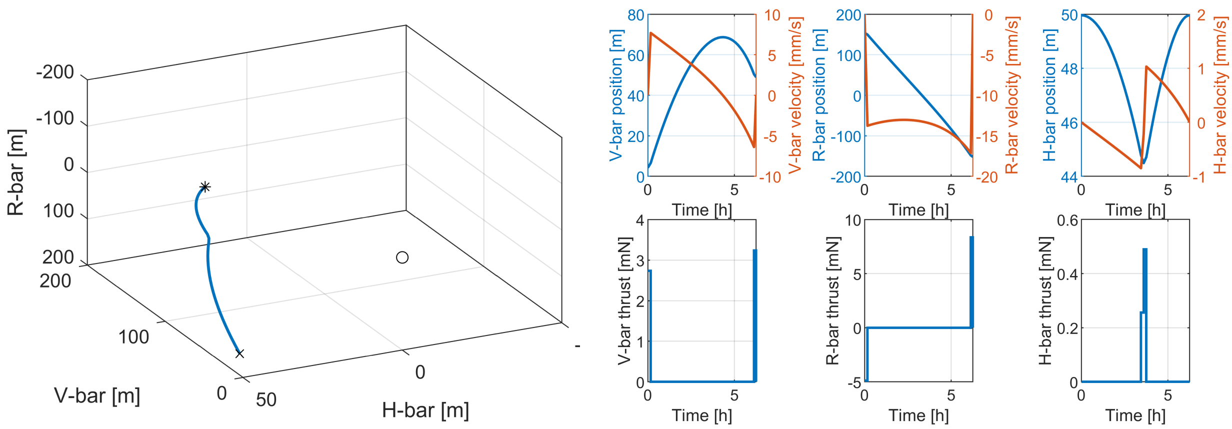


Figure 4.33: PROBA-3 manoeuvre with VH-MPC, with a maximum transfer of one orbit and no manoeuvre duration cost.

We see from table 4.4 that the computational load for the VH-MPC formulation is significantly higher than for FH-MPC, due to the fact that the first is a MILP and the latter an LP. It can then be infeasible to use this formulation online, especially with the inclusion of the very computationally heavy passive safety constraints.

4.7.3 Passive Safety

This section features simulations of rendezvous manoeuvres with the passive safety constraint. The OAONP and OAILP techniques for obstacle avoidance presented in Section 4.5.3 are utilized, which allow for the online optimization problem to be an LP. The first requires one offline nonlinear optimization; the latter requires several sequential linear optimizations, although this will only be performed once offline since no disturbances are considered, and thus the online work also becomes solving only one linear program. The simulation parameters and results are presented in table 4.5, where I_{LP} denotes the number of OAILP iterations.

Offline Nonlinear Programming

We will first consider the OAONP technique in order to achieve passive safety with online linear optimization. Figure 4.34 shows a one-orbit V-bar transfer without the passive safety constraint, and the

Figure	T_s	E_s	e	θ_0	N	S	I_{LP}	ΔV	t_{max}	$t_{offline}$
4.34	193.4 s	-	0	-	30	0	-	1.38 mm/s	12.3 ms	-
4.35	193.4 s	-	0	-	30	30	-	1.62 mm/s	21.7 ms	1.31 s
4.36	193.4 s	-	0	-	30	60	-	2.80 mm/s	33.2 ms	5.57 s
4.37	-	4.5 deg	0.8111	245 deg	40	0	-	1.49 mm/s	11.1 ms	-
4.38	-	4.5 deg	0.8111	245 deg	40	80	-	1.51 mm/s	81.7 ms	15.1 s
4.39	-	3.83 deg	0.8111	30 deg	45	0	-	161.4 mm/s	9.52 ms	-
4.40	-	3.83 deg	0.8111	30 deg	45	90	-	165.2 mm/s	139.8 ms	26.6 s
4.41	193.4 s	-	0	-	30	30	1	20.9 mm/s	18.1 ms	24.1 ms
4.42	193.4 s	-	0	-	30	30	3	1.62 mm/s	17.8 ms	212.4 ms
4.43	-	4.5 deg	0.8111	245 deg	40	80	1	1.53 mm/s	54.7 ms	56.3 ms
4.44	-	3.83 deg	0.8111	30 deg	45	90	1	165.2 mm/s	116.7 ms	63.9 ms

Table 4.5: Controller parameters and results for passive safety experiments.

failure trajectories, represented in red, are propagated for one orbit and starting from each discrete point in the nominal trajectory. Since there are no intermediate control actions, in this case all failure trajectories are superimposed. The target spacecraft safety region, which is a two metre radius circle, is also represented around the origin, which appears as an ellipse due to different scales being used. It can be observed that if the final thrust fails, a collision with the target spacecraft occurs after almost one orbital period.

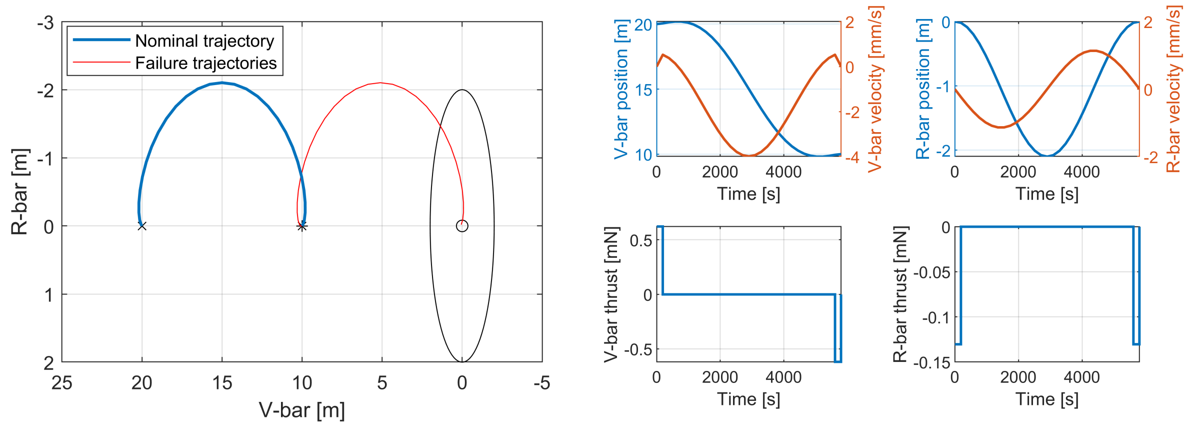


Figure 4.34: V-bar transfer manoeuvre in one orbit, in a circular orbit and without passive safety constraint.

In figure 4.35, a passive safety horizon of one orbital period is included. Initially the trajectory is the same as before, and toward the end of the manoeuvre there is some extra non-sparse actuation, which results in a slight widening of the end of the trajectory such that all failure trajectories now stop exactly at the edge of the safety region. Given the prediction horizon and safety horizon used, presented in table 4.5, this requires the addition of 900 optimization constraints, which nearly doubles the computation time of the online optimization when compared to the that without the passive safety constraint. However, note that the computation time of the offline nonlinear optimization used to determine the online linear constraints is two orders of magnitude higher. Notice also that, as could be expected, the ΔV for the passively safe trajectory has increased by approximately 17% in comparison to the non-safe one. Finally, the resulting closed-loop trajectory exactly matches that obtained obtained with the offline optimization despite using pure linear optimization, which is due to the absence of any disturbances.

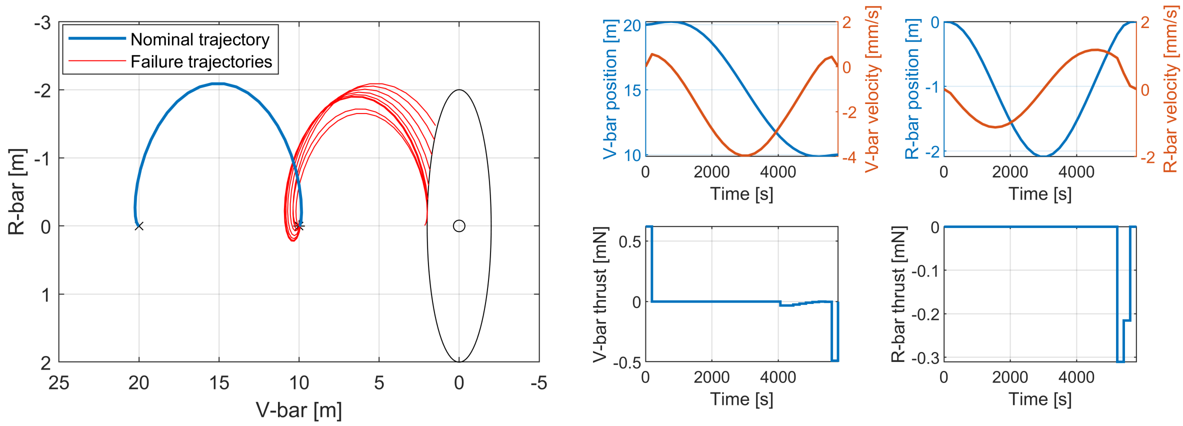


Figure 4.35: One-orbit V-bar transfer manoeuvre in one orbit, in a circular orbit and with one-orbit passive safety horizon with OAONP.

In figure 4.36, we repeat the previous manoeuvre but with a passive safety horizon of two orbital periods. Once again, we see that most of the trajectory is the same as the non-passively-safe one, but now the differences start earlier in the manoeuvre. The extra R-bar actuation before the 4000 seconds widens the approach in a such way that failure trajectories do not collide within two orbits. The ΔV for this manoeuvre is almost double that of the unsafe one, and represents an increase of over 70% in respect to the one with a one orbit safety horizon. Furthermore, the offline computation time increased by over four times, while the worst case online time increased by approximately 50%. Note that as the safety horizon increases, the R-bar actuation increases too. This results in the trajectory increasingly resembling the V-bar transfer manoeuvre with radial impulses, which, as mentioned previously in Section 4.5, guarantees passive safety in an infinite horizon but is over four times more costly.

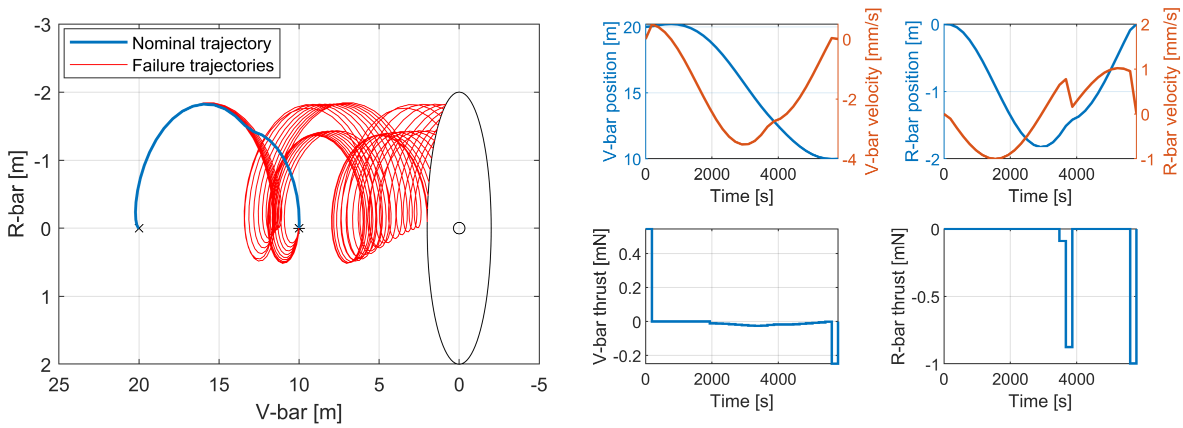


Figure 4.36: V-bar transfer manoeuvre in one orbit, in a circular orbit and with two-orbit passive safety horizon with OAONP.

Figure 4.37 shows an arbitrary in-plane transfer in the conditions of the PROBA-3 mission and without the safety constraints. It can be seen that both the nominal and failure trajectories violate the safety region. In figure 4.38 the passive safety constraint is included with a one-orbit horizon, where the nominal and failure trajectories are now safe with a minimal increase in ΔV, but with an eight-fold increase in the worst-case online optimization.

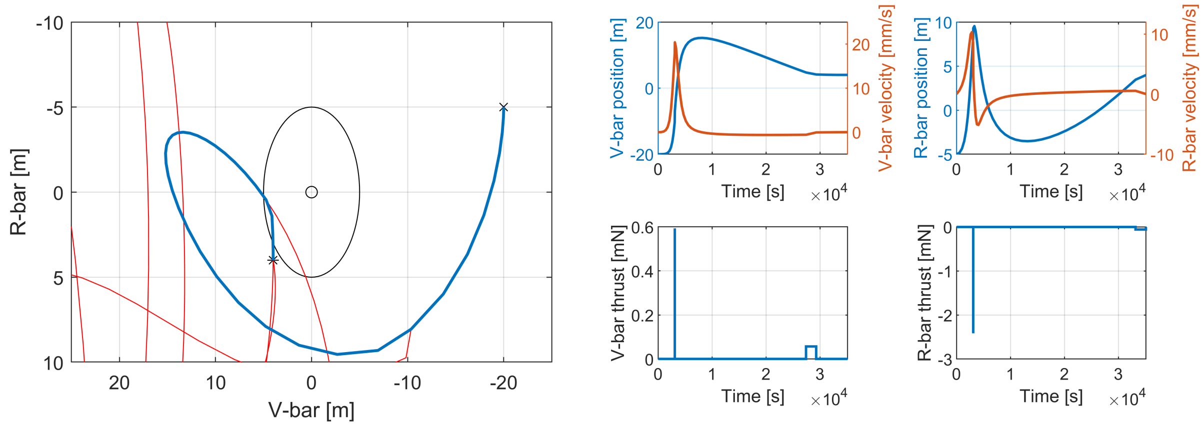


Figure 4.37: Arbitrary half-orbit in-plane transfer manoeuvre in PROBA-3 orbit and without safety constraint.

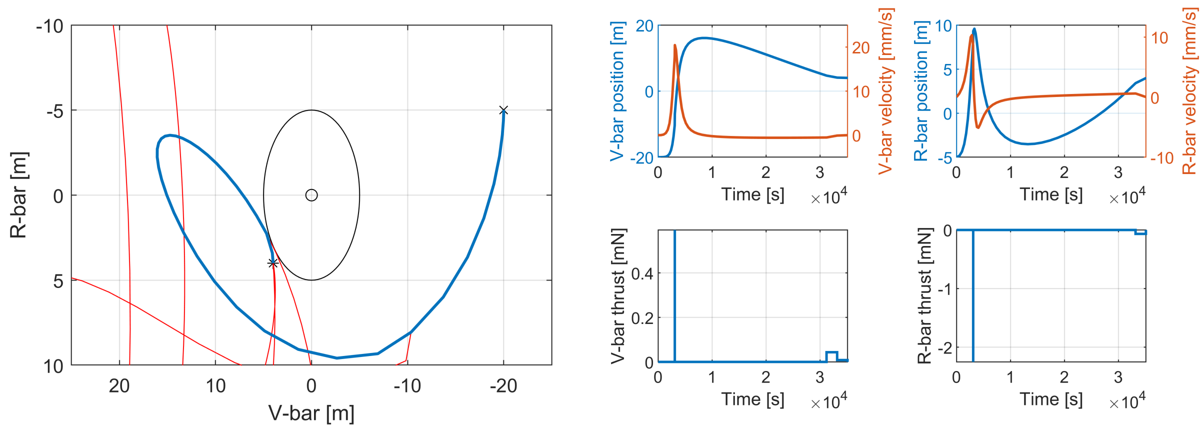


Figure 4.38: Arbitrary half-orbit in-plane transfer manoeuvre in PROBA-3 orbit and with one-orbit passive safety horizon with OAONP.

Figure 4.39 presents a manoeuvre with the addition of the third dimension, H-bar, and thus the safety region is now a sphere, with a radius of 10 metres. Without the passive safety constraint, it can be seen that one failure trajectory violates the safety region. In figure 4.40 the constraint is added and that failure trajectory no longer violates the safety region, at the cost of an increase in ΔV of 2.35%. It can be seen that most failure trajectories are overlapped or very far from the safety region, and thus after an offline analysis these could potentially be removed from the online optimization for better performance.

Iterative Linear Programming

We will now experiment with the OAILP technique presented in Section 4.5.3, where the offline optimization becomes a sequence of linear programs. In the same conditions as in figure 4.35, one iteration of this technique yields the result from figure 4.41. It can be seen that passive safety is indeed achieved, but in a much more inefficient way as that obtained with the nonlinear optimization, with more than ten times the cost in ΔV. On the other hand, the computation time for the offline work is now several orders of magnitude lower than before.

If three iterations of the OAILP are performed, the result from figure 4.42 is obtained instead, which

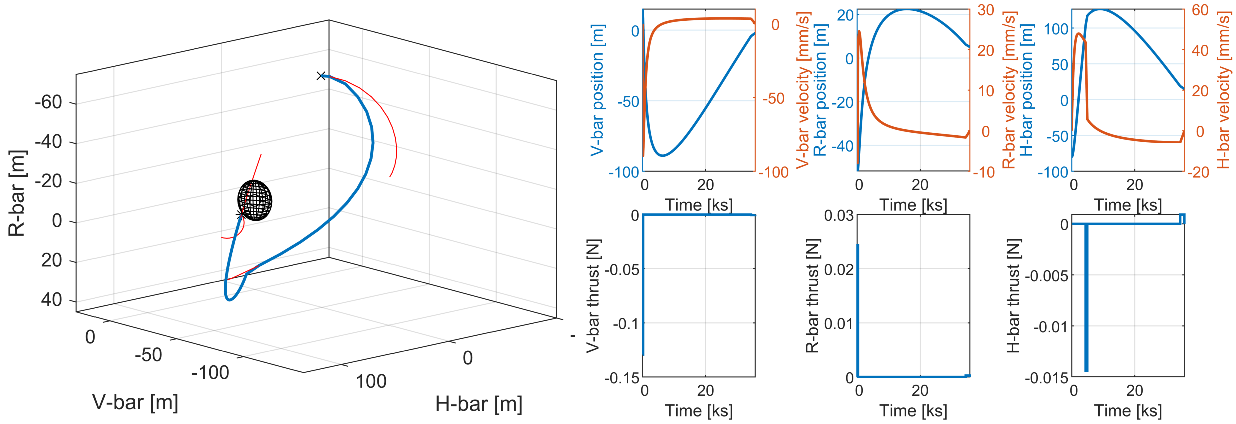


Figure 4.39: Arbitrary half-orbit transfer manoeuvre in PROBA-3 orbit and without passive safety constraint.

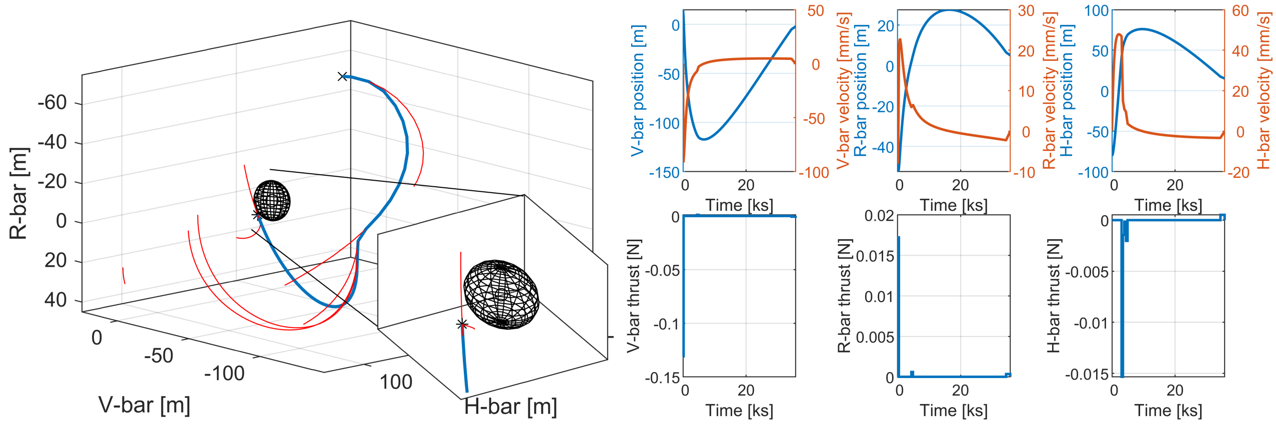


Figure 4.40: Arbitrary half-orbit transfer manoeuvre in PROBA-3 orbit and with one-orbit passive safety horizon with OAONP.

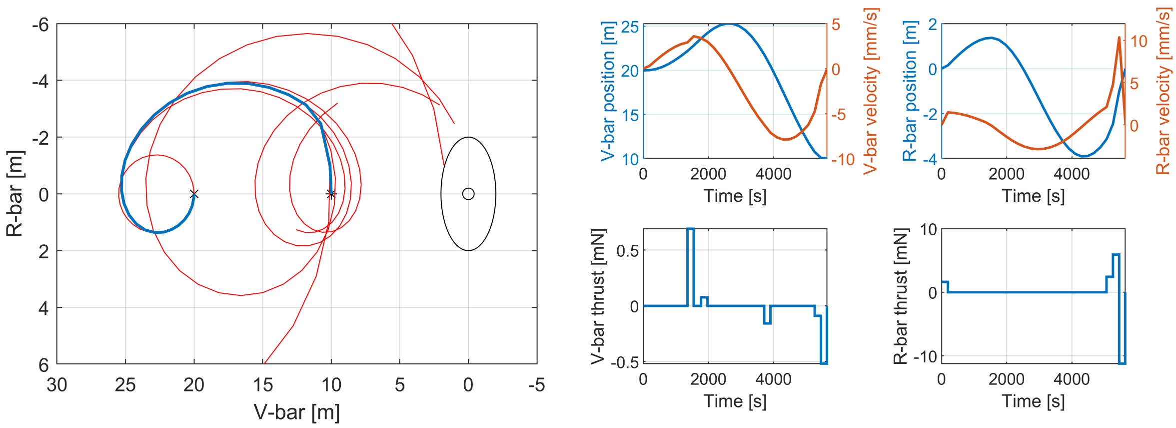


Figure 4.41: V-bar transfer manoeuvre in one orbit, in a circular orbit and with two-orbit passive safety horizon with OAILP.

exactly matches that with the nonlinear optimization from figure 4.35 and has the same ΔV. The offline computation time has also increased, but remains one order of magnitude lower than that of the nonlinear optimization. If the passive safety horizon is increased to two orbits like in figure 4.36, the linear constraints obtained after the first unconstrained optimization yield an empty feasible region, and thus the OAILP technique fails.

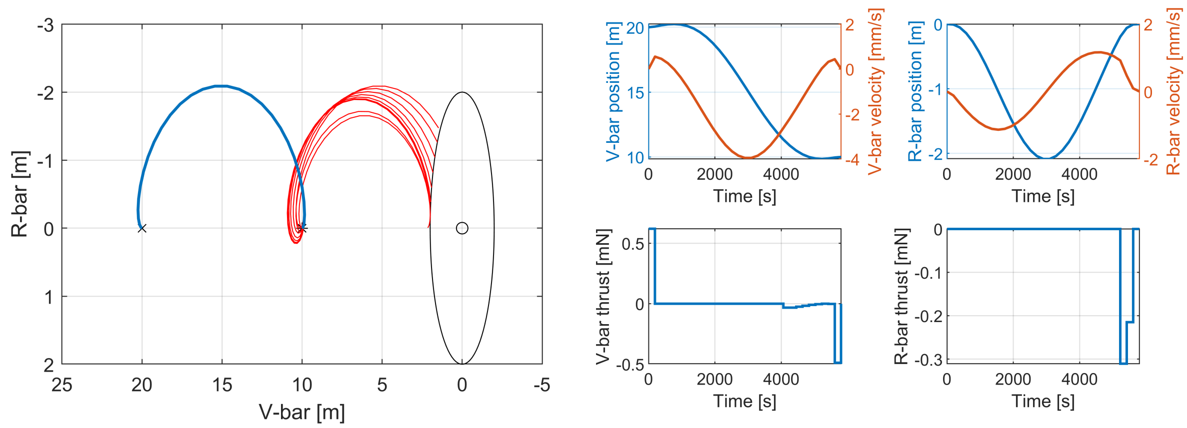


Figure 4.42: V-bar transfer manoeuvre in one orbit, in a circular orbit and with two-orbit passive safety horizon with OAILP.

Figure 4.43 is a manoeuvre in the same conditions as in figure 4.38. With OAILP passive safety with one iteration the closed-loop trajectory is very similar than that with the nonlinear optimization, with a small increase in ΔV of 1.3% but a very significant reduction in the offline work of three orders of magnitude. Note that the trajectory does not brush the edge of the safety region as it does with the nonlinear optimization, thus being more conservative. Although we do not show the result here, if two OAILP iterations are used, the result exactly matches that with the nonlinear optimization constraints.

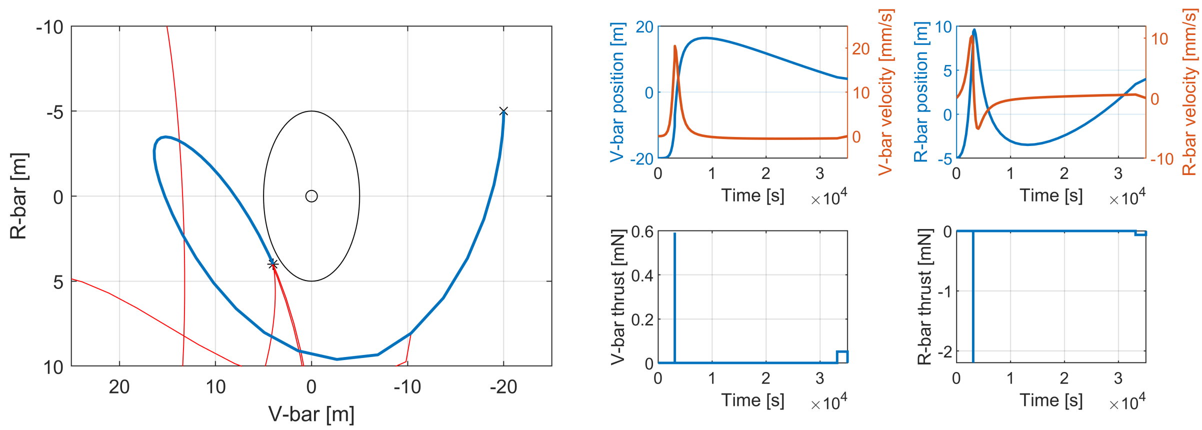


Figure 4.43: Arbitrary half-orbit in-plane transfer manoeuvre in PROBA-3 orbit and with one-orbit passive safety horizons with OAILP.

Finally, figure 4.44 presents the same manoeuvre as in figure 4.40 but with the OAILP strategy. It can be seen that the trajectory remains very similar and the ΔV is the same, even with just one iteration.

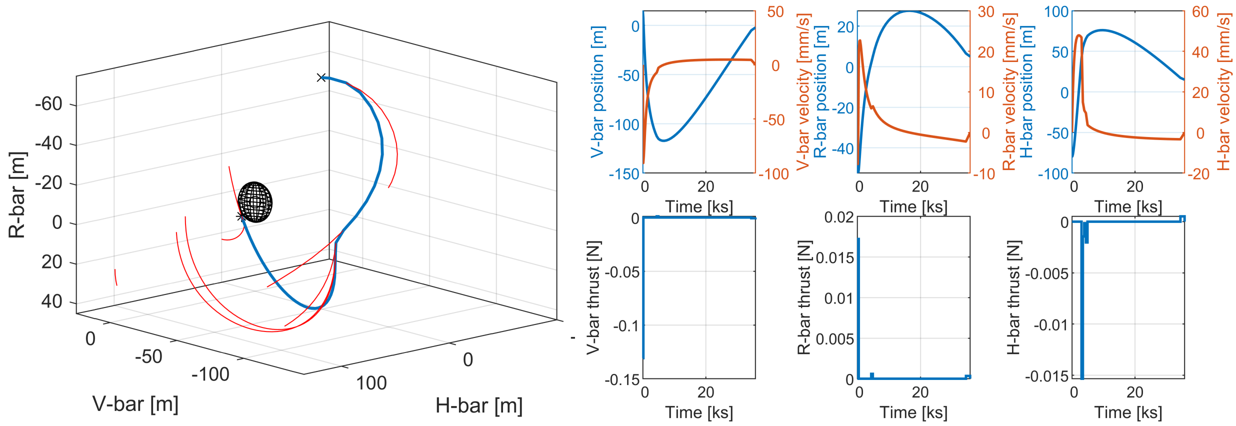


Figure 4.44: Arbitrary half-orbit transfer manoeuvre in PROBA-3 orbit and with one-orbit passive safety horizons with OAILP.

4.7.4 Robustness Experiments

This section will experiment with the FH-MPC strategy in the presence of disturbances, namely modelling errors, navigation errors and thruster errors. We will then apply the robust techniques covered in Section 4.6 to improve the robustness of the FH controller, regarding feasibility and performance. We will also show the need for robust constraint satisfaction techniques for the passive safety constraint, although no such techniques will be demonstrated here. The controller parameters and simulation results are presented in table 4.6, where e_{pos} and e_{vel} are the terminal errors in position and velocity.

Figure	E_s	N	ΔV	e_{pos}	e_{vel}	t_{max}	t_{avg}
4.45	$1.70\,\mathrm{deg}$	100	$407.4\,\mathrm{mm/s}$	$63.61\,\mathrm{m}$	$57.84\,\mathrm{mm/s}$	$12.7\,\mathrm{ms}$	-
4.46	$1.70\,\mathrm{deg}$	100	$129.6\,\mathrm{mm/s}$	$10.51\,\mathrm{m}$	$291.3\,\mathrm{mm/s}$	$14.4\,\mathrm{ms}$	$8.34\,\mathrm{ms}$
4.47	$1.70\,\mathrm{deg}$	100	$497.5\,\mathrm{mm/s}$	$25.44\,\mathrm{cm}$	$4.746\,\mathrm{mm/s}$	$10.1\,\mathrm{ms}$	$8.25\,\mathrm{ms}$
4.48	$0.340\,\mathrm{deg}$	500	$422.0\,\mathrm{mm/s}$	$4.196\,\mathrm{mm}$	$0.2457\,\mathrm{mm/s}$	$16.1\,\mathrm{ms}$	$10.8\,\mathrm{ms}$

Table 4.6: Controller parameters and results for robustness experiments with PROBA-3 manoeuvre.

A valid strategy for robustness to disturbances for MPC is to simply rely on its inherent robustness, due to it being a closed-loop strategy. Figure 4.45 exemplifies the performance of open-loop MPC in face of disturbances, with the same PROBA-3 manoeuvre presented in figure 4.25 and simulating the system with the nonlinear dynamics, instead of with the linearised model that is also used as the MPC prediction model and thus introducing modelling errors. Furthermore, control limits of 1 N are added in each thrust direction for this and all subsequent simulations. The chaser follows a similar trajectory as that in the absence of disturbances, having exactly the same ΔV, but the fact that its prediction model is imperfect results in a very significant terminal error of 63.6 m for position and 57.8 mm/s for velocity.

Performing the same manoeuvre but in closed-loop yields the result in figure 4.46. It can be seen that the controller can now better approach the reference state despite the prediction model not being perfect, although the trajectory is slightly different than what was obtained in the unperturbed case. However, the optimization problem becomes infeasible in the final iteration due to the terminal constraint, and thus the manoeuvre ends with a position error of 10.5 m and a relative velocity error of 291 mm/s. Thus, as

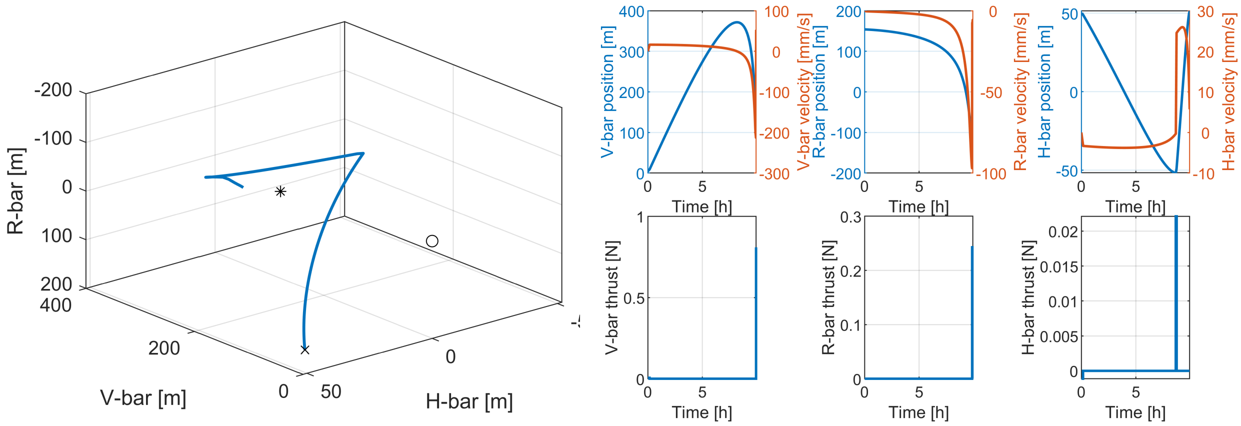


Figure 4.45: PROBA-3 manoeuvre with nonlinear dynamics simulation in open-loop.

previously mentioned in Section 4.6.2, this formulation suffers from robust feasibility issues.

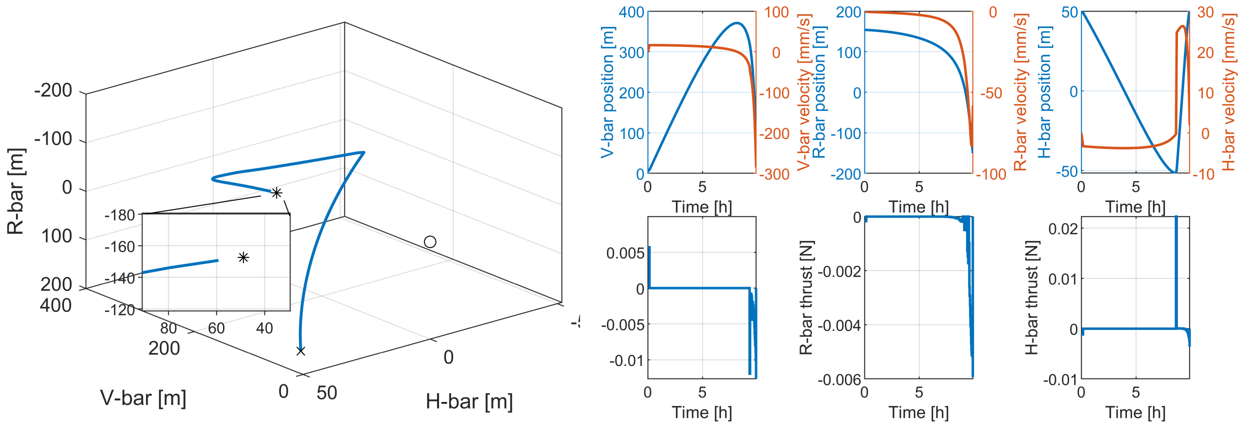


Figure 4.46: PROBA-3 manoeuvre with nonlinear dynamics simulation in closed-loop.

Robust Feasibility

To deal with infeasibility in the terminal state constraint, the feasible terminal box technique presented in Section 4.6.2 is applied. This yields the result from figure 4.47, where the optimization problem again becomes feasible at the final iteration, allowing for the manoeuvre to be completed, without any significant computational load increase. Despite this, it can be seen from table 4.6 that there is still a residual error in position and velocity of 24.4 cm and 4.75 mm/s, respectively, which is due to the fact that the feasible terminal box relaxed the terminal constraint, but also due to disturbances on the last step further affecting the system. Finally, the ΔV required to perform the manoeuvre has increased by 22% from that in the absence of this disturbance, which is a significant increase in fuel.

In figure 4.48, the prediction horizon is increased by five times, while maintaining the manoeuvre duration. The residual errors are now significantly smaller, at 4.2 mm and 0.25 mm/s, and the ΔV only represents an increase of 3.58% from the undisturbed case. Thus, modelling errors, or more specifically linearisation errors, can be compensated by increasing the prediction horizon. Despite this significant increase of the prediction horizon and respective increased performance, the computational load only

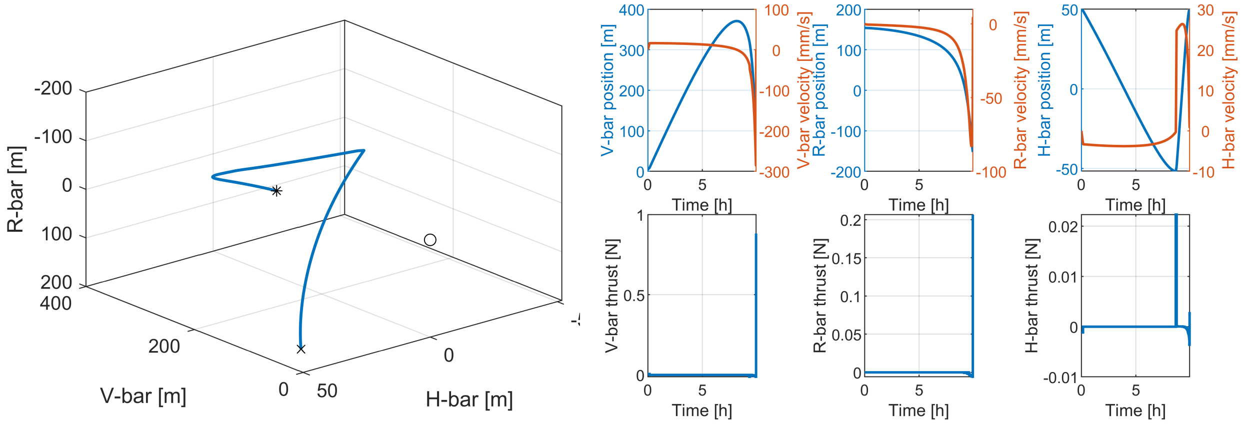


Figure 4.47: PROBA-3 manoeuvre with nonlinear dynamics simulation and feasible terminal box.

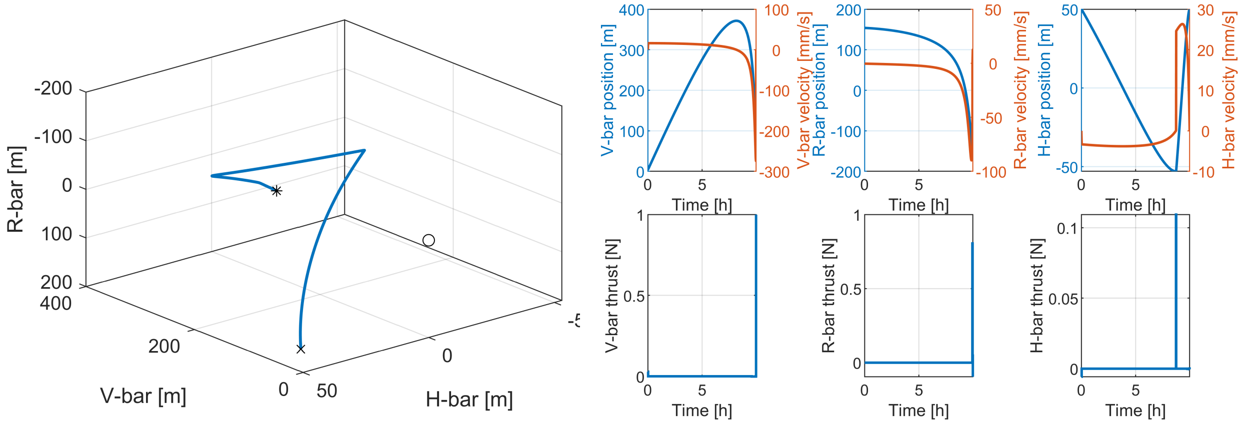


Figure 4.48: PROBA-3 manoeuvre with nonlinear dynamics simulation, feasible terminal box and increased prediction horizon.

increased by approximately 60% in the worst case, and 31% on average.

Robust Performance

We will now analyse and attempt to improve the fuel performance and accuracy of the FH-MPC formulation in face of other disturbances. We will consider additive Gaussian navigation noise in the measured state received by the controller at every step, with a standard deviation of 10 cm on position and 1 mm/s on velocity. Although these values may seem inconsequential, such a degree of uncertainty in the initial conditions may imply a dispersion of hundreds of meters after half an orbit. Thruster errors are also modelled, where the thrust magnitude has a standard deviation of 10% and the orientation of the thrust vector has a standard deviation of 0.5 degrees in each direction. Furthermore, the PWM cut-off thrust is modelled by ignoring control commands with a magnitude lower than 1 mN for each direction.

Because the simulation is now stochastic, 20 repetitions are performed and the results are averaged out and presented in table 4.7. In the above disturbance conditions yields the results from figure 4.49, where the dispersion between the trajectories can be observed, although all of them are able to converge on the reference. The thrust plot presented corresponds to only one of the simulations and is only meant to be representative. It can be observed that the control action is now less sparse, since now

Figure	E_s	N	$\varepsilon_{0,p}$	$\varepsilon_{0,v}$	N_T	ΔV_{avg}	$e_{pos_{avg}}$	$e_{vel_{avg}}$	$t_{max_{avg}}$
4.49	1.70 deg	100	-	-	-	577 mm/s	1.08 m	28.1 mm/s	9.72 ms
4.50	0.68 deg	250	-	-	-	899 mm/s	44.6 cm	13.4 mm/s	13.4 ms
4.51	1.70 deg	100	8 m	5 mm/s	-	499 mm/s	12.1 m	21.7 mm/s	11.34 ms
4.52	1.70 deg	100	5 m	5 mm/s	-	545 mm/s	79.7 cm	18.4 mm/s	9.98 ms
4.53	1.70 deg	100	5 m	5 mm/s	10	557 mm/s	28.6 cm	3.57 mm/s	9.54 ms

Table 4.7: Controller parameters and results for stochastic robustness experiments with PROBA-3 manoeuvre.

the controller corrects the trajectory at every step in an attempt to satisfy the terminal state constraint. This results in an increased ΔV of 41% on average in respect to the unperturbed case, which is very significant. We also see an average terminal state error of 1.1 m and 28 mm/s.

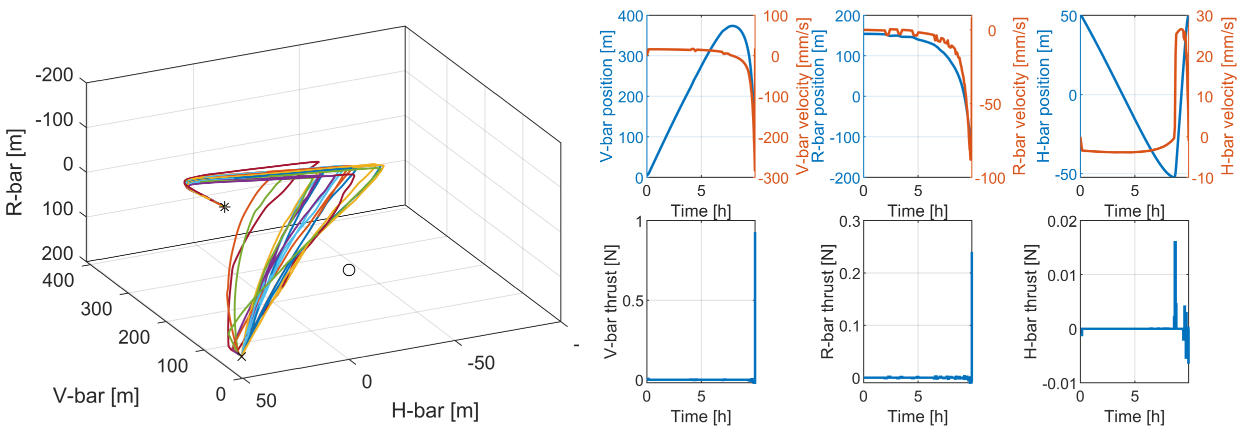


Figure 4.49: PROBA-3 manoeuvre with nonlinear dynamics simulation, navigation and actuator errors.

In an attempt to decrease the required fuel, we increase the number of samples in the prediction horizon. The result is presented in figure 4.50, where the the trajectories are less dispersed and the residual error is smaller. However, the average ΔV has increased to more than double of that in the undisturbed case. This is due to the fact that now the controller performs more correction manoeuvres, which are planned based on imperfect information, and thus more fuel is wasted in attempts to drive the predicted terminal state to the reference. Furthermore, these corrections are not performed as planned, due to actuator errors, inciting further corrections. Thus, increasing the prediction horizon has the opposite effect on fuel expenditure in the presence of navigation and actuator errors than solely in the presence of modelling errors, although terminal error does improve.

To decrease the controller sensitivity to stochastic disturbances, we substitute the terminal state constraint with a terminal box, as discussed in Section 4.6.3. Using an 8 m terminal box for position ($\varepsilon_{0,p}$) and 5 mm/s for velocity ($\varepsilon_{0,v}$), and maintaining these dimensions constant throughout the manoeuvre, yields the result from figure 4.51. There is now less manoeuvre correction, which results in a significant decrease of 78 mm/s in respect to the result from figure 4.49. However, the terminal error for position is significantly higher at 12 m, which is due to the loosening of the terminal constraint, although the terminal error for velocity is comparable.

To maintain accuracy and reduce sensitivity to disturbances, the dynamic terminal box approach presented in Section 4.6.3 is now used, with a linear decrease with time of the terminal box dimensions.

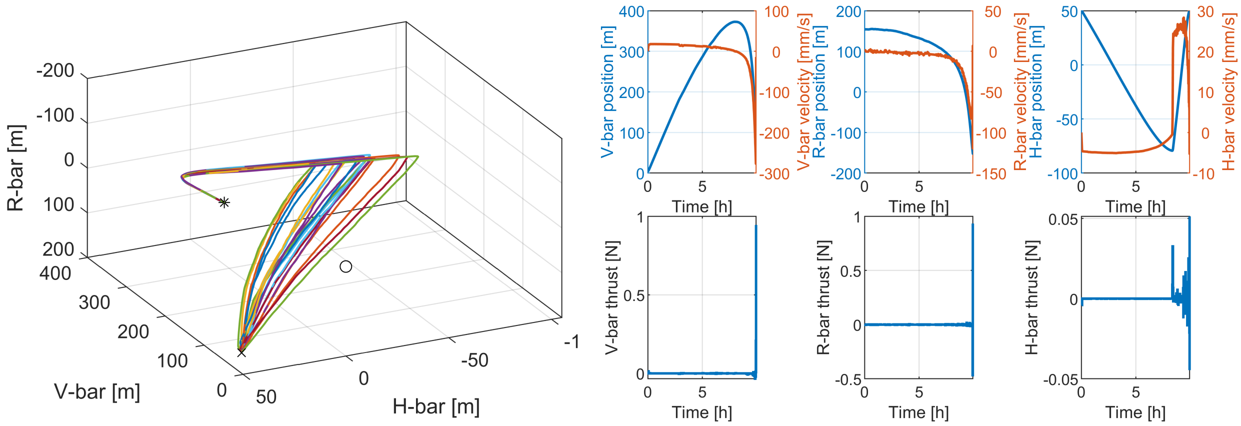


Figure 4.50: PROBA-3 manoeuvre with nonlinear dynamics simulation, navigation and actuator errors, and increased prediction horizon.

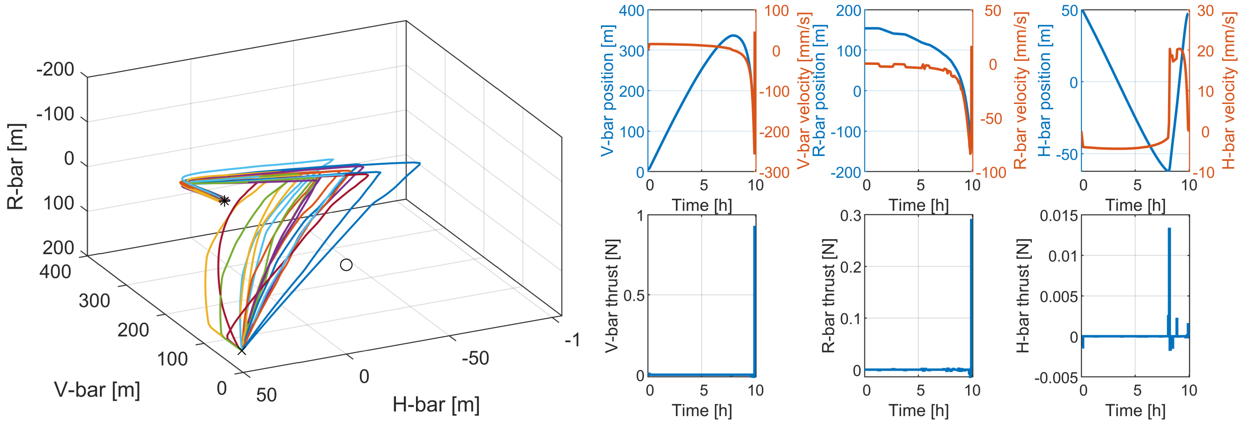


Figure 4.51: PROBA-3 manoeuvre with nonlinear dynamics simulation, navigation and actuator errors, and terminal box.

With an initial box size of 5 m and 5 mm/s, yields the result from figure 4.52. The ΔV has increased from the previous simulation, although it remains below that without the use of a loose terminal box. Furthermore, the terminal error for position and velocity has improved in comparison to the result from figure 4.49. Although the gain in ΔV is not significant, this result validates this type of approach for achieving better fuel performance in face of stochastic disturbances. With further parameter tuning and a different method for varying the terminal box dimensions, even better performance might be achieved.

Finally, we note that the terminal velocity at the end of the manoeuvre is significant, at 18 mm/s for the previous simulation. This value is still significantly higher than the navigation uncertainty, and thus can be improved. As mentioned in Section 4.6.4, this is due to the sparsity of the FH-MPC controller. Thus, we attempt to improve this by suggesting the use of a terminal quadratic controller that substitutes the final iteration of the nominal controller, as described in that section. With a prediction horizon of $N_T = 10$, an input cost matrix of $R = I$ and terminal state cost matrix $Q_f = \text{diag}\,[1, 1, 1, 10^4, 10^4, 10^4]$ yields the result from figure 4.53, where the thrust profile of the quadratic controller is shown. We see some increase in the ΔV, although it is less than the decrease in residual velocity, which at 3.6 mm/s

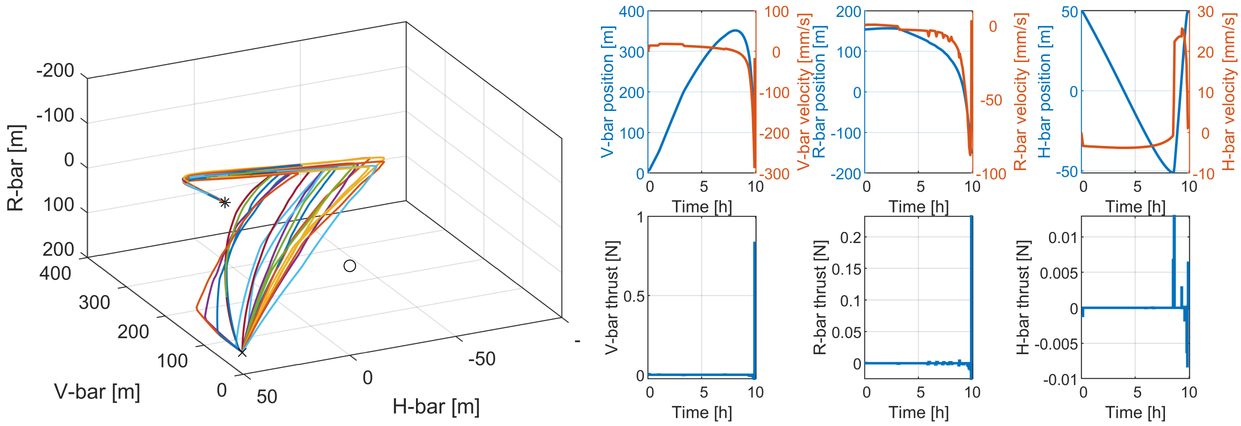


Figure 4.52: PROBA-3 manoeuvre with nonlinear dynamics simulation, navigation and actuator errors, and dynamic terminal box.

is comparable to the navigation uncertainty, alike the residual position which is now only 29 cm. The terminal controller only operates during what would instead be the last sample of the FH-MPC controller, and thus incurs no increase in manoeuvre duration. Better performance might be achieved with more tuning of the terminal horizon N_T or the terminal cost matrices R and Q_f, although this result validates this approach of achieving better accuracy.

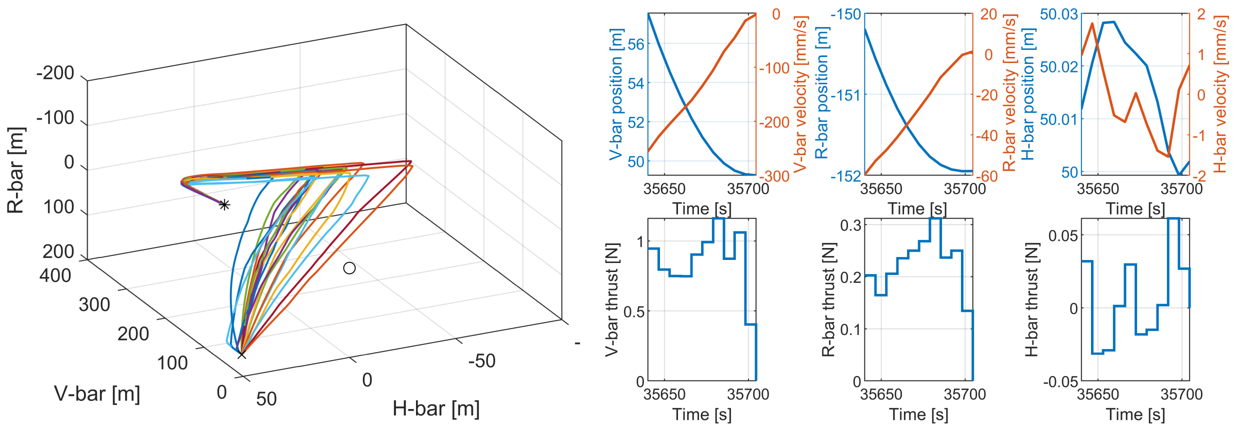


Figure 4.53: PROBA-3 manoeuvre with nonlinear dynamics simulation, navigation and actuator errors, dynamic terminal box and quadratic terminal controller.

Chapter 5

Conclusions

Model predictive control is an appropriate guidance and control strategy for use in orbital rendezvous. By using a cost function proportional to the fuel spent, a terminal state constraint, and the fixed-horizon strategy, a fuel-optimal formulation is obtained, where the manoeuvre duration is pre-defined. Furthermore, because this cost function can be re-written as a linear one, and because the relative dynamics between the spacecraft can be accurately linearised via the Yamanaka-Ankersen state transition matrix [16], allowing for the use of a linear prediction model, this fuel-optimal formulation known as FH-MPC [19] becomes a linear program, which can be optimized very efficiently and thus possibly enabling real-time use, although this also depends on the specific hardware of each mission.

The FH-MPC formulation presents advantages in fuel-consumption over traditional guidance techniques, which usually rely on two-impulse manoeuvres. Since MPC is not constrained to just two control decisions, intermediate thrust actions are naturally exploited by the controller to generate more efficient trajectories which are not possible with the two-impulse approach. This becomes especially advantageous for manoeuvres in elliptical orbits, such as the PROBA-3 mission rendezvous experiment [4], where the dynamics are time-varying and the instants at which burns are executed become more critical. However, the FH-MPC formulation only optimizes the fuel for a given pre-defined transfer duration. The VH-MPC formulation thus allows for optimizing both manoeuvre duration and fuel, by formulating the problem as a MILP [19]. Although the online use of MILP may be infeasible, VH-MPC can be used to determine offline the optimal manoeuvre duration, then executing the manoeuvre with FH-MPC.

The main advantage of the MPC strategy over other control strategies is that it allows for the explicit modelling of control and state constraints [2]. In a rendezvous scenario, this is useful to model thruster limitations, which can be accomplished realistically with linear constraints if the spacecraft has multi-directional thrust, as does the PROBA-3 chaser spacecraft, thus maintaining the FH-MPC formulation as an LP. Another crucial operational constraint in close-proximity operations is passive collision safety [43]. This requires, for each failure trajectory, the addition of several obstacle avoidance constraints, which are naturally non-convex and thus incur in a high computational cost. These constraints have previously been formulated as linear constraints [29, 30, 33, 43, 44], though with limited applicability. Thus, this work proposed two new approaches for achieving collision avoidance with linear constraints.

Firstly, The OAONP approach relies on a single offline nonlinear optimization with the original obstacle constraints, from which linear constraints are determined for online use. On the other hand, the OAILP technique performs a sequence of optimizations with linear constraints such that the solutions converge to that with the nonlinear constraints. Although simulations showed this after a small number of iterations, convergence has not been theoretically proved, and scenarios where the first iteration is infeasible limits this technique. However, the satisfaction of non-convex constraints with convex optimization is desirable and promising, and thus this technique may warrant further research.

Another contribution of this work is the use of the Ankersen zero-order hold particular solution [17], which, to the best of the author's knowledge, has not been used in an MPC for rendezvous context. Typically, a simpler impulsive discretization is used instead, while a constant thrust parametrization more realistically models the spacecraft thrust profile. Because the sampling intervals are very large, however, it may be undesirable to command constant thrust actions for such long periods of time. Nonetheless, this may be overcome by easily defining, in the particular solution, the thrust duration to be less than the sampling period, in what would become a partial zero-order hold discretization. Moreover, many spacecraft, including the PROBA-3 Occulter Spacecraft, only have non-throttleable on/off thrusters, and thus perform intermediate thrust via PWM. An even more realistic model is then to use the PWM parameters as decision variables, although this results in a non-linear prediction model. In [34] the authors Vazquez et al. achieve this via iterative linear programming.

Another crucial aspect is robustness in the presence of disturbances, of which the most significant in a rendezvous scenario are modelling errors, navigation uncertainty, and execution errors. Although MPC can inherently achieve robust convergence [2], due to the fact that it is a closed-loop strategy, additional techniques might be required to achieve robust feasibility, performance, and constraint satisfaction. To ensure that the terminal constraint present in the FH-MPC formulation never renders the optimization problem infeasible, we use the feasible terminal box approach [51], which relaxes the terminal constraint to the minimum size that allows the problem to be feasible, while maintaining the problem as an LP and not significantly increasing the computational complexity. Robust performance, which here refers to robust fuel-performance and manoeuvre accuracy, is often not addressed in the literature. For avoiding over-correction of the trajectory due to stochastic disturbances, we proposed a dynamically variable terminal box constraint that substitutes the equality terminal state constraint, showing better fuel-performance and maintaining accuracy. This, however, requires significant tuning and can benefit from further research, for example, to compute the size of the terminal box as a function of the uncertainty. We also proposed the use of a quadratic terminal controller, which substitutes the final iteration of the FH-MPC controller, allowing for a more precise braking manoeuvre and granting better terminal accuracy. Neither of these proposed techniques increases the computational complexity of the problem. Finally, although robust constraint satisfaction techniques were not considered here, the need for these techniques was shown for the passive safety constraint. The chance-constrained MPC approach presented in [32] is a good candidate, since it employs constraint tightening with online uncertainty estimation, and does not significantly increase computational complexity.

Although MPC has the potential to be used for guidance and control of real rendezvous missions, its feasible implementation depends on the specific hardware for that mission. Furthermore, while MPC can offer increased autonomy and better fuel consumption profiles, this must be balanced with its greatest downside, which is the required computational complexity. Also, before it becomes an attractive alternative, there is a need to analyse weather MPC indeed offers fuel benefits over the traditional approach, taking into account all disturbances present in a rendezvous mission, which can greatly reduce its performance. This ties into the need for a standard approach for granting robustness that remains feasible to implement in real-time, which does not yet exist and thus calls for further research.

Finally, we summarize future work suggested along this thesis:

- Study convergence of the OAILP algorithm to a local minimum of the original nonlinear problem, and improve it to consider cases where the linear constraints yield an infeasible problem.
- Test the passive safety constraint, with OAONP or OAILP, in the presence of disturbances, and apply robust constraint satisfaction techniques.
- Extend passive safety constraint to include mid-thrust failures.
- Improve upon the methods here presented here for robust performance, namely the dynamic terminal box, by directly taking account the uncertainty, and the terminal quadratic controller.
- Compare the MPC algorithms developed with the conventional approaches in a realistic simulator and evaluate if it indeed offers fuel benefits.
- Implement the MPC algorithm in an embedded environment comparable to flight hardware (*e.g.* [52]).

Bibliography

[1] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge University Press, 2003. ISBN 0521824923.

[2] J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition, 2017.

[3] P. Bodin, R. Noteborn, R. Larsson, and C. Chasset. System test results from the GNC experiments on the PRISMA in-orbit test bed. *Acta Astronautica*, 68(7):862 – 872, 2011. ISSN 0094-5765. doi: 10.1016/j.actaastro.2010.08.021.

[4] P. Rosa, P. F. Silva, B. Parreira, M. Hagenfeldt, A. Fabrizi, A. Pagano, A. Russo, S. Salvi, M. Kerr, S. Radu, et al. Autonomous close-proximity operations in space: The PROBA-3 rendezvous experiment (P3RVX). In *69th International Astronautical Congress (IAC 2018)*. International Astronautical Federation, 2018.

[5] J. Han, Y. Hu, and S. Dian. The state-of-the-art of model predictive control in recent years. *IOP Conference Series: Materials Science and Engineering*, 428:012035, October 2018. doi: 10.1088/1757-899X/428/1/012035.

[6] E. N. Hartley. A tutorial on model predictive control for spacecraft rendezvous. pages 1355–1361, July 2015. doi: 10.1109/ECC.2015.7330727.

[7] C. V. Rao, S. J. Wright, and J. B. Rawlings. Application of interior-point methods to model predictive control. *Journal of optimization theory and applications*, 99(3):723–757, 1998.

[8] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 668–674. IEEE, 2012.

[9] D. Liao-McPherson, M. Huang, and I. Kolmanovsky. A regularized and smoothed Fischer-Burmeister method for quadratic programming with applications to model predictive control. *IEEE Transactions on Automatic Control*, 2018.

[10] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[11] G. W. Hill. Researches in the lunar theory. *American Journal of Mathematics*, 1(1):5–26, 1878. ISSN 00029327, 10806377.

[12] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960. doi: 10.2514/8.8704.

[13] J. P. De Vries. Elliptic elements in terms of small increments of position and velocity components. *AIAA Journal*, 1(11):2626–2629, 1963.

[14] J. Tschauner and P. Hempel. Rendezvous with a target in an elliptical orbit. *Astronautica Acta*, 11 (2):104–109, 1965.

[15] T. E. Carter. State transition matrices for terminal rendezvous studies: brief survey and new example. *Journal of Guidance, Control, and Dynamics*, 21(1):148–155, 1998.

[16] 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, 2002. doi: 10.2514/2.4875.

[17] F. Ankersen. *Guidance, Navigation, Control and Relative Dynamics for Spacecraft Proximity Maneuvers: Ph.D. Thesis*. Institut for Elektroniske Systemer, 2010. ISBN 9788792328724.

[18] L. Breger and J. P. How. Gauss's variational equation-based dynamics and control for formation flying spacecraft. *Journal of guidance, control, and dynamics*, 30(2):437–448, 2007.

[19] A. Richards and J. How. Performance evaluation of rendezvous using model predictive control. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, November 2003. doi: 10.2514/6.2003-5507.

[20] G. F. Franklin, J. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 2006.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[22] D. Ruscio. Model predictive control with integral action: A simple MPC algorithm. *Modeling, Identification and Control: A Norwegian Research Bulletin*, 34:119–129, January 2013. doi: 10.4173/mic.2013.3.2.

[23] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. January 2002.

[24] J. B. Rawlings. Tutorial: model predictive control technology. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 1, pages 662–676 vol.1, June 1999. doi: 10.1109/ACC.1999.782911.

[25] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on control systems technology*, 18(2):267–278, 2009.

[26] G. Deaconu, C. Louembet, and A. Théron. Designing continuously constrained spacecraft relative trajectories for proximity operations. *Journal of Guidance, Control, and Dynamics*, 38(7):1208–1217, 2014.

[27] K. Kanjanawaniskul. Motion control of a wheeled mobile robot using model predictive control: A survey. *Asia-Pacific Journal of Science and Technology*, 17(5):811–837, 2012.

[28] D. F. Lawden. Fundamentals of space navigation. *Journal of the British Interplanetary Society*, 13 (2):87–101, 1954.

[29] S. Di Cairano, H. Park, and I. Kolmanovsky. Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, 22(12):1398–1427, 2012.

[30] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski. Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice*, 20(7):695–713, 2012.

[31] L. Breger, J. How, and A. Richards. Model predictive control of spacecraft formations with sensing noise. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2385–2390. IEEE, 2005.

[32] F. Gavilan, R. Vazquez, and E. F. Camacho. Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation. *Control Engineering Practice*, 20(2):111–122, 2012.

[33] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky. Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. *IEEE Transactions on Control Systems Technology*, 23(4):1638–1647, 2015.

[34] R. Vazquez, F. Gavilan, and E. F. Camacho. Model predictive control for spacecraft rendezvous in elliptical orbits with on/off thrusters. *IFAC-PapersOnLine*, 48(9):251–256, 2015.

[35] S. Zhu, R. Sun, J. Wang, J. Wang, and X. Shao. Robust model predictive control for multi-step short range spacecraft rendezvous. *Advances in Space Research*, 62(1):111–126, 2018.

[36] y. Hegrenæs, J. Gravdahl, and P. Tøndel. Spacecraft attitude control using explicit model predictive control. *Automatica*, 41:2107–2114, December 2005. doi: 10.1016/j.automatica.2005.06.015.

[37] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra, and F. Y. Hadaegh. A model predictive control-based approach for spacecraft formation keeping and attitude control. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 6, pages 4258–4262 vol.6, June 1999. doi: 10.1109/ACC.1999.786367.

[38] J. Lilja. Combined attitude and orbital MPC for thruster based spacecrafts, 2017.

[39] A. Walsh, S. Di Cairano, and A. Weiss. MPC for coupled station keeping, attitude control, and momentum management of low-thrust geostationary satellites. pages 7408–7413, July 2016. doi: 10.1109/ACC.2016.7526842.

[40] P. Lu and X. Liu. Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, 36(2):375–389, 2013.

[41] A. Richards and J. How. Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. volume 5, pages 4034 – 4040 vol.5, July 2003. ISBN 0-7803-7896-2. doi: 10.1109/ACC.2003.1240467.

[42] A. Richards, E. Feron, J. How, and T. Schouwenaars. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance Control and Dynamics*, 25, July 2002. doi: 10.2514/2.4943.

[43] L. S. Breger and J. P. How. Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1478–1489, 2008.

[44] J. B. Mueller and R. Larsson. Collision avoidance maneuver planning with robust optimization. In *7th International ESA Conference on Guidance, Navigation and Control Systems, Tralee, County Kerry, Ireland*, June 2008.

[45] P. O. Scokaert and D. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic control*, 43(8):1136–1142, 1998.

[46] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[47] J. P. How and M. Tillerson. Analysis of the impact of sensor noise on formation flying control. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 5, pages 3986–3991. IEEE, 2001.

[48] G. Deaconu, C. Louembet, and A. Théron. Minimizing the effects of navigation uncertainties on the spacecraft rendezvous precision. *Journal of Guidance, Control, and Dynamics*, 37(2):695–700, 2014.

[49] C. Louembet, D. Arzelier, and G. Deaconu. Robust rendezvous planning under maneuver execution errors. *Journal of Guidance, Control, and Dynamics*, 38(1):76–93, 2014.

[50] A. G. Richards. *Robust constrained model predictive control*. PhD thesis, Massachusetts Institute of Technology, 2005.

[51] M. Tillerson, G. Inalhan, and J. P. How. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(2-3):207–242, 2002.

[52] E. N. Hartley and J. M. Maciejowski. Predictive control for spacecraft rendezvous in an elliptical orbit using an fpga. In *2013 European Control Conference (ECC)*, pages 1359–1364. IEEE, 2013.

Appendix A

Particular Solution Integrals

In this appendix, the expressions for the integrals in equations (3.43) and (3.44) are presented, as solved in [17].

$$I_{s_3} = \frac{1}{2e}\left[\frac{1}{\rho(\theta_t)^2} - \frac{1}{\rho(\theta_0)^2}\right]. \tag{A.1}$$

$$I_{c_3} = (1-e^2)^{-\frac{5}{2}}\left[(1+e^2)(\sin(E_t)-\sin(E_0)) - \frac{e}{2}(\sin(E_t)\cos(E_t)-\sin(E_0)\cos(E_0)+3(E_t-E_0))\right], \tag{A.2}$$

$$I_{s_2} = \frac{1}{e}\left[\frac{1}{\rho(\theta_t)} - \frac{1}{\rho(\theta_0)}\right] \tag{A.3}$$

$$I_{c_2} = (1-e^2)^{-\frac{3}{2}}\left[\sin(E_t)-\sin(E_0)-e(E_t-E_0)\right] \tag{A.4}$$

$$\begin{gathered} I_3 = (1-e^2)^{-\frac{5}{2}}\left[(\frac{1}{2}e^2+1)(E_t-E_0)+\right. \\ \left.+\frac{1}{2}e^2(\sin(E_t)\cos(E_t)-\sin(E_0)\cos(E_0))-2e(\sin(E_t)-\sin(E_0)\right] \end{gathered} \tag{A.5}$$

$$I_1 = (1-e^2)^{-\frac{1}{2}}(E_t-E_0) \tag{A.6}$$

$$I_{1J} = (1-e^2)^{-2}\left[\frac{1}{2}(E_t^2-E_0^2)+e(\cos(E_t)-\cos(E_0))+(e\sin(E_0)-E_0)(E_t-E_0)\right] \tag{A.7}$$

$$\begin{gathered} I_{s_2J} = (1-e^2)^{-\frac{5}{2}}\left[\sin(E_t)(1+\frac{e}{2}\cos(E_t))-E_t(\frac{e}{2}+\cos(E_t))-\right. \\ \left.\sin(E_0)(1+\frac{e}{2}\cos(E_0))+E_0(\frac{e}{2}+\cos(E_0))-(e\sin(E_0)-E_0)(\cos(E_t)-\cos(E_0))\right] \end{gathered} \tag{A.8}$$