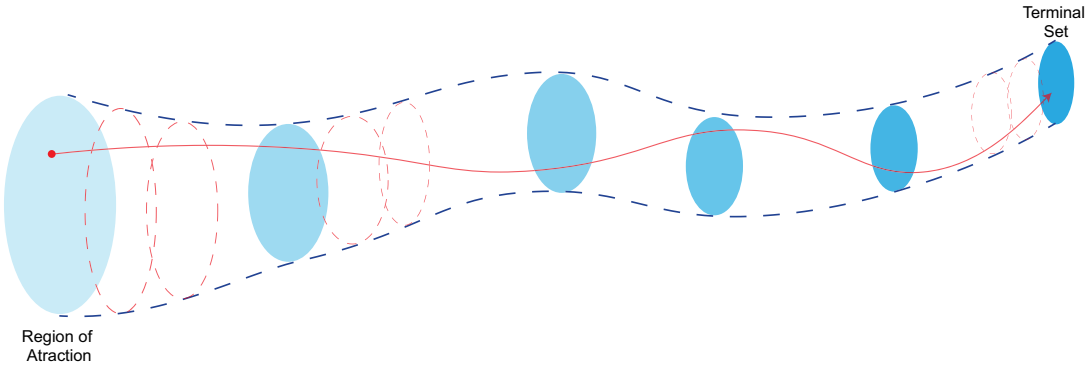
TÉCNICO LISBOA



Stability Guarantees for Model Predictive Controllers

Rita Nogueira Palma

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Doutor Daniel de Matos Silvestre
Prof. Doutor Rita Maria Mendes de Almeida Correia da Cunha

Examination Committee

Chairperson: Prof. Doutor Teresa Maria Canavarro Menéres Mendes de Almeida
Supervisor: Prof. Doutor Rita Maria Mendes de Almeida Correia da Cunha
Member of the Committee: Prof. Doutor Bruno João Nogueira Guerreiro

October 2023

Declaration

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

Education is the most powerful weapon you can use to change the world.

- Nelson Mandela.

Acknowledgments

First and foremost, I want to express my appreciation to my professors. The extensive meetings with Professor Daniel Silvestre over the past year have significantly influenced my growth. Your patient mentorship has deepened my understanding of this intricate work and inspired higher aspirations. Equally appreciated is Professor Rita Cunha, whose guidance and expertise have been invaluable. Additionally, I express my gratitude to Professor Francisco Rego for his unwavering support throughout this journey.

A dissertation is the culmination of years of meticulous labour, and my knowledge has expanded significantly. I am grateful to all my professors for kindling my curiosity and fostering an enriching atmosphere at Instituto Superior Técnico. Each of you has left a mark on my academic journey, and I extend heartfelt thanks for your guidance and inspiration.

To my cherished friends who have been by my side since day one. My university experience would be incomplete without you. You have been friends but also role models, consistently motivating me to strive for excellence. Through countless nights of work, your optimism has motivated me to strive for my potential. We have taken on difficult challenges, had interesting discussions, and celebrated each accomplishment as one. You have enriched my academic journey in ways I could never imagine. I look forward to our future adventures. Here's to Xagueira and the cherished times we have shared. Especially to David Gomes, who has greatly inspired me during my university path and still does nowadays.

I could not forget to mention JUNITEC, a place that became my second home throughout my university years and holds a special place in my heart. I thank everyone who came into my life, brightening my path and making me a better person.

My final acknowledgements go to my parents and brother, whose unwavering love, endless support, and constant encouragement have been my guiding lights throughout this journey. Your sacrifices and belief in me have shaped not only my academic pursuits but also my character. Thank you for always being my pillars of strength.

Resumo

Os recentes avanços tecnológicos e as suas aplicações em diferentes áreas em conjunto com o crescente interesse em veículos autónomos marcam o início de uma nova era para estes últimos. O domínio aeroespacial abrange diferentes veículos como naves espaciais para o lançamento de satélites, lançamento de foguetões e veículos aéreos não tripulados com os mais diferentes propósitos como é o caso de vigilância e transporte. Cada uma destas áreas exige estratégias de controlo cada vez mais robustas para garantir que os veículos interagem de uma forma eficaz com ambientes externos e são capazes de executar manobras complexas. Contrariamente, os métodos de controlo tradicionais já não são capazes de satisfazerem de uma forma viável os atuais requisitos, o que impossibilita os sistemas em questão de alcançar um desempenho ótimo.

Motivada por estes desafios, a abordagem proposta implementa um Modelo de Controlo Preditivo com restrições específicas, que obrigam o sistema a entrar num conjunto invariante representado como um Gerador Convexo Restrito, reduzindo o conservadorismo associado. Desta forma, o controlador garante a viabilidade do mesmo ainda que na presença de perturbações. Além disso, é possível obter uma descrição exata da região de atração para que o sistema alcance o conjunto terminal. Os resultados da simulação validam o desempenho das trajetórias do Modelo de Controlo Preditivo, enquanto confirmam que os tempos de computação são adequados para uma aplicação em tempo real.

Palavras-chave: Modelo de Controlo Preditivo (MPC), Gerador Convexo Restrito (CCG), Região de Atração, Robusto Positivamente Invariante (RPI), Estabilidade, Viabilidade Recursiva

Abstract

The technological advances and their application across various sectors together with the growing interest in autonomous vehicles mark the beginning of a new era for the latter. The aerospace domain encompasses different vehicles such as spacecrafts for satellite deployment, rocket launch, and UAV with purposes such as surveillance and transportation. Each of these areas demands more robust control strategies to ensure that the vehicles can interact efficiently with external environments and can execute complex manoeuvres. Contrarily, traditional control methods are no longer able to viably meet the evolving requirements, which limits the systems from achieving an optimal operational performance.

Motivated by these challenges, the proposed approach implements a MPC with specific constraints to force the system to enter into an invariant set represented as a CCG, which reduces the related conservatism. In doing so, the controller ensures its feasibility even with the presence of disturbances. Moreover, it is also possible to obtain a description of the region of attraction for the system to reach the terminal set. Simulation results validate the performance of MPC trajectories whilst confirming that the computational times are suitable for a real-time application.

Keywords: Model Predictive Control (MPC), Constrained Convex Generators (CCGs), Region of Attraction, Robust Positively Invariant (RPI), Stability, Recursive Feasibility

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Context and Motivation

Over the past decade, research in autonomous vehicles has expanded significantly due to technological advancements and the rising demand for innovative solutions across various domains. In space, the complexity of missions, satellite deployment, and exploration requires the development of more stable and robust control strategies [1]. On Earth, multirotor Unmanned Aerial Vehicle (UAV)s are reshaping industries through applications such as remote surveillance, photography, inspection, search, transport, and rescue operations [2]. These applications showcase the versatility of aerospace autonomous vehicles.

The investigation of these control systems continues unabated, driven by the demand for robust strategies where vehicles interact with external environments and execute complex manoeuvres to fulfil specific missions or navigate around obstacles. As a result, there is a growing need to ensure the feasibility and success of each endeavour.

The global space economy continues to expand, reaching $469 billion in 2021 according to *The Space Report 2022 Q2* [3], representing an increase of 9% considering the previous year. This growth highlights the space market's exponential expansion, with 72 rockets launching 1022 identified spacecrafts into orbit from January 1 to June 30, 2022. Consequently, more launches have occurred in these six months than in the first 52 years of the Space Age. These statistics demonstrate the industry's drive to innovate and the vision for the upcoming space era.

While control theories have evolved significantly, traditional Proportional-Integral-Derivative (PID) control remains unmistakably dominant in aerospace. Since 1957, it has served as the baseline controller for over 99% of the spacecraft launched [4]. Demonstrating that PID can satisfy the majority of the fundamental mission requirements [5]. However, the new era of spacecrafts should exhibit unique characteristics to meet ultimate high-level performance conditions such as high flexibility and high-frequency oscillations. According to the data provided by Hanson, from 1991 to 2001, 41% of all the failures regarding space launches could have been avoided by deploying advanced guidance and control methods. Thus, the traditional PID controller is gradually becoming unable to satisfy the increasing demands in

this area [6]. Other techniques include adaptive control [7] developed in the 1960s and robust control [8] created in the 1970s.

Quadrotors describe a specific category of UAV, distinguished by their exceptional versatility due to notable characteristics. It excels at manoeuvring through tight spaces, maintaining stable hover during flight, and offering a lightweight and cost-effective solution [9]. Therefore, the use of quadrotors has led to the emergence of various control methods aimed at enabling attitude stabilisation and trajectory tracking for these aerial vehicles.

Kamel et al. defends the importance of precise trajectory tracking for aerial robots, particularly in real-world environments where external disturbances can significantly impact flight performance. However, ensuring global stability typically requires addressing the problem of introducing both a terminal cost and a terminal constraint for the states. This task can be challenging, and simply adding a terminal constraint may not be feasible. It highlights that in many practical situations, the terminal constraint is not enforced during the control design procedure but rather verified *a posteriori* (i.e., by increasing the prediction horizon if not satisfied).

In an era marked by the desire for more efficient and cost-effective parcel delivery, small-scale unmanned aircraft have emerged, ranging from fixed-wing to multirotors, tilt-rotors, and tilt-wings. The REPLACE project [11], driven by this ambition, focuses on harnessing shuttle drones for launch and capture manoeuvres involving vehicles and objects. It encompasses two primary scenarios: cooperation between shuttle drones and the vehicles they launch or capture and dealing with the launch and capture of non-cooperative objects or drones. Although there are no assurances of feasibility before commencing the mission, this lack of certainty can lead to accidents and compromise the successful execution of intricate manoeuvres. Introducing a risk to the smooth interaction of parcel exchange.

The research by Eren et al. emphasises the potential of Model Predictive Control (MPC) algorithms to shape control strategies and enhance the development of autonomous systems. The evolving requirements raised the bar for the next generation of these vehicles to increase robustness, adaptability, and performance. These advanced UAVs must exhibit improved mobility, motion accuracy, and energy efficiency compared to traditional ones [2]. Islam et al. has conducted extensive studies exploring various control methods to UAVs across diverse applications, underscoring the garnered attention in MPC concerning flight control.

According to [12], the success of MPC is due to its ability to simultaneously handle constraints (e.g., design specifications) and optimise performance via repetitive online optimisation. However, because MPC operates as a feedback loop, conducting stability analysis becomes crucial for ensuring the system's robustness and performance. This methodology allows the optimisation methods to directly impose system constraints on robust controlled invariant sets [14]. In contrast, conventional methods often exhibit conservatism in describing these sets, restricting the controller from achieving optimal operational performance. This mismatch becomes more evident in the aerial domain because it is notoriously difficult to define due to its abundance of uncertainties and disturbances.

This dissertation addresses the challenge of ensuring feasibility and robustness in MPC for autonomous vehicles. As the demand for autonomous vehicle applications increases across various sec-

tors, from space exploration to parcel delivery, the existing control methods may fall short of meeting the evolving requirements.

Motivated by these observations, the envisaged solution guarantees the mission feasibility *a priori*. Adjusting the optimisation problem and representing the robust controller invariant sets with minimal conservatism through Constrained Convex Generator (CCG) formulation [15]. This definition adapts Constrained Zonotope (CZ) [16] for polytopes to ensure generators belong to a convex set. It allows the system to operate at its limits whilst preventing infeasibility. Such infeasibility poses a significant concern as it could compromise the integrity of the mission, enhancing the safety and reliability of the MPC control system. This approach ensures that domains requiring complex manoeuvres for autonomous vehicles can benefit from guaranteed feasibility before embarking on missions.

1.2 Contributions

The main contributions are summarised as follows:

- Development of a method to backpropagate the minimal Robust Positively Invariant (mRPI) to determine the region of attraction.
- Design of a robust predictive controller strategy that guarantees MPC feasibility *a priori* and adapts constraints for missions with duration exceeding the prediction horizon.
- Conduct a performance evaluation that compares set descriptions using both CZ and CCG formulations.

1.3 Outline

This document is structured into six chapters, including the present one. Chapter 1 provides an overview of the motivation and objectives of the thesis. Chapter 2 conducts a literature review concerning the application of MPC in aerospace systems, trajectory generation, aerodynamics modelling, and the importance of stability and robustness to the controller's algorithms. Chapter 3 delves into the problem formulation by presenting the theoretical principles that underpin this research as the MPC and the stability proof for this controller, the CCGs formulation and the definition for the vehicle model, it finishes with the problem statement. The subsequent chapters address the core research aspects. Chapter 4 outlines the methodologies for ensuring feasible trajectories, encompassing the reachability analysis related to CCGs and the implementation of the MPC. Chapter 5 focuses on the results and discusses the approach employed by comparing the results with other methods. Finally, Chapter 6 concludes the dissertation by summarising the contributions, discussing the implications of the results, and suggesting future research directions.

Chapter 2

Literature Review

This chapter provides an overview of the latest project researching advanced strategies for aerospace control systems, covering trajectory generation, aerodynamics modelling, stability, and robustness.

2.1 MPC in Aerospace Systems

To achieve accuracy and efficiency in aerospace systems, Eren et al. emphasises the need for robust control strategies. It considers MPC a promising solution due to promoting an effective management of aerospace vehicles across diverse contexts. These encompass scenarios such as spacecraft *rendezvous*, autonomous drone flight, and trajectory optimisation for aircraft and rockets.

In MPC formulation, it is possible to create different architectures by modifying cost and constraint functions. This formulation considers the states' environmental details (e.g., stall angle of attack, velocity limits) and inputs (e.g., maximum deflections of control surfaces, thrust limits, minimum impulse bit, and thruster on/off time). It accounts for complex nonlinearities and couplings (e.g., aircraft dynamics, control structure interactions, and mission-specific modes) that exhibit powerful dynamics in aerospace systems.

[17] considers MPC schemes to control the spacecraft during the final phase of the *rendezvous* manoeuvre to satisfy the mission constraints. With a focus on the robustness and computational efficiency of a classical MPC, the study introduces a new approach called Tube-based Model Predictive Control (TMPC). To minimise a particular cost function (e.g., fuel consumption or deviation from the reference trajectory), the design of the controller tightens the mission constraints (i.e., states and control inputs). It allows the trajectories of the uncertain system (e.g., affected by disturbance) to lie in a tube centred on the nominal one, where each path associates with a specific realisation of the uncertainty at each discrete-time step. Furthermore, this algorithm splits into two components: offline constraint evaluation and online application to nominal trajectories representing the centre of the tube itself.

Various analyses have been developed to assess obstacle avoidance within the aerospace domain. An approach related to linear optimisation is discussed in [18], which tackles this topic by substituting the obstacle constraint with a convex set that effectively avoids the obstacle but remains fixed throughout

the prediction. A convex collision avoidance formulation provides computationally fast solutions at a small fuel expense. However, this approach often confines the trajectory within a significantly more conservative region, leading to infeasibility.

The investigation by [19] in collaboration with the project REPLACE [11] focuses on planning optimal relay manoeuvres using multirotors to enhance parcel exchanges during flight. The strategy generates MPC optimal polynomial trajectories with acceleration constraints to ensure effective attitude coordination between the vehicles during package exchange.

Therefore, the translation dynamics of the multirotor are modelled as a triple integrator, which allows for the derivation of control input sequences that adhere to specific limits on thrust and angular velocity. The results reveal that the aircraft demonstrates commendable tracking performance, successfully enabling the mid-air exchange of parcels between the two vehicles. However, the aircraft exhibits relatively high roll and pitch rates. Despite the planned trajectories being optimal for minimising the required input body torques, it is desirable to introduce bound constraints on position derivatives. This addition ensures the management of saturation and rate limits in trajectory generation, although these additional inequality constraints may lead to numerical issues.

2.2 Trajectory Generation

Reliable trajectory generation methods are critical to safeguarding public trust and the high safety standards expected of autonomous systems. A trajectory generation algorithm is a computationally efficient algorithm that generates a discrete multidimensional temporal state and control signal that meets specifications whilst optimising key mission objectives (e.g., avoiding certain obstacles). A convex optimisation-based trajectory generation provides a systematic method to deal with nonconvexities and generate a trajectory based on a convex solver.

As the mission progresses and additional information becomes available to the autonomous vehicle, the trajectory undergoes real-time adjustments and updates. The repetitive trajectories generate feedback that is employed to control the system. MPC has been applied for simultaneous trajectory generation and control as it tracks trajectories with a separate feedback controller [20]. At each discrete-time step, MPC can approximate a nonconvex trajectory generation problem as a convex quadratic optimisation.

The integration of MPC with Linear-Quadratic Regulator (LQR) offers a promising approach, as defended by Mousavi et al.. Dealing with the challenge of manoeuvring through dynamic environments while ensuring stochastic target tracking introduces probabilistic constraints. In such scenarios, the complexity of the optimisation-based path planning problem increases. Consequently, the method takes on a nonconvex nature, introducing complexities related to solvability and computational demands. A notable innovation lies in reinterpreting the dimensions of the vehicle and obstacles. By approximating sharp corners or complex forms with circles, the strategy promotes the generation of smoother and more efficient vehicle trajectories. Thus, the search space is linearised at discrete-time intervals. It results in a series of linear constraints enforced at each instant (formerly nonlinear probabilistic constraints), en-

abling robust obstacle avoidance. This reinterpretation transforms optimisation into a convex problem, yielding more efficient trajectories and robust navigation.

The baseline trajectory incorporates considerations for various disturbances. UAVs may encounter variations in weather conditions (e.g., wind and rain), obstacles obstructing their path, and potential sensor malfunctions. The spacecraft disturbances (e.g., gravitational fields of multiple celestial objects, solar radiation pressure, third-body effects, and atmospheric drag) are all aimed at designing a robust controller. A practical trajectory provides optimal support for payload objectives, navigation, power generation, propulsion, communication, and data transfer, among other subsystem considerations.

2.3 Aerodynamics Modelling

Precisely representing aerodynamic effects holds the utmost significance in controllers for aerospace systems. Aerodynamics are pivotal in determining the behaviour, trajectory, and control to ensure mission success and safety. The majority of the complexity of aerodynamic modelling is overlooked due to several minor details of aerospace engineering that are beyond the scope of this thesis.

In autonomous systems, the study of aerodynamic modelling often entails simplification, with a primary focus on steady-state thrust and reaction torque within unobstructed space [19]. Nonetheless, this neglects secondary aerodynamic influences that appear beyond controlled scenarios.

The work presented in [22] investigates various aerodynamics modelling techniques. It aims to incorporate and discuss different models focusing on linear systems. This research covers a variety of elements, such as the effect of perturbations and atmospheric drag forces. Notably, particular models produce linear differential equations for the relative position, velocity, and acceleration; however, the linearity does not extend to the angular velocity. Consequently, this dynamic results in a time-variant nature of relative motion.

2.4 Stability and Robustness

Ensuring the control problem's feasibility is crucial to providing safe and reliable control performance. One potential disadvantage of the MPC is the inability to guarantee the feasibility of specific optimisation difficulties, as stated in [23]. In real-time applications, it is frequently preferable to anticipate system states using a nominal, linearised model for computational simplicity. Several researchers have conducted substantial studies on the problem of recursive feasibility in the context of MPC.

As analysed in [24], achieving recursive feasibility may be possible by constraining the states to a pre-computed robust controlled invariant set. Establishing the importance of computing sets that guarantee the robustness of the controller. Although, usually, these robust controlled invariant sets can have a complicated geometry. As a result, limiting the states in these sets imposes too many state restrictions, making solving the online optimisation issue computationally costly. In real-time MPC, computing resources must be allocated to account for worst-case complexity, needing extensive resources for global optimality. However, the high computing load can make real-time applications impracticable.

Hanema et al. incorporates finite-step terminal conditions within the context of TMPC applied to Linear Parameter-Varying (LPV) systems. The study demonstrates the property of recursive feasibility under certain assumptions about tube parameterisation. The primary goal of building such tubes in TMPC is to ensure that perturbed system trajectories are close to a nominal trajectory. These tubes consist of translated and scaled examples from the fundamental shape set. Furthermore, analysing a more robust MPC framework ensures the feasibility of the nominal trajectory by default.

In the study by Limon et al., occurs the substitution of the terminal condition with a contractive constraint from a sequence of reachable sets. These sets represent a progression, not necessarily invariant, wherein the system is admissible to steer from one set to the subsequent one, ultimately converging to the target set. It emphasises the cost-effectiveness of computing the sequence through offline computation starting from a positively invariant set (i.e., employing an inner approximation to the terminal set). Combining the MPC controller with this sequence by including the terminal constraint guarantees the placement of the terminal state within the subsequent computed reachable set.

However, the lack of assurance invariance represents a drawback of the approach by Limon et al.. This ambiguity in achieving invariance arises because the sequence composition might not include the previous set due to the inherent approximations made. Ultimately, the controller attains asymptotic stability and local optimality for the Closed-Loop (CL) system. This research enables the development of computationally less demanding algorithms for determining a sequence of invariant or merely reachable sets.

Chapter 3

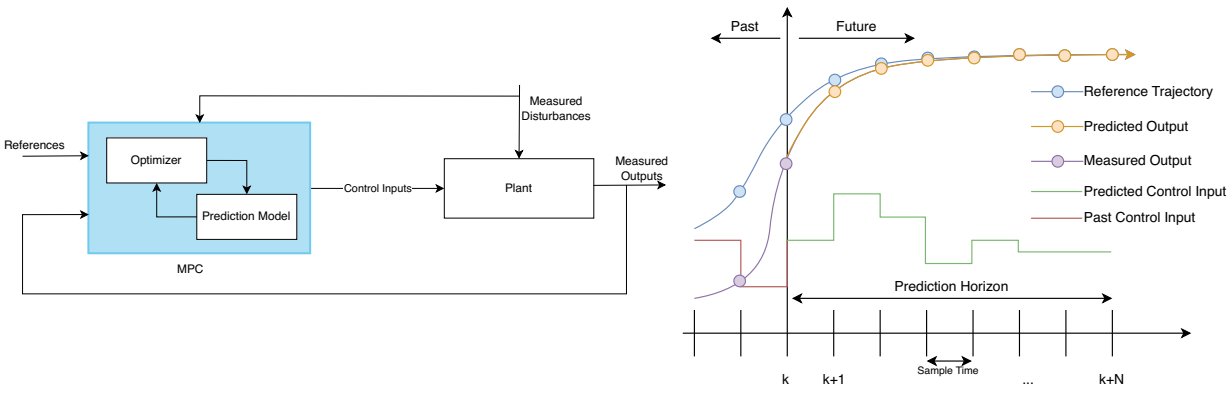
Problem Formulation

This chapter introduces the required background on MPC formulation, reachability analysis using CCGs, and vehicle modelling and then presents the problem statement.

3.1 Model Predictive Control

MPC is an iterative feedback control technique with roots in optimal control. MPC is evolving as an approach for state feedback stabilisation in constrained, nonlinear, discrete-time, and time-invariant systems [12]. Its state-of-the-art design methodologies are documented in [27], [28], [29], and [30].

In contrast to conventional control, which frequently relies on a pre-computed state or output feedback control law, predictive control employs a discrete-time model to obtain an estimate (i.e., prediction) of its future behaviour. The optimisation problem uses a prediction over a finite horizon N for the state values that adhere to the dynamics. The feedback control law is then obtained in a receding horizon by applying only the first element of the computed sequence of optimal controls to the system and repeating the whole process at the next discrete-time instant for the updated state. The feedback control repeatedly runs the CL to correct discrepancies between actual and previously assumed optimal states. Figure 3.1a depicts a MPC-based control loop and the MPC principle is present in Figure 3.1b.



(a) (b)

Figure 3.1: Illustrations of the practical MPC formulation (adapted from [31]). (a) Simplified block diagram of a MPC-based control loop. (b) The function principle of a MPC.

Assuming that the prediction model in discrete-time is given by

$$\boldsymbol{x}(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right), \ \forall k \geq 0, \tag{3.1}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$ is the state vector at instant k; $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ is the control input vector at instant k; and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ calculates the next state $\boldsymbol{x}(k+1) \in \mathbb{R}^{n_x}$. If f is a linear function, the system is characterised by the linear state model

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k), \ \forall k \geq 0, \tag{3.2}$$

where $A \in \mathbb{R}^{n_x \times n_x}$ represents the state-transition matrix and $B \in \mathbb{R}^{n_x \times n_u}$ the control input matrix.

Constraints may also be imposed on the measured state $\boldsymbol{x}(k) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ and measured control input $\boldsymbol{u}(k) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$, subject to $(\boldsymbol{x}(k), \boldsymbol{u}(k)) \in \mathbb{G}$. The state-control constraint set $\mathbb{G} \subseteq \mathbb{X} \times \mathbb{U}$, derived from the Cartesian product, represents all constraints resulting from factors such as physical limitations, safety requirements, and desired operation modes. Eren et al. defends that the success of MPC algorithms in control systems may be attributed, in part, to its intuitive approach to the control problem and capability to incorporate state-dependent control constraints.

Quality of performance associated with the pair of state and control variables $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ is measured via values of a stage cost function $\ell(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ evaluated at $(\boldsymbol{x}(k), \boldsymbol{u}(k)) : \ell(\boldsymbol{x}(k), \boldsymbol{u}(k))$. Therefore, this cost penalises the state and input predicted trajectories up to (but excluding) the terminal condition. To drive the system's state towards the reference, it is critical to penalise the state at the end of the prediction horizon by considering the terminal cost $V_f(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^+$. Therefore, the MPC methodology is defined as the following constrained optimal problem, according to Appendix A:

$$\min_{\boldsymbol{u}(0),\ldots,\boldsymbol{u}(N-1)} \ J\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right) := \sum_{k=0}^{N-1} \ell(\boldsymbol{x}(k), \boldsymbol{u}(k)) + V_f(\boldsymbol{x}(N)) \tag{3.3a}$$

$$\text{s.t. } \boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k)), \text{ for } k = 0, 1, \ldots, N-1, \tag{3.3b}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{X}, \tag{3.3c}$$

$$(\boldsymbol{x}(k), \boldsymbol{u}(k)) \in \mathbb{G}, \text{ for } k = 0, 1, \ldots, N-1, \tag{3.3d}$$

$$\boldsymbol{x}(N) \in \mathbb{X}_f \subseteq \mathbb{X}, \tag{3.3e}$$

where (3.3b) translates the system model; (3.3c) sets the first element of the trajectory as the initial state; (3.3d) certifies that the pair of state and control variables is contained in the constraint set for all discrete-time steps; (3.3e) guarantees that the terminal condition belongs to the terminal set $\mathbb{X}_f$, representing the target region the system should reach at the end of the prediction horizon.

The value of the objective function (3.3a) represents the cost-to-go from instant 0 to N under the control inputs. A common choice in MPC is selecting a quadratic cost function because of its numerous advantages, like being convex, smooth and easy to tune [12]. The quadratic cost function can be

expressed as

$$J\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right) = \boldsymbol{x}(N)^\top P \boldsymbol{x}(N) + \sum_{i=0}^{N-1} \left(\boldsymbol{x}(i)^\top Q \boldsymbol{x}(i) + \boldsymbol{u}(i)^\top R \boldsymbol{u}(i) \right). \tag{3.4}$$

The tuning parameters are the matrices $Q \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$, chosen based on performance specifications. Large values of Q compared to R indicate a higher priority to quickly drive the state towards the origin, which requires a more expensive control action. Although punishing the control action with a higher R relative to Q is one way to reduce the control action and slow the rate at which the state approaches the origin. One of the MPC challenges is the tuning of the matrices Q and R. The final state penalty has a different weighting matrix, denoted as $P \in \mathbb{R}^{n_x \times n_x}$, calculated for exact Q and R values to guarantee recursive feasibility and stability.

The ℓ_1-norm is also used for the MPC cost function, but it is less common than the quadratic cost (i.e., ℓ_2-norm) [22]. This cost function produces sparse solutions in which the control becomes violent, implying that the actuators are either fully turned on or turned off. The primary drawback is that the cost is no longer differentiable at the origin, requiring the employment of non-smooth optimisation techniques like sub-gradients and proximal operators.

Dynamic Programming (DP) is a method developed in the 1950s by Bellman as an efficient means for solving multistage optimisation problems. The fundamental idea is to decompose a complex problem into smaller, simpler sub-problems and solve them recursively. Instead of computing the solution for the optimal control problem presented in (3.3), there exist similar sub-optimisation problems that are solved. The method under consideration is denoted as backward DP because the variables are found in reverse order. Thus, the optimal solutions are discovered as functions of the variables optimised in the following stage.

In the context of MPC, the DP method seeks to minimise the trajectory cost between sets $\mathcal{X}_j$ characterising all the admissible states that can be driven to $\mathbb{X}_f$ by an admissible control sequence (i.e., a control sequence that satisfies the constraints of (3.3)) in j steps. The value function $V^*(\boldsymbol{x}|j)$ describes the cost-to-go between sets, with j representing the time-to-go (i.e., defined in discrete-time instants). Each value function has an implicit MPC control law $\boldsymbol{\kappa}(\boldsymbol{x}|N)$ associated. DP delivers a sequence of optimal control inputs $\boldsymbol{u}^*(k) := \{\boldsymbol{u}^*(0), \boldsymbol{u}^*(1), \ldots, \boldsymbol{u}^*(N-1)\}$ implying the respective control laws $\boldsymbol{\kappa}^* = \{\boldsymbol{\kappa}^*(N), \boldsymbol{\kappa}^*(N-1), \ldots, \boldsymbol{\kappa}^*(1)\}$, with $\boldsymbol{u}_j^* : \mathcal{X}_j \to \mathbb{U}$. Thus, a MPC's system satisfies

$$\boldsymbol{x}^*(k+1) = f\left(\boldsymbol{x}(k), \boldsymbol{\kappa}(\boldsymbol{x}|N-k)\right), \ \forall i = 0, 1, \ldots, N-1. \tag{3.5}$$

For all $\boldsymbol{x} \in \mathcal{X}_j$, the DP equations [14] are described by

$$\begin{aligned} V^*(\boldsymbol{x}|j) &= \min_{\boldsymbol{u} \in \mathbb{U}} \left\{ \ell(\boldsymbol{x}, \boldsymbol{u}) + V^*(\boldsymbol{x}|j-1) \mid f(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{X}_{j-1} \right\}, \\ \boldsymbol{\kappa}^*(\boldsymbol{x}|j) &= \arg\min_{\boldsymbol{u} \in \mathbb{U}} \left\{ \ell(\boldsymbol{x}, \boldsymbol{u}) + V^*(\boldsymbol{x}|j-1) \mid f(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{X}_{j-1} \right\}. \end{aligned} \tag{3.6}$$

Nevertheless, this problem can occasionally become intractable [14], which is unfortunate given that MPC for constrained systems is rarely linear. However, the model's linearity along the polytopic struc-

ture of stage and terminal constraints and the convexity of the cost function results in computationally practical optimisation problems. In the following, some required definitions are provided.

Definition 3.1.1 (Control Invariant Set, [14])**.** A set $\mathcal{A}$ is control invariant for $\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k)), \boldsymbol{u} \in \mathbb{U}$, if, for all $\boldsymbol{x}(k) \in \mathcal{A}$, there exists an $\boldsymbol{u}(k) \in \mathbb{U}$ such that $f(\boldsymbol{x}(k), \boldsymbol{u}(k)) \in \mathcal{A}\ \forall k \in \mathbb{N}_0$.

In practice, if a state belongs to a specific set, all subsequent states must belong to the same one. Therefore, the system is invariant.

Definition 3.1.2 (Constraint Admissible Set, [14])**.** Given a control law $\boldsymbol{\kappa}(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$, a set of admissible states $\mathbb{X} \subset \mathbb{R}^{n_x}$, and a set of state and input constraints $\mathcal{Z} \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ the set $\mathbb{X}$ is called constraint admissible if

$$\forall \boldsymbol{x}(k) \in \mathbb{X} \text{ it holds that } (\boldsymbol{x}(k), \boldsymbol{\kappa}(\boldsymbol{x}(k)) \in \mathcal{Z}. \tag{3.7}$$

In conclusion, if the terminal set $\mathbb{X}_f$ is control invariant, the set of initial conditions $\mathcal{X}_N$, for which (3.3) is feasible, is positive invariant.

Definition 3.1.3 (Recursive Feasibility, [14])**.** If the problem is feasible at $k = 0$, it implies the problem's feasibility $\forall k \geq 0$.

This property is of the utmost importance because it ensures that the optimal control problem is feasible for all successor states if it is for the initial condition.

3.1.1 Stability Analysis

Stability is critical in MPC systems for ensuring reliable and robust performance. The ability of the algorithm to maintain desirable behaviour and avoid erratic or divergent responses is referred to as its stability. The stability of a control system has been extensively researched in a vast amount of literature. Its principles are defined by Lyapunov in his studies of the stability and control of dynamical systems [33]. The central idea of Lyapunov's theory is to define a Lyapunov Function (LF) that characterises the system and assesses stability. Figure 3.2 is introduced to better characterise the Lyapunov stability.

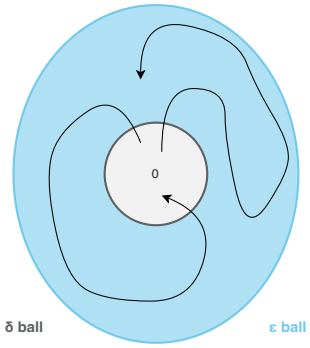


Figure 3.2: Geometric interpretation of Lyapunov Stability (adapted from [34]).

The origin of a discrete-time dynamical system is Lyapunov stable if

$$\forall \epsilon > 0, \exists \delta > 0 : \|x(0)\| < \delta \Rightarrow \|x(k)\| < \epsilon, \forall k \geq 0. \tag{3.8}$$

The origin of a discrete-time dynamical system is Lyapunov asymptotically stable in $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ if

$$\text{(3.8) and } \lim_{k\to\infty} \|x(k)\| = 0, \forall x(0) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}. \tag{3.9}$$

Consider a discrete-time system $x(k+1) = f(x(k))$ with $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ continuous and $f(0) = 0$. A LF definition is presented below.

Definition 3.1.4 (Lyapunov Function, [14])**.** A continuous function $V : \mathbb{R}^{n_x} \to \mathbb{R}^+$ is called a LF if:

1. $V(0) = 0$;
2. $V(\xi) > 0, \forall \xi \in \mathbb{X} \ \wedge \ \xi \neq 0$;
3. $V(f(\xi)) - V(\xi) \leq 0, \forall \xi \in \mathbb{X}$.

To ensure 1. and 2. $\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$ where $\alpha_1,\ \alpha_2$ are $\mathcal{K}_\infty$ - functions.

Definition 3.1.5 (Class $\mathcal{K}$ function, [14])**.** A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.

There are different types of LF:

- LF: $V(f(\xi)) - V(\xi) \leq 0, \forall \xi \in \mathbb{X}$;
- Strict LF: $V(f(\xi)) - V(\xi) < 0, \forall \xi \neq 0$;
- Uniformly strict LF: $V(f(\xi)) - V(\xi) \leq -\alpha_3(\|\xi\|) \ \vee \ V(f(\xi)) - \rho V(\xi) \leq 0, \rho \in [0, 1)$.

Consider a discrete-time system $x(k+1) = g(x(k), u(k))$ with $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ continuous and $g(0, 0) = 0$.

Definition 3.1.6 (Control Lyapunov Function, [14])**.** A continuous function $V : \mathbb{X} \to \mathbb{R}^+$ defined on a region $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ containing the origin in its interior is called a LF if guarantees condition 1. and 2. from Definition 3.1.4 and

3. There exists a control law $u = \kappa(\xi)$ such that

$$\begin{aligned} &V(g(\xi, \kappa(\xi))) - V(\xi) \leq 0, \ \forall \xi \in \mathbb{X} \wedge \xi \neq 0, \\ &g(\xi, \kappa(\xi)) \in \mathbb{X}, \ \kappa(\xi) \in \mathbb{U}, \forall \xi \in \mathbb{X}. \end{aligned} \tag{3.10}$$

Proposition 3.1.1. Suppose V is a control Lyapunov function for the system dynamics g and the control law κ. Then the origin of the CL system $x(k+1) = g(x(k), \kappa(x(k)))$ is asymptotically Lyapunov Stable (LS) in $\mathbb{X}$.

To ensure asymptotic stability for the system $x(k+1) = (A + BK)x(k)$ is important to choose $P \succ 0$ (i.e., positive definite) such that $P - (A + BK)^\top P(A + BK) \succ 0$. It means that choosing P to satisfy the previous equation, for some $Z \succ 0$, the discrete Lyapunov Equation

$$(A + BK)^\top P(A + BK) - P = -Z. \tag{3.11}$$

So $(A + BK)$ is stable if, and only if, $P \succ 0$ is found for any $Z \succ 0$.

Theorem 3.1.1 (Stability of MPC)**.** The origin is an asymptotically stable equilibrium point of the CL system

$$x(k+1) = Ax(k) + BKx(k),$$

if the following conditions are met:

1. $Q \succ 0$ and $R \succ 0$;

2. Terminal cost weight P satisfies the discrete Lyapunov Equation (3.11) with $Z = Q + K^\top RK$, where $K \in \mathbb{R}^{n_u \times n_x}$ is any matrix with $\rho(A + BK) < 1$.

The spectral radius (maximum absolute value of the eigenvalues) of the CL system matrix is denoted by ρ. It is used as a stability criterion to ensure the CL system is stable. If the spectral radius is less than 1 (i.e., $\rho(A+BK) < 1$), the system is stable and converges to a steady-state solution. On the other hand, if the spectral radius is greater than or equal to 1 (i.e., $\rho(A + BK) \geq 1$), the system is unstable and does not converge to a steady-state solution.

The type of MPC considered determines the stability formulation. Furthermore, two analyses are performed: unconstrained MPC and constrained MPC. Assuming $Q \succ 0$ and $R \succ 0$ for an unconstrained MPC it pursues that

1. $V(0) = 0$;

2. $V(\xi) \geq \xi^\top Q\xi, \forall \xi \neq 0$;

3. $V(\xi) \to \infty$ as $\|\xi\| \to \infty$.

There is a need to ensure, in addition, that

$$V(x(k+1)) - V(x(k)) < 0, \ \forall x(k) \neq 0. \tag{3.12}$$

The input sequence at the next instant $k+1$ is characterised by $\tilde{U}_{k+1}$. The cost function associated is defined as

$$\begin{aligned}
\tilde{V}\left(x(k+1)\right) = J\left(x(k+1), \tilde{U}_{k+1}\right) = &+ V\left(x(k)\right) && \text{Optimal cost at } k \text{ for } x(k) \\
&- \ell\left(x(k), u^*{}_{0|k}\right) && \text{First stage cost at } k \\
&- V_f\left(x^*{}_{N|k}\right) && \text{Terminal cost at } k \\
&+ \ell\left(x^*{}_{N|k}, Kx^*{}_{N|k}\right) && \text{Last } (N-1) \text{ stage cost at } k+1 \\
&+ V_f\left((A+BK)x^*{}_{N|k}\right) && \text{Terminal cost at } k+1
\end{aligned}$$

where the notation $x^*_{i|k}$ and $u^*_{i|k}$ denotes the optimal value of the predicted state and input at instant i, based on the plant state $x(k)$ at instant k. Thus, $x_{0|k} = x(k)$ and $u(k) = u_{0|k}$.

If it is possible to prove that the sum of the last three terms is negative, then (3.12) becomes

$$\tilde{V}(x(k+1)) - V(x(k)) \leq -\ell\left(x(k), {u^*}_{0|k}\right) \leq -x(k)^\top Q x(k) < 0,\ \forall x(k) \neq 0, \tag{3.13}$$

this condition is equivalently represented for any ${x^*}_{N|k} = \xi$ as

$$V_f\left(f(\xi)\right) - V_f\left(\xi\right) \leq \ell(\xi, K\xi) \leq -\xi^\top Q \xi < 0, \forall \xi \neq 0, \tag{3.14}$$

which corresponds to Property 3. of control LF for $V(\xi) = V_f(\xi)$ and $u = \kappa(\xi) = K\xi$.

Also, the optimality of the value function $V(x(k+1)) = J\left(x(k+1), U^*_{k+1}\right)$ it holds that

$$V(x(k+1)) = J\left(x(k+1), U^*_{k+1}\right) \leq J\left(x(k+1), \tilde{U}_{k+1}\right) = \tilde{V}(x(k+1)). \tag{3.15}$$

Hence

$$V(x(k+1)) - V(x(k)) \leq \tilde{V}(x(k+1)) - V(x(k)) < 0,\ \forall x(k) \neq 0. \tag{3.16}$$

Therefore, it is established that the MPC value function $V(\cdot)$ is LF for the CL MPC system, which further yields asymptotic Lyapunov Stability by Lyapunov's Theorem.

Consider the CL system subject to the constraints $Mx(k) + Eu(k) \leq b, \forall k \in \mathbb{N}$ and assuming the terminal cost satisfies the stability conditions for unconstrained linear MPC. If $\mathbb{X}_f = \{\xi : M_N \xi \leq b_N\}$ is chosen to be an invariant and constraint admissible set for the CL system and the constrained linear MPC problem is feasible at $k = 0$ for $x(0)$. Thus, the MPC problem is feasible at instant k for $x(k), \forall k \in \mathbb{N}$, where $x(k+1)$ is generated by

$$x(k+1) = Ax(k) + Bu^*_{0|k} = Ax(k) + BKx(k). \tag{3.17}$$

Some design steps can be taken to ensure constraint fulfilment and stability:

1. Choose $Q \succ 0$ and $R \succ 0$;

2. Compute a stabilising terminal control law $u(k) = Kx(k)$ (i.e., $\rho(A+ BK) < 1$);

3. Compute the terminal cost weight P such that $P \succ 0$ and

$$(A + BK)^\top P(A + BK) - P \preceq -Q - K^\top RK,$$

   which means that the terminal cost $V_f(x) = x^\top Px$ is a Control Lyapunov Function for the CL dynamics $(A + BK)$;

4. Compute a terminal set $\mathbb{X}_f = \{\xi \in \mathbb{R}^n : M_N \xi \leq b_N\}$ such that for all $\xi \in \mathbb{X}_f$ it holds that $(M + EK)\xi \leq b$ is constraint admissible and $M_N(A + BK)\xi \leq b_N$ is invariant.

3.2 Constrained Convex Generators

Modern control algorithms extensively operate with sets such as intervals, ellipsoids, zonotopes, and polytopes as computational objects to characterise reachable or invariant sets of dynamical systems [35]. Nonetheless, this may introduce unnecessary conservatism into the solution. In [15], a direct approach can over-approximate sets by a polytope, whether described in hyper-plane or CZ formulation.

Definition 3.2.1 (Constrained Zonotope, [16])**.** A set Z is a CZ defined by the tuple $(G, c, A, b) \in \mathbb{R}^{n\times n_g} \times \mathbb{R}^n \times \mathbb{R}^{n_c\times n_g} \times \mathbb{R}^{n_c}$ such that:

$$Z = \{G\xi + c : \|\xi\|_\infty \leq 1,\ A\xi = b\},$$

where n is the number of states, n_g the number of generators and n_c the number of constraints.

Computations for linear estimation and control problems typically involve standard set operations such as Minkowski sums, linear mappings, and intersections.

Consider three CZs: $Z = (G_z, c_z, A_z, b_z) \subset \mathbb{R}^n$; $W = (G_w, c_w, A_w, b_w) \subset \mathbb{R}^n$; $Y = (G_y, c_y, A_y, b_y) \subset \mathbb{R}^m$; matrix $R \in \mathbb{R}^{m\times n}$ and vector $t \in \mathbb{R}^m$. The set operations are as follows:

$$RZ + t = (RG_z,\ Rc_z + t,\ A_z,\ b_z), \tag{3.18}$$

$$Z \oplus W = \left(\begin{bmatrix} G_z & G_w \end{bmatrix},\ c_z + c_w,\ \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix},\ \begin{bmatrix} b_z \\ b_w \end{bmatrix} \right), \tag{3.19}$$

$$Z \cap_R Y = \left(\begin{bmatrix} G_z & 0 \end{bmatrix},\ c_z,\ \begin{bmatrix} A_z & 0 \\ 0 & A_y \\ RG_z & -G_y \end{bmatrix},\ \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix} \right). \tag{3.20}$$

These operations are obtained in the CZ formulation for Z, W, and Y as:

$$RZ + t = \{RG_z\xi + Rc_z + t : \|\xi\|_\infty \leq 1,\ A_z\xi = b_z\} = \{RG_z\xi + Rc_z + t : \xi \in \mathcal{C}_z,\ A_z\xi = b_z\}, \tag{3.21}$$

$$Z \oplus W = \{G_z\xi_z + G_w\xi_w + c_z + c_w : A_z\xi_z = b_z, A_w\xi_w = b_w,\ \xi_z \in \mathcal{C}_z,\ \xi_w \in \mathcal{C}_w\}, \tag{3.22}$$

$$Z \cap_R Y = \{G_z\xi_z + c_z : A_z\xi_z = b_z, A_y\xi_y = b_y,\ \xi_z \in \mathcal{C}_z,\ \xi_y \in \mathcal{C}_y,\ RG_z\xi + Rc_z = G_y\xi_y + c_y\}. \tag{3.23}$$

$\mathcal{C}_{z,w,y}$ represents the unit ℓ_∞-norm ball. It is clear that nothing is pushing the use of the unit ball following a p-norm as the generator and could also extend the definition to convex cones.

To represent state estimation sets for discrete-time Linear Time-Varying (LTV) systems as interval arithmetic, zonotopes, ellipsoids, CZ, and polytopes can be employed, especially when dealing with nonlinear systems. In this case, the dynamics are approximated by a linear function that allows set propagation. However, these approaches are infeasible if the intended goal is to ensure safe vehicle passage without collision with obstacles modelled as convex bodies.

Silvestre observed that CZ for polytopes may be extended to incorporate generators from within a convex set, which motivated the CCGs formulation. This innovative concept for sets includes ℓ_2 unit balls to generate smooth surfaces, ℓ_∞ unit balls to develop facets, and cones to yield unbounded sets.

Definition 3.2.2 (Constrained Convex Generators, [15])**.** A CCG $Z \in \mathbb{R}^n$ is defined by the tuple $(G, c, A, b, \mathfrak{C})$ with $G \in \mathbb{R}^{n \times n_g}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{n_c \times n_g}$, $b \in \mathbb{R}^{n_c}$, and $\mathfrak{C} := \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_p}\}$ such that:

$$\mathcal{Z} = \left\{ G\xi + c : A\xi = b, \xi \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_{n_p} \right\}.$$

The sets Z, W, and Y are able to be specified as CCGs: $Z = (G_z, c_z, A_z, b_z, \mathfrak{C}_z) \subset \mathbb{R}^n$; $W = (G_w, c_w, A_w, b_w, \mathfrak{C}_w) \subset \mathbb{R}^n$; $Y = (G_y, c_y, A_y, b_y, \mathfrak{C}_y) \subset \mathbb{R}^m$; matrix $R \in \mathbb{R}^{m \times n}$, vector $t \in \mathbb{R}^m$, and the set operations as:

$$RZ + t = (RG_z,\ Rc_z + t,\ A_z,\ b_z,\ \mathfrak{C}_z), \tag{3.24a}$$

$$Z \oplus W = \left(\begin{bmatrix} G_z & G_w \end{bmatrix},\ c_z + c_w,\ \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix},\ \begin{bmatrix} b_z \\ b_w \end{bmatrix},\ \{\mathfrak{C}_z,\ \mathfrak{C}_w\} \right), \tag{3.24b}$$

$$Z \cap_R Y = \left(\begin{bmatrix} G_z & 0 \end{bmatrix},\ c_z,\ \begin{bmatrix} A_z & 0 \\ 0 & A_y \\ RG_z & -G_y \end{bmatrix},\ \begin{bmatrix} b_z \\ b_y \\ c_y - Rc_z \end{bmatrix},\ \{\mathfrak{C}_z,\ \mathfrak{C}_y\} \right). \tag{3.24c}$$

The three primary set operations are presented as closed-form solutions with results on efficient computational times for real-time scenarios. Due to these inherent characteristics, coupled with the intrinsic convexity of CCGs, this approach is suitable for tasks such as state estimation and fault detection in LTV models [15]. Keeping track of the generator type associated with each entry of the auxiliary variable vector ξ becomes imperative. This representation allows an arbitrary set of convex sets to cover the generator variables, characterising a novel concept of the CZ polytopes. It strikes a trade-off between the accuracy required to represent the set and the complexity of the computations.

3.3 Vehicle Model

This thesis resorted to the quadrotor model as the application. The quadrotor consists of four identical rotors and propellers arranged in a configuration resembling a square, displayed in Figure 3.3. Thrust and torque are generated perpendicular to the plane formed by the square configuration of these components. The references are the inertial frame $\{\overrightarrow{x}_I, \overrightarrow{y}_I, \overrightarrow{z}_I\}$ and the body-fixed frame $\{\overrightarrow{x}_B, \overrightarrow{y}_B, \overrightarrow{z}_B\}$ with the origin positioned at the vehicle's centre of mass.

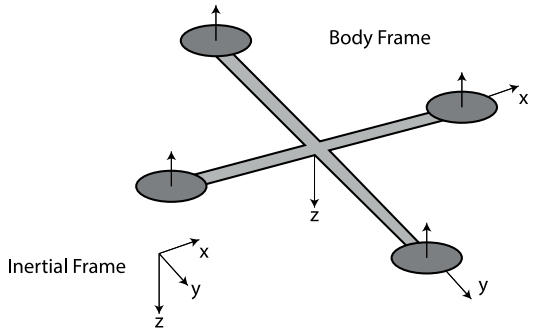


Figure 3.3: Quadrotor model (adapted from [36]).

The attitude of any rigid body is typically represented by a rotation matrix belonging to the Special Orthogonal Group SO(3). Nonetheless, the configuration is defined concerning the centre of mass and attitude of the inertial frame, represented by the rotation matrix $\boldsymbol{R}$. Therefore, the configuration manifold is the Special Euclidean group SE(3) [37]. Furthermore, this thesis refrains from delving into the intricacies of rotor and propeller dynamics. As a result, it assumes the following system configuration: i) the thrust of each propeller is directly controlled; ii) the torque generated by each propeller is directly proportional to its thrust; iii) a sequence of alternating propeller rotations: one propeller rotates anticlockwise, followed by another that rotates clockwise, and so on. The torque generated by the i-th propeller $\tau_i \in \mathbb{R}$ can be described as $\tau_i = (-1)^i \; c \; t_i$, where $t_i \in \mathbb{R}$ is the thrust generated by the i-th propeller and c a fixed constant. Hence, the total moment $\boldsymbol{M} \in \mathbb{R}^3$ is given by

$$\boldsymbol{M} = \begin{bmatrix} d(t_4 - t_2) & d(t_1 - t_3) & c(-t_1 + t_2 - t_3 + t_4) \end{bmatrix}^\top, \tag{3.25}$$

where $d \in \mathbb{R}$ is the horizontal distance from the centre of mass to the centre of a rotor.

The equations of motion of this quadrotor UAV can be described as

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{3.26a}$$

$$m\dot{\boldsymbol{v}} = mgz_I - T\boldsymbol{R}z_I, \tag{3.26b}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R} \times \boldsymbol{\omega}, \tag{3.26c}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} = \boldsymbol{M}, \tag{3.26d}$$

where: $\boldsymbol{p} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{v} \in \mathbb{R}^{3\times3}$ are the position and velocity for the inertial frame of the vehicle; $m \in \mathbb{R}^+$ is the total mass; g is the constant $g = 9.8 \, \mathrm{m \, s^{-2}}$ for the Earth's gravity; $z_I \in \mathbb{R}^3$ is the standard basis vectors in the inertial frame; $T \in \mathbb{R}$ is the total thrust resulting from the sum of the thrust generated by each propeller; $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ symbolises the rotation transformation from the body-fixed to the inertial frame; for the body-fixed frame: $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is the inertia tensor; $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity; and $\boldsymbol{M} \in \mathbb{R}^3$ is the total moment.

The details for this formulation are presented in the study [37]. The vehicle model serves as the foundation for subsequent set-based analyses, allowing the investigation of its states and the development of effective control strategies to improve performance.

3.4 Problem Statement

The quadrotor is modelled as a triple integrator, which fits the formulation for a linear dynamical system. The vehicle dynamics are given by

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) + \boldsymbol{d}(k), \; \forall k \geq 0, \tag{3.27}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$, $\boldsymbol{u}(k) \in \mathbb{R}^{n_x}$, and $\boldsymbol{d}(k) \in \mathbb{R}^{n_d}$ are respectively the state, input and disturbance signals at instant k. Additionally, the MPC is defined by the constrained optimal problem in (3.3). The system is subject to constraints on states and control inputs and is susceptible to bounded disturbances.

Under such circumstances, the research directs attention to two separate yet interconnected problems.

Problem 3.4.1 (Region of Attraction)**.** The objective is to define the exact region of attraction, which encompasses all the initial states that guarantee the feasibility and stability of the system throughout its mission. Therefore, this set must exhibit invariance, adhere to constraints, and ensure recursive feasibility for all states within its boundaries (Definition 3.1.1, 3.1.2, and 3.1.3).

Problem 3.4.2 (Model Predictive Controller)**.** Design the MPC considering the following requirements:

- Subject to the physical and safety constraints inherent in the quadrotor model.
- Focusing on acceleration constraints to ensure effective attitude coordination whilst handling saturation and rate limits in trajectory generation.
- Once the system reaches the terminal set, it remains there indefinitely, facilitating a seamless transition to the LQR.
- Feasibility is guaranteed for a duration significantly exceeding the prediction horizon.

Chapter 4

Proposed Solution

After reviewing fundamental concepts and theoretical foundations in the preceding chapters, the present chapter consolidates the prior knowledge to emphasise the development of the proposed solution. This solution determines all the launch points of the manoeuvre that certify the vehicle attains a suitable relative orientation. Consequently, the problem feasibility is guaranteed *a priori* accounting for the worst-case estimation for sets analysis.

4.1 Methodologies

The vehicle dynamics presented in Section 3.3 is the basis for accurately defining the constraint sets in Section 4.2.1. The controller is employed to manage the translation dynamics of a quadrotor, wherein the mathematical model represents it as a triple integrator, accounting for acceleration, velocity, and position variables. The novel technique for set-valued state estimation presented in Section 3.2 enables the accurate representation of the worst-case estimation of the states. Furthermore, the strategy relies on the exact definition of the terminal set that the system eventually reaches and remains there despite disturbances. This set is defined as a mRPI in Section 4.2.2, which allows the smooth transition to the LQR controller after MPC reaches the set. Additionally, the robust terminal set enables the backward propagation to define the region of attraction presented in Section 4.2.3. Given the admissible values for the unknown signals, this region encompasses all the possible values for the initial state, yielding a feasible trajectory. The system is examined under severe conditions to guarantee its precise estimation.

The quadrotor model provides a linear system, admitting the integration of CCGs as constraint sets of the MPC designed in Section 4.3. The backward propagation allows the definition of the admissible states in each instant of the trajectory (i.e. respecting system constraints and avoiding obstacles), incorporated as a varying terminal set in the MPC. Finally, attention is devoted to the critical aspects of obstacle avoidance in Section 4.4 outlining the strategies and algorithms.

4.2 Set-Based Analysis

This section delves into the globe of set-based analysis to understand the complexities of modelling vehicle dynamics and harness its potential for improved control strategies. It accounts for all the disturbances that affect the system to ensure safety for future instances and prevent collisions. The set estimations technique builds upon the CCGs formulation and operations presented in Section 3.2.

The CCGs are directly encoded in YALMIP [38] on MATLAB [39] using the Reach Toolbox [40] developed by Silvestre. YALMIP is a modelling language that eases the development of optimal problems.

4.2.1 Admissible Sets

Admissible sets define the feasible system states that satisfy specified constraints and performance criteria. As mentioned by Yang et al., these sets often exhibit intricate geometries. The inclusion of CCGs plays a pivotal role in this facet of the research, as it facilitates the establishment of a robust geometry. By defining these sets, it is possible to earn insights into the vehicle's manoeuvrability and operational boundaries.

The translation dynamics are modelled as a triple integrator to access the vehicle's orientation and angular velocity. To effectively control a vehicle's motion, it is essential to consider translation aspects (e.g., position and velocity) and rotational elements (e.g., orientation and angular velocity). The orientation in vehicle dynamics refers to the vehicle's spatial positioning, including roll, pitch, and yaw angles. Understanding and controlling orientation is vital for tasks like attitude stabilisation, autonomous flight, or precise manoeuvring. Angular velocity characterises the rate at which a vehicle rotates around its axes. Assumes critical importance in fundamental tasks, such as sustaining stable flight, managing a vehicle's heading, and preventing undesirable spinning or tumbling. The triple integrator is implemented as

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{v}} &= \boldsymbol{a}, \\ \dot{\boldsymbol{a}} &= \boldsymbol{u}, \end{aligned} \tag{4.1}$$

where the states are the position $\boldsymbol{p} \in \mathbb{R}^{3\times3}$, the velocity $\boldsymbol{v} \in \mathbb{R}^{3\times3}$ and the linear acceleration $\boldsymbol{a} \in \mathbb{R}^{3\times3}$, which is driven by the control input $\boldsymbol{u} \in \mathbb{R}^3$. Thus, each state is related to x, y, and z directions.

Firstly, the triple integrator is discretised with sampling time T_s and the assumption that the input $\boldsymbol{u}$ and disturbances $\boldsymbol{d}$ remain constant between sampling instances,

$$\begin{aligned} \boldsymbol{p}(k+1) &= \boldsymbol{p}(k) + T_s\boldsymbol{v}(k) + \frac{{T_s}^2}{2}\boldsymbol{a}(k) + \frac{{T_s}^3}{6}\boldsymbol{u}(k) + \boldsymbol{d}(k), \\ \boldsymbol{v}(k+1) &= \boldsymbol{v}(k) + T_s\boldsymbol{a}(k) + \frac{{T_s}^2}{2}\boldsymbol{u}(k) + \boldsymbol{d}(k), \\ \boldsymbol{a}(k+1) &= \boldsymbol{a}(k) + T_s\boldsymbol{u}(k) + \boldsymbol{d}(k), \end{aligned} \tag{4.2}$$

which is represented in the following state-space model

$$\begin{bmatrix} \boldsymbol{p}(k+1) \\ \boldsymbol{v}(k+1) \\ \boldsymbol{a}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & T_s\boldsymbol{I} & \frac{{T_s}^2}{2}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I} & T_s\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}(k) \\ \boldsymbol{v}(k) \\ \boldsymbol{a}(k) \end{bmatrix} + \begin{bmatrix} \frac{{T_s}^3}{6}\boldsymbol{I} \\ \frac{{T_s}^2}{2}\boldsymbol{I} \\ T_s\boldsymbol{I} \end{bmatrix} \boldsymbol{u}(k) + \boldsymbol{d}(k), \tag{4.3}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$.

To describe a correct trajectory and to be aware of the vehicle's limitations, position, velocity, and acceleration bounds must be defined. Thus, the linear velocity can be enforced to prevent the vehicle from reaching speeds where the aerodynamic drag force is no longer negligible [19]. The set $\mathbb{P}$ is related to the position and $\mathbb{V}$ to the velocity. Regarding the CCGs representation (Definition 3.2.2), these constraints are expressed as a hypercube (i.e., defined by an ℓ_∞-norm) centred at the origin in which the generator matrix G contains the maximum of each parameter on the diagonal,

$$\begin{aligned} G_{\mathbb{P}} &= \operatorname{diag}\left(p_{\max_x}, p_{\max_y}, p_{\max_z}\right), \\ G_{\mathbb{V}} &= \operatorname{diag}\left(v_{\max_x}, v_{\max_y}, v_{\max_z}\right), \\ c_{\mathbb{P}} &= c_{\mathbb{V}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top. \end{aligned} \tag{4.4}$$

The final state constraint $\mathbb{A}$ should consider the maximum thrust and angular rates the vehicle can achieve and the gravity force. The maximum value of thrust $T_{max} \in \mathbb{R}$ is associated with the rotating propellers. According to the linear velocity dynamics (3.26b), the acceleration results in a sphere

$${a_x}^2 + {a_y}^2 + (a_z - g)^2 \leq \left(\frac{T_{max}}{m}\right)^2. \tag{4.5}$$

Hence, the CCG set is represented by an ℓ_2-norm and the matrices

$$\begin{aligned} G_{\mathbb{A}} &= \frac{T_{max}}{m} I_3, \\ c_{\mathbb{A}} &= \begin{bmatrix} 0 & 0 & g \end{bmatrix}^\top. \end{aligned} \tag{4.6}$$

In [19], the boundary of this sphere is approximated with a polyhedron, underscoring the significant advantages of employing CCGs to depict this set.

To have the sets in agreement with the limitations of the states, a Cartesian product must be applied as presented in the following definition.

Definition 4.2.1 (Cartesian Product between Constrained Convex Generators (CCGs))**.** Given the sets Z and W able to be specified as CCGs: $Z = (G_z, c_z, A_z, b_z, \mathfrak{C}_z) \subset \mathbb{R}^n$; $W = (G_w, c_w, A_w, b_w, \mathfrak{C}_w) \subset \mathbb{R}^n$. The Cartesian product is characterised as

$$Z \times W = \left(\begin{bmatrix} G_z & 0 \\ 0 & G_w \end{bmatrix}, \begin{bmatrix} c_z \\ c_w \end{bmatrix}, \begin{bmatrix} A_z & 0 \\ 0 & A_w \end{bmatrix}, \begin{bmatrix} b_z \\ b_w \end{bmatrix}, \{\mathfrak{C}_z, \mathfrak{C}_w\} \right). \tag{4.7}$$

The set of admissible states $\mathbb{X}$ is obtained as $\mathbb{X} := \mathbb{B} \times \mathbb{A}$, where $\mathbb{B} := \mathbb{P} \times \mathbb{V}$. The matrices A

and vectors b for the states set are empty with defined dimensions to maintain consistency through operations.

The study conducted by da Silva Pinto et al. illustrates that the roll p and pitch q rates can be expressed through the dot product between u and the basis vectors in the body frame as

$$\begin{aligned} p &= -\frac{m}{T} u_x x_B, \\ q &= -\frac{m}{T} u_y y_B. \end{aligned} \tag{4.8}$$

The vector's norm $\|\boldsymbol{u}\|$ must be bounded to control the roll and pitch rates, imposed by $\boldsymbol{\omega} = [p\ q\ r]^\top$. Assuming $p_{max} = q_{max} = \Omega \in {\mathbb{R}_0}^+$ it comes

$$\|u\| \leq \frac{\tilde{T}}{m}\Omega, \tag{4.9}$$

where $\tilde{T} \leq T_{max}$. The selection of $\tilde{T}$ signifies the degree to which the design allows angular rate inputs to exceed the maximum achievable value.

The admissible control inputs $\mathbb{U}$ results in a sphere. As mentioned earlier, the approach presented in [19] does not consider this geometry. It samples points from the sphere's edge and calculates the convex hull of those points, resulting in $\mathbb{U}$ described as a polyhedron. The set is represented in CCG format by ℓ_2-norm with the centre at the origin and the radius on the diagonal of the generator matrix as

$$\begin{aligned} G_{\mathbb{U}} &= \frac{\tilde{T}}{m}\Omega I_3, \\ c_{\mathbb{U}} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top. \end{aligned} \tag{4.10}$$

Lastly, the disturbance signals must be bounded to guarantee $\boldsymbol{d}(k) \subset \mathbb{D},\ \forall k \geq 0$. These disturbances affect the measured control input, influencing the vehicle's acceleration. The set $\mathbb{D}$ is defined as an ℓ_∞-norm centred at the origin, with maximum disturbance values $D_{\max} \in \mathbb{R}$ concerning the acceleration within the generator matrix as

$$\begin{aligned} G_{\mathbb{D}} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & D_{\max}\boldsymbol{I} \end{bmatrix}^\top, \\ c_{\mathbb{D}} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top, \end{aligned} \tag{4.11}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ and $\mathbf{0} \in \mathbb{R}^{3\times3}$.

4.2.2 Robust Positively Invariant Set

The concept of a Robust Positively Invariant (RPI) set is crucial for designing control strategies that guarantee the vehicle's stability, even in the presence of disturbances. Once entered, the system remains within it indefinitely. When the system is stable without disturbances, the RPI is trivially a set with zero because all trajectories converge to zero after a sufficient time.

As examined in [24], one approach to guarantee feasibility involves formulating a terminal set as a pre-computed robust controller invariant set by imposing constraints on the states. Thus, the terminal

region is controlled invariant, constrained admissible and guarantees the recursive feasibility for the rest of the trajectory, leading to a positively invariant set.

The system under investigation is designed to be CL stable (i.e., $\rho(A + BK) < 1$) and contains unknown disturbances. Consequently, the conditions to compute a RPI have been met. Looking into the vehicle dynamics of the CL system

$$\boldsymbol{x}(k+1) = (A - BK)\boldsymbol{x}(k) + \boldsymbol{d}(k),\ \forall k \geq 0. \tag{4.12}$$

Therefore, the dynamics in order of $\boldsymbol{x}(k)$ can be simplified as

$$\boldsymbol{x}(k) = (A - BK)^k \boldsymbol{x}(0) + \sum_{i=0}^{k=N} (A - BK)^{k-i} \boldsymbol{d}(i). \tag{4.13}$$

The terminal set depicts the domain to where all states converge in infinity, which may be computed as

$$\mathbb{X}_f := \lim_{k\to\infty} \boldsymbol{x}(k) = \underbrace{\mathcal{A}^k \boldsymbol{x}(0)}_{\to 0} + \sum_{i=0}^{k=N} \mathcal{A}^{k-i} \boldsymbol{d}(i), \tag{4.14}$$

where $\mathcal{A} \in \mathbb{R}^{n_x \times n_x}$ represents the CL-matrix of the system $\mathcal{A} := A - BK$. It is important to note that if, and only if, $\boldsymbol{d}(k) \subset \mathbb{D}$ and $\mathbb{D}$ does not change over time, the terminal set can be described as

$$\mathbb{X}_f := \sum_{i=0}^{k=N} \mathcal{A}^{k-i} \mathbb{D}. \tag{4.15}$$

The RPI sets typically have many approximations, but there are techniques for simplifying the computations. Due to their computational efficiency, zonotopes have been implemented in MPC reach set calculations, as stated in [41] and [42].

The proposed method uses CCGs to represent sets which portray a new concept of CZs of polytopes that allows the application of some of the methods presented by Raghuraman and Koeln to calculate the RPI. One of the strategies involves operations like the Minkowski sum, which can be performed on CCGs as demonstrated in (3.24b). However, computing such a set raises a trade-off between accuracy, complexity, and computation time.

Inner approximations are commonly employed in the set domain. This technique improves the practicality of these sets' representation and ensures accuracy - a significant aspect for computing backwards reachable sets from the RPI. If the representation lacks accuracy, the inaccuracy is carried forward throughout the process. This technique restricts the complexity of the set, ensuring it satisfies a predetermined upper limit on the number of generators and constraints. Additionally, it facilitates the calculations required for the analysis and control process. The approximation of RPI generates a set named mRPI, defined below.

Definition 4.2.2 (The minimal Robust Positively Invariant, [42])**.** The mRPI set F_∞ is the RPI set that is

contained in every closed RPI set of (4.12) and is given by

$$F_\infty = \bigoplus_{i=0}^{\infty} \mathcal{A}^i \mathbb{D}. \tag{4.16}$$

However, due to the infinite sequence of Minkowski sums, obtaining an explicit characterisation of F_∞ presents a challenge. As a result, the mRPI is defined in this dissertation by the calculation of F_s iteratively as

$$F_s = \left(\bigoplus_{i=0}^{N} \mathcal{A}^i \ \mathbb{D} \right) \oplus \left(\left((I_{n_x} - \mathcal{A})^{-1} - \sum_{0}^{N} \mathcal{A}^i \right) \mathbb{D} \right), \tag{4.17}$$

where $\mathbb{D}$ is the disturbances set. In this case, $\mathbb{D}$ is a set with disturbances only in the three dimensions influencing the vehicle's acceleration.

Analysing (4.17), the left side calculates the inner approximation of the RPI by only summing until N. Also, the right side performs a numerical evaluation of the upper limit (i.e., overbound) for the residual portion of the expression. Consequently, the impact of this upper limit on the disturbances set is considered by employing the linear map. This technique allows the robust estimation of the upper bound by effectively accounting for all dimensions influenced by disturbances. Finally, the Minkowski sum of these two parts guarantees an overbound of the inner approximation of the RPI, representing the mRPI.

Algorithm 1 implements the F_s computation. It improves the method's time by considerably reducing the number of iterations required to calculate the mRPI, achieving a good balance between computational time and accuracy with this strategy.

Algorithm 1 mRPI considering an inner approximation.

Input: $\mathcal{A}$, $\mathbb{D}$
Output: mRPI

1: innerRPI $\leftarrow \mathbb{D}$
2: **for** $i = 1 : N$ **do**
3: Calculate the inner approximation: innerRPI $\leftarrow$ innerRPI $\oplus$ $(\mathcal{A}^i \mathbb{D})$
4: **end for**
5: $R \leftarrow (I_{n_x} - \mathcal{A})^{-1}$
6: **for** $i = 0 : N$ **do**
7: Numerical evaluation of the upper limit: $R \leftarrow R - \mathcal{A}^i$
8: **end for**
9: Compute the upper limit within the set: $\mathbb{O} \leftarrow R\mathbb{D}$
10: Compute the mRPI: $F_s \leftarrow$ innerRPI $\oplus$ $\mathbb{O}$

The concept of inner approximations proves highly valuable in practical applications. By accurately approximating the RPI, the offline computation of the backward reachable sets from a mRPI results in a decreased computational cost, as highlighted by [26]. These successive computations result in an accurate and reliable set of initial states (i.e., the region of attraction), ensuring that the system eventually reaches a specified target region (i.e., the mRPI).

The region of attraction is determined by the size of the terminal set and the control horizon. As a result, enlarging the control horizon expands the zone but at a higher computational cost, whereas increasing the terminal set produces an enlargement at no extra cost [26].

4.2.3 Region of Attraction

The region of attraction encompasses all the initial states that guarantee the feasibility and stability of the system. The proposed approach computes this region through backward propagation to ensure that the system ultimately converges for the terminal set, defined as a mRPI. Thus, it is possible to switch to the LQR and remain in the set regardless of disturbances. This concept is depicted in Figure 4.1 by tracing the state trajectory, elucidating the vehicle's progression with each state residing within its respective set.

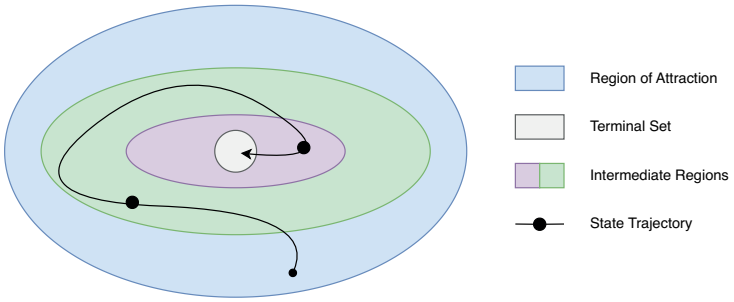


Figure 4.1: Illustration of state trajectories within specific regions.

The method relies on the principles outlined in Section 3.1 regarding DP to minimise the trajectory cost across sets. A fundamental aspect of computing backwards reachable sets is ensuring invariance. Once each set contains the previous one, the region of attraction is invariant, guaranteeing the recursive feasibility of the controller. This characteristic provides a significant advantage compared to the method introduced by Limon et al., which does not account for invariance sets due to the approximations made during the backward computation.

To obtain the mathematical formulation of the initial conditions of the system that converges to the terminal set, it is necessary to incorporate the computation of the backward reachable sets from the mRPI as

$$\mathbb{X}_f = A\mathcal{X}_j + B\mathbb{U} \;\; \Rightarrow \;\; \mathcal{X}_j = A^{-1}\left(\mathbb{X}_f - B\mathbb{U}\right), \tag{4.18}$$

where $\mathbb{X}_f$ represents the mRPI and $\mathcal{X}_j := \mathcal{X}_1$ is the set that in a single discrete-time step reaches the mRPI considering the worst-case for the disturbances that can impact the system. This process is iterative, wherein each subsequent step, $\mathbb{X}_f := \mathcal{X}_1$ and $\mathcal{X}_j := \mathcal{X}_2$, and so on. It continues until $\mathcal{X}_j$ becomes the region of attraction.

Moreover, it is noteworthy that (4.18) does not account for disturbances as they are considered in the backward propagation process initiating from the mRPI (i.e., which contemplates disturbances).

The subtraction operation on the right side of (4.18) is a Pontryagin Difference of two sets, and it is also commonly referred to as the Minkowski difference.

Definition 4.2.3 (Pontryagin Difference, [35])**.** Given two sets $\mathbb{Z}_1,\ \mathbb{Z}_2 \in \mathbb{R}^n$, the Pontryagin difference $\mathbb{Z}_d = \mathbb{Z}_1 \ominus \mathbb{Z}_2$ is defined as

$$\mathbb{Z}_d = \{z \in \mathbb{R}^n | z \oplus \mathbb{Z}_2 \subseteq \mathbb{Z}_1\}. \tag{4.19}$$

Algorithm 2 is deployed for CCGs, based on the technique by Raghuraman and Koeln. Ultimately,

the final set appended to the sequence is the region of attraction, which defines the set encompassing all states after a specified number of steps K (i.e., the number of steps is larger than the prediction horizon) converge to the mRPI.

Algorithm 2 Sequence of backwards reachable sets.

Input: mRPI
Output: Sequence of reachable sets

1: Define the number of steps K to converge to the mRPI
2: $\mathcal{X}_j \leftarrow$ mRPI
3: **for** $i = 1 : K$ **do**
4: Compute the set estimation: $\hat{\mathcal{X}}_j \leftarrow A^{-1}\mathcal{X}_j \oplus \left(-A^{-1}B\right)\mathbb{U}$
5: Compute the intersection between $\hat{\mathcal{X}}_j$ and $\mathbb{X}$: $\mathcal{X}_j \leftarrow \hat{\mathcal{X}}_j \cap \mathbb{X}$
6: Add $\mathcal{X}_j$ to the sequence
7: **end for**

When conducting a Monte Carlo simulation to test the model, it is essential to consider different potential scenarios the vehicle might encounter. The vehicle is positioned at the system's boundaries at launch to confirm its optimal performance under extreme conditions. This positioning means that the initial condition $\boldsymbol{x}(0)$ lies precisely on the boundary of the region of attraction $\boldsymbol{x}_0 \in \partial\mathcal{X}_j$. In this case, it requires maximum actuation in some directions. As a result, the MPC's alternatives are limited. If it reaches the mRPI, it guarantees the system adheres to the intended trajectory for any initial condition, underscoring the effectiveness of the regions of attraction's calculation. Otherwise, if $\boldsymbol{x}(0) \in \mathcal{X}_j$ (i.e., not in the boundary), the solver has more options.

The following methodology is designed to address the numerical challenges posed by the solver, which enhances the effectiveness of the testing process at the set's boundary to exclude states beyond the region of attraction.

To determine the initial condition of the controller, the optimal variable is ξ within the ξ space (i.e., the set of generators). This space is subject to inherent numerical errors, which propagate to the corresponding state within the region of attraction. Consequently, it is imperative to consider error propagation directly from the initial stage of optimising ξ. Figure 4.2 is depicted to present this idea more clearly. For a comprehensive understanding of the procedure, the definition of CCGs (Definition 3.2.2) should be revisited.

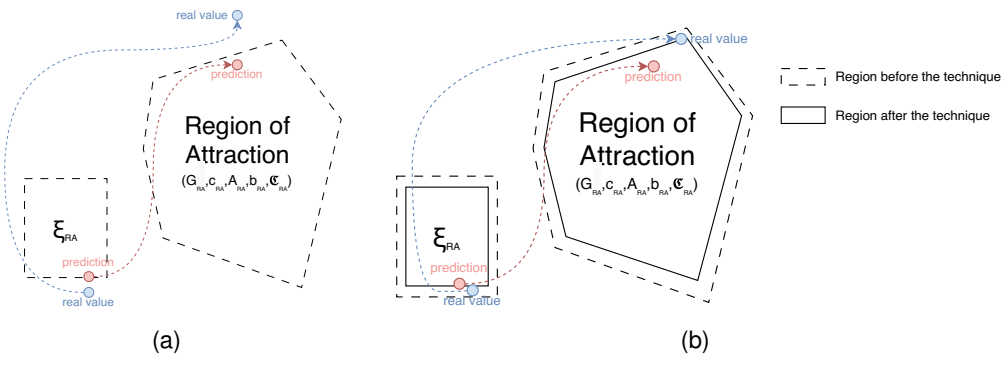


Figure 4.2: The ξ space of the region of attraction and the set itself, along with the corresponding error propagation between the prediction and the actual value obtained by the solver. (a) Without preventing numerical errors. (b) Preventing numerical errors.

To provide a more detailed explanation of the concept portrayed in Figure 4.2a, it is essential to understand that when the optimiser seeks the optimal ξ value, it may initially seem as though it is selecting the red value. Nevertheless, in reality, the solver chooses the blue value. Consequently, when this selected value is matched with the tuple $(G_{RA}, c_{RA}, A_{RA}, b_{RA}, \mathfrak{C}_{RA})$, the state should ideally align with the red state, but it corresponds to the blue one, located outside of the region of attraction. This discrepancy results in an infeasible initial condition.

The concept underlying the process is elucidated in Figure 4.2b. This illustrative representation demonstrates that upon implementing the technique, the actual value no longer resides within the newly defined region but remains within the original domain, ensuring the feasibility of the initial conditions.

In the context of CCGs formulation, a more straightforward strategy involves the adjustment of the matrix of generators. As a result, it is more intuitive to scale all the entries by a factor rather than performing other operations. Thus, two critical aspects are considered when reducing the set to make it more robust: the error ϵ that is decreased around it and the set's maximum distance d (i.e., distance from the centre to the furthest point).

Firstly, a hypercube is formed with a side length equivalent to the maximum error tolerated by the solver. This strategy establishes a space encompassing all the discrepancies introduced by the solver. However, to incorporate this deviation accounting for its correlation with the set undergoing the technique, a linear map occurs between this space and G_{RA}. Consequently, a set that systematically takes the propagation of errors when applied to the region of attraction is derived. Moreover, a bounding box is employed to contain the outcome of the linear mapping, allowing for an estimation of the maximum extent of the set. This estimated value corresponds to ϵ, the intended importance to be decreased around the set.

The generator matrix of the set under consideration (i.e., the region of attraction) is analysed to identify the distance from the centre to the furthest point in each set's parameter. This assessment facilitates the calculation of the vector d, which encompasses distinct maximum distances for position, velocity, and acceleration. The vector d is calculated as

$$d = |G_{\mathsf{RA}}(a : b, :)|_{\infty}, \tag{4.20}$$

where a and b represent the limits of each set's parameter (i.e., position, velocity, and acceleration). After taking all of this into account, the factor that is multiplied by each entry of G is

$$\frac{d - \epsilon}{d} = 1 - \frac{\epsilon}{d}. \tag{4.21}$$

Algorithm 3 is presented to provide an overview of the described method.

Algorithm 3 A method for preventing numerical errors when selecting the initial condition within the region of attraction.

Input: Region of Attraction: Set RA
Output: Smaller Region of Attraction: Set RRA

Define the maximum error of the solver $\mathsf{E}_{\max}$
Create the set of errors: $\mathbb{E} \leftarrow (\mathsf{E}_{\max} \cdot I_{n_x}, 0_{n_x}, 0_{n_x}, 0_1, \mathcal{B}_\infty)$
Create a box set $\mathbb{B}$ around the linear map $G_{\mathsf{RA}} \cdot \mathbb{E}$
Compute the value to be decreased around the set: $\epsilon \leftarrow \max(\max(G_{\mathbb{B}}))$
idx $\leftarrow 1$
for $i = 1 : n_u : n_x$ **do**
    for $j = 0 : (n_u - 1)$ **do**
        Compute the distance for each parameter: $d(i+j) = |G_{\mathsf{RA}}\,(i : (\mathsf{idx} \cdot n_u), :)|_\infty$
    end for
    idx $\leftarrow$ idx $+ 1$
end for
for $i = 1 :$ size$(G_{\mathsf{RA}}, 1)$ **do**
    for $j = 1 :$ size$(G_{\mathsf{RA}}, 2)$ **do**
        Compute the new generator's matrix: $G_{\mathsf{RRA}}(i,j) = G_{\mathsf{RA}}(i,j) \cdot \left(1 - \frac{\epsilon}{d(i)}\right)$
    end for
end for

4.3 Adaptive Model Predictive Control

This section outlines the methodology used to control a drone's translation to force it to enter into a specific set considering the vehicle and the environment's boundaries. The proposed solution relies on the computation of the set containing all the states once the pre-defined steps converge to the mRPI, with these steps extending beyond the prediction horizon.

The MPC cost function is quadratic, chosen for its favourable characteristics as outlined earlier in Section 3.1. Additionally, the optimisation problem resembles the one introduced in [26], primarily due to the terminal constraint within the sequence of $N_c \cdot N$ reachable sets, where N denotes the prediction horizon and N_c the number of convergence steps. Convergence steps simplify the idea of the proposed MPC by counting the times the algorithm runs over N to reach mRPI. The optimisation problem is established as

$$
\begin{aligned}
\min_{\boldsymbol{u}(0),\ldots,\boldsymbol{u}(N-1)} \; & \boldsymbol{x}(N)^\top P \boldsymbol{x}(N) + \sum_{i=0}^{N-1} \left(\boldsymbol{x}(i)^\top Q \boldsymbol{x}(i) + \boldsymbol{u}(i)^\top R \boldsymbol{u}(i) \right) \\
\text{s.t. } & \boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) + \boldsymbol{d}(k), \text{ for } k = 0, 1, \ldots, N-1, \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0, \\
& \boldsymbol{x}(k) \in \mathbb{X}, \text{ for } k = 0, 1, \ldots, N-1, \\
& \boldsymbol{u}(k) \in \mathbb{U}, \text{ for } k = 0, 1, \ldots, N-1, \\
& \boldsymbol{d}(k) \in \mathbb{D}, \text{ for } k = 0, 1, \ldots, N-1, \\
& \boldsymbol{x}(N) \in \mathcal{X}_j \subseteq \mathbb{X}, j = \max\left((N_c - 1) \cdot N - k, 0\right).
\end{aligned}
\tag{4.22}
$$

In each discrete-time step, the terminal desired region is updated to tend the system to the mRPI.

Including this terminal constraint guarantees the placement of the terminal state within the subsequent computed reachable set [26]. Figure 4.3 is depicted to illustrate the methodology implemented.

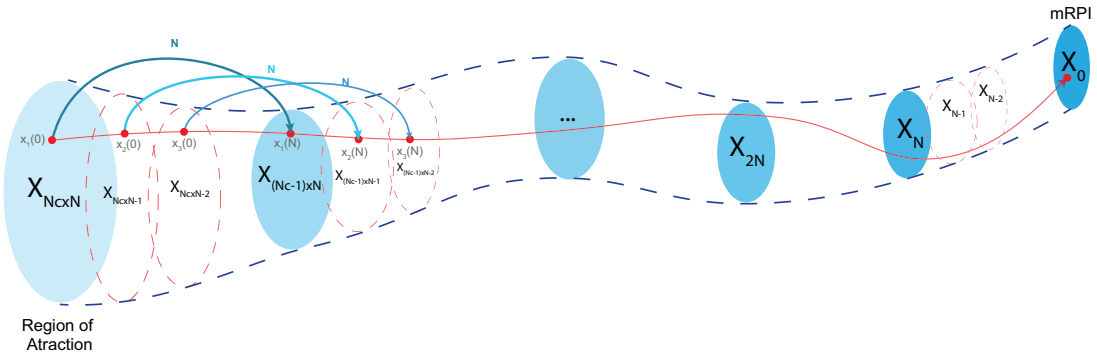


Figure 4.3: Illustration of the adapted MPC process with $x_{k+1}(0) \in \mathcal{X}_{N_c \cdot N-k}$ then $\mathbb{X}_f := \mathcal{X}_j$, $j = \max\left((N_c - 1) \cdot N - k, 0\right)$.

The simulation is conducted using YALMIP [38]. One of YALMIP's key advantages is the ability to pre-compile controllers with pre-defined cost functions and constraints using the `optimizer` [43] command. The implementation relies on external solvers MOSEK [44] and GUROBI [45] in MATLAB Hybrid Toolbox. The solver's choice is automated and relies on the problem's structure and properties. More precisely, MOSEK is employed for the controller (e.g., generating the optimal sequence of control inputs), and GUROBI is used for set searches (e.g., identifying initial conditions for testing the controller in a Monte Carlo simulation, verifying at each step if the system reaches the terminal set).

4.4 Obstacle avoidance

Obstacle avoidance is a critical challenge for autonomous vehicles, ensuring safe and effective operation in complex and dynamic environments. These systems must perceive and react to obstacles in their path, preventing collisions and enabling intelligent decision-making to adapt to changing surroundings. Evaluating the effectiveness of MPC-based trajectories in obstacle avoidance provides real-world insights, validating the approach's efficacy and enhancing the reliability and safety of autonomous systems.

To incorporate obstacles in the MPC trajectory, it is imperative to ensure the convexity of each set, confirming that it remains in CCGs formulation. Highlighting that constructing a set with voids is not feasible, it becomes evident that the region where the obstacle is introduced undergoes modifications to ensure convexity. Introducing an obstacle at a particular set k on the sequence implies the intersection operation of both sets (the obstacle's set and the corresponding set at index k). Therefore, all subsequent sets up to the mRPI necessitate adjustments considering a forward method, presented in Algorithm 4.

Algorithm 4 Sequence of forward reachable sets considering obstacles.

Input: Obstacle set $\mathbb{O}$ and the sequence of backward reachable sets $\mathcal{X}_j,\ j = 0, \ldots,\ j = 0, \ldots, K$
Output: Sequence of forward reachable sets considering obstacles $\mathcal{O}_j,\ j = 0, \ldots,\ j = 0, \ldots, K$

```
Define the index k where the obstacle is introduced
for the set i being the region of attraction to the mRPI do
    if the set is previous to the set where the obstacle is introduced then
        The set is equal to the one at the backward sequence: O_i ← X_i
    else if the set is where the obstacle is introduced (i.e., the index is k) then
        Compute the intersection between X_i and 𝕆: O_i ← X_i ∩ 𝕆
    else if the set is subsequent to the set where the obstacle is introduced then
        Compute the set estimation: Ô_i ← A O_i ⊕ B𝕌
        Compute the intersection between Ô_i and X_i : O_i ← Ô_i ∩ X_i
    end if
    Add O_i to the forward sequence
end for
```

When facing an obstacle, it restricts the range of possible states, forming a subset of the previous set, essentially narrowing down the allowable space. To illustrate this concept, consider a car on a road. The road's limits at each moment define the set's boundaries. When an obstacle appears on one side of the road, it does not create new limits but restricts the available space. In the context of this approach, the intersections following the obstacle introduction involve the corresponding sets from the backward sequence, encompassing all potential states at each instant.

The offline computation of the sequence serves the primary purpose of reducing computational complexity. Additionally, it addresses a crucial scenario where the MPC may confront unexpected obstacles before predicting their presence N steps ahead. If these computations are performed in real-time during MPC execution, the navigation around unanticipated obstacles is a challenge. Consequently, the MPC remains analogous to the one described in Section 4.3, with the difference being the sequence provided to the controller.

Chapter 5

Simulation Results and Discussion

The codebase and resources related to the research presented in this thesis can be found in the GitHub Repository [46].

This chapter presents the results of this research and discusses their implications. It shows the advantages of using CCGs to represent the sets under study regarding accuracy and computational time. Additionally, it demonstrates the implications of different RPI methods, the conservatism associated with the region of attraction and the results of the proposed approach to mitigate numerical issues arising from the solver. Subsequently, it delves into the domain of control using the MPC regulator, offering insights into its performance across different trajectories and considering obstacles.

The quadrotor model implemented to present the following results is the DJI Phantom 4 Pro depicted in Figure 5.1.

Figure 5.1: DJI Phantom 4 Pro V2.0. [36].

5.1 Admissible Sets Representation

As clarified in Section 4.2.1, the precise definition of the vehicle's parameters (i.e., position, velocity, and acceleration) plays a crucial role in the controller's operation to further define the robust controlled invariant sets in the sequence. The preference of representing the sets using CCGs is based on the advantages discussed previously.

Firstly, it is necessary to establish the boundaries for each parameter to provide the initial framework for the proposed algorithm. These limits are determined using the model presented in Figure 5.1 and outlined specifications in [47]. Based on the assumptions summarised in Section 4.2.1, it is possible to define the set of states $\mathbb{X}$, as shown in Figure 5.2.

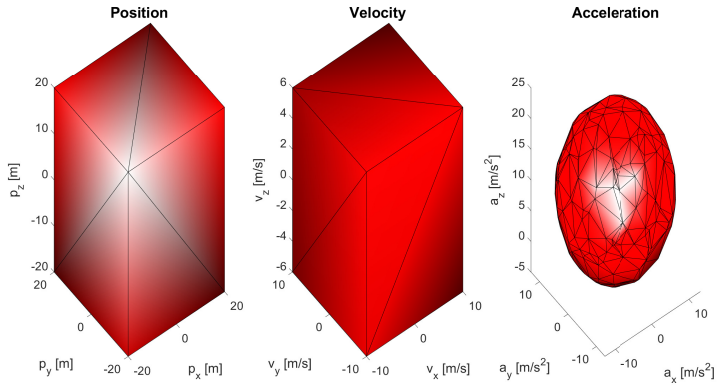


Figure 5.2: Projection of the states set $\mathbb{X}$ for DJI Phantom System.

The state set $\mathbb{X}$ exhibits a distinct geometric structure due to the employment of CCGs. Notably, the position and velocity are hypercubes, but the acceleration component takes the form of a sphere, established in (4.5).

As proven in (4.9), the configuration of $\mathbb{U}$ takes the form of a sphere illustrated in Figure 5.3.

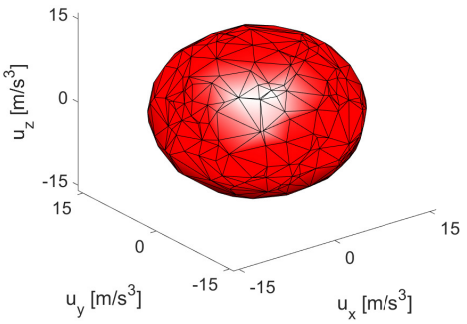


Figure 5.3: Control Input set $\mathbb{U}$ for DJI Phantom System.

While the theoretical ideal portrays the acceleration and the control input as a sphere, an approximation undertaken by da Silva Pinto et al. departs from this ideal, employing a polyhedron representation. Nonetheless, it does not accurately depict these sets, potentially compromising the vehicle's performance. This observation supports the reduction of conservatism when representing sets using CCGs.

As stated earlier, the disturbance set $\mathbb{D}$ is constructed across all the dimensions; however, its impact is limited to the control input of the system, specifically influencing the vehicle's acceleration. Figure 5.4 displays the set $\mathbb{D}$.

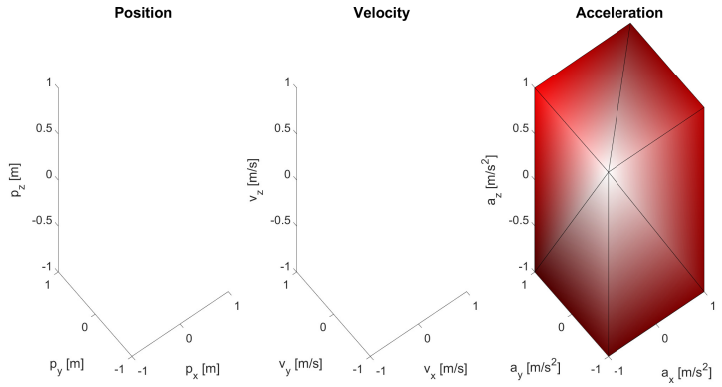


Figure 5.4: Projection of the disturbances set $\mathbb{D}$ for DJI Phantom System.

5.2 Computation of the minimal RPI

The mRPI representation describes the feasible region where system states converge and stay invariant despite disturbances. An analysis of the different methodologies described in Section 4.2.2 to calculate the mRPI is presented in Table 5.1, along with a visual representation in Figure 5.5.

The computation of F_∞ is based on (4.16) and given by

$$\bigoplus_{i=0}^{K} \mathcal{A}^i \, \mathbb{D}, \tag{5.1}$$

with $K \gg N$. The set denoted as inner RPI is computed through (5.1) with $K = N$, corresponding to the left side of (4.17). F_s is calculated via (4.17), which is the mRPI of the proposed solution. Lastly, it is worth noting that the selection of K in (5.1) is determined through various simulations using different K values until there are no observable differences in the resulting set.

Table 5.1: Number of generators and computational time for different mRPI computation methods.

	F_s	Inner RPI	F_∞
Nº of Generators	39	36	30 003
Computational Time [s]	3.04×10^{-3}	2.26×10^{-3}	3.98

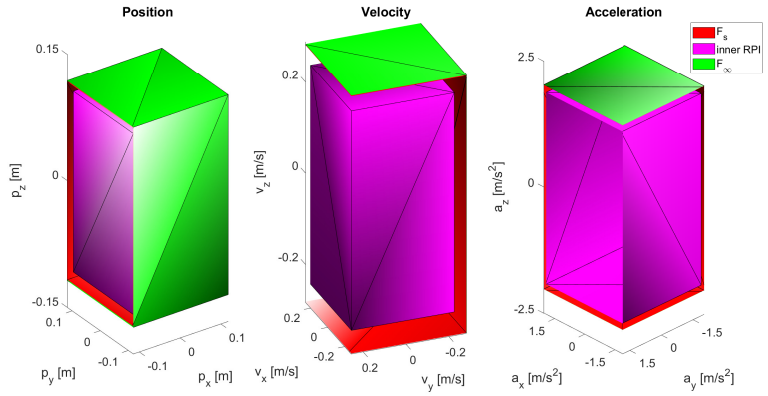


Figure 5.5: Projection of different RPI computation methods.

In Figure 5.5, it is evident that the inner RPI effectively serves as an inner approximation of the mRPI. While this set offers optimal computational efficiency, it lacks precision in the representation of the mRPI. On the other hand, the set F_s offers an outstanding approximation of the F_∞. This set has fewer generators (39 instead of $30\ 003$) and maintains excellent computational performance. In this manner, the proposed method to compute the mRPI has remarkable computational efficiency whilst keeping accuracy.

The parameters for landing a drone in terms of position, velocity, and acceleration are determined by the specific requirements, the capabilities of the drone, and the surrounding environment. The preferred landing position is usually near the ground, ideally at a designated landing pad or target region. The value depends on the drone's height, the altitude of the landing pad, and any obstacles in the landing space. The drone's descent seeks a gentle landing by carefully regulating its vertical velocity towards zero. To avoid collisions, the vertical velocity of the drone is gradually slowed as it approaches the target landing place. Acceleration control is often employed indirectly through velocity control and rarely applied directly during landing. The mRPI for the system under study is portrayed in Figure 5.6 (corresponds to F_s in Figure 5.5).

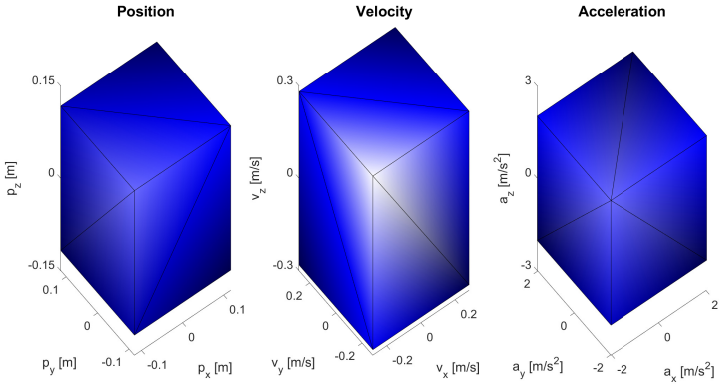


Figure 5.6: Projection of the mRPI for DJI Phantom System.

The successful computation and visualisation of the mRPI reaffirm its effectiveness in designing stable systems. The set depicted in Figure 5.6 aligns with the specific specifications and constraints of

the drone system under consideration. By incorporating the system parameters, dynamics, and convex set representing disturbances, the mRPI characterise the feasible region where the drone can easily switch to the LQR controller and remain there. This tailored technique ensures stability and respects the drone's physical limits and operating boundaries.

5.3 Conservatism of Region of Attraction

Computing the region of attraction with the highest precision is critical to verify the stability and feasibility of control systems. This section underscores the impact of the mRPI computation on the region of attraction and compares two distinct approaches to its computation. The primary objective is to investigate and evaluate the advantages and disadvantages of each technique.

The number of generators and computational time needed to calculate the region of attraction is examined through backward propagation starting from the sets F_s and F_∞ (i.e., calculated with (5.1) with $K \gg N$). The results are displayed in Table 5.2, highlighting the computational efficiency of the proposed method to represent the region of attraction with fewer generators considering the mRPI definition via F_s.

Table 5.2: Number of generators and computational time for the region of attraction computation using different mRPI sets for backwards propagation.

	F_s	F_∞
Nº of Generators	183	$30\ 147$
Computational Time [s]	1.12×10^{-2}	4.27

Figure 5.7 depicts the computation of the region of attraction using CCGs and CZs to define the control input set $\mathbb{U}$ with a narrow horizon (i.e., $N = 4$) in 2-Dimensions (2D). The strategy delivered by da Silva Pinto et al. to determine the region of attraction tends to over-approximate the ℓ_2-norm balls by a polytope. When doing the backwards propagation, it increases the possibility of selecting an initial condition that does not converge to the terminal set. This case is represented in Figure 5.7a. However, when the control input set is represented by ℓ_∞-norm ball, the computation of the region of attraction with CCGs or CZ coincides, displayed in Figure 5.7b.

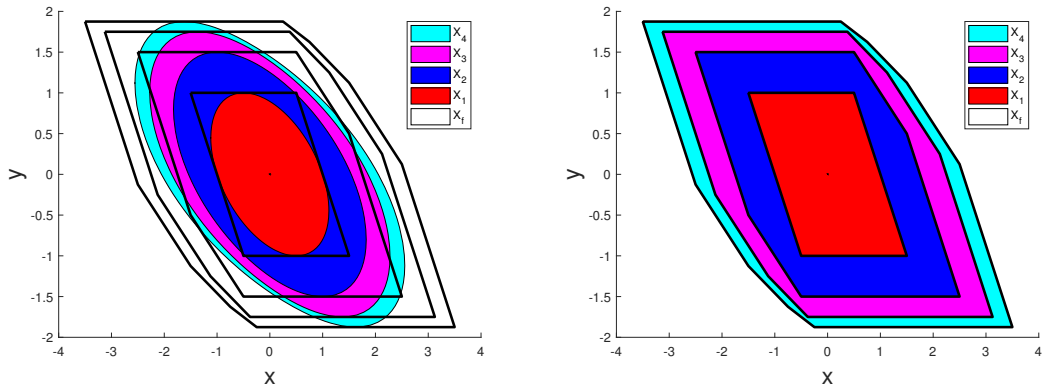


Figure 5.7: Region of attraction computed with CCGs (coloured filled sets) and CZs (black outline sets) with the control input set $\mathbb{U}$ described as (a) ℓ_2-norm. (b) ℓ_∞-norm.

From Figure 5.7a, it is observable that the $\mathcal{X}_1$ is over-approximated when computed with CZs (the nearest polytope representation), and then the error is propagated for the following sets, especially in the x-axis given the system model $(A = [2, 1; 0, 2], B = [1, 0; 0, 1])$. When $\mathbb{U}$ is described as ℓ_2-norm, the strategy of da Silva Pinto et al., in a Monte Carlo simulation, has approximately 32% of states as infeasible initial points. Confirming the capacity of CCGs to depict sets with reduced conservatism.

Moreover, the focus shifts to the practical application of this comparative analysis within the quadrotor model. This case is represented in Figure 5.8, showcasing the principal differences between both procedures.

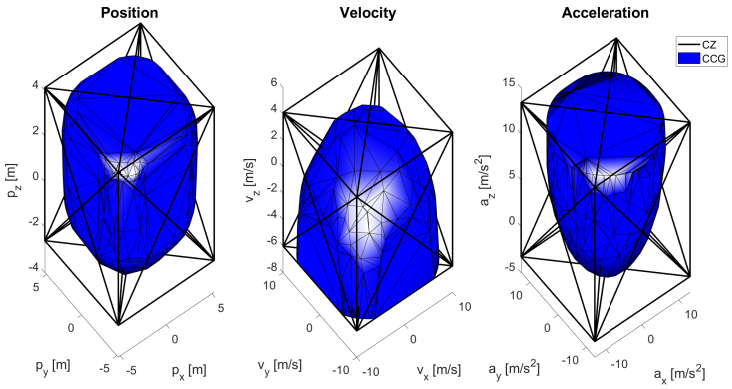


Figure 5.8: Comparison of the projected region of attraction when computed with a sequence of CCGs and CZs for DJI Phantom System.

The reduction of conservatism provided by the developed method confers a notable advantage as it facilitates the precise description of ℓ_2-norm sets, which holds particular significance when considering the maximum thrust rotating propellers can produce, as detailed in Section 4.2.1 and proven in Section 5.1.

The linear velocity dynamics (4.5) yields a set in the shape of a sphere for acceleration portrayed in Figure 5.2. Also, the control input (4.9) illustrated in Figure 5.3 is a sphere. By accurately representing these ℓ_2-norm sets, the CCG representation ensures precise and reliable control over the vehicle's

motion. Furthermore, in a Monte Carlo simulation considering the region of attraction defined with CZ formulation, approximately 23% of states are infeasible. This proportion can considerably impair the controller's performance, emphasising the need for a more precise and robust representation of the region of attraction. These results validate the CCG formulation to represent these sets.

5.4 Preventing Numerical Errors

The forthcoming section delves into the outcomes of the technique presented in Algorithm 3. By harnessing this methodology, the search for initial conditions beyond the region of attraction is avoided. For each parameter (i.e., position, velocity, and acceleration), the difference between the previous set and the current set varies.

The results are demonstrated through a 2D simulation with a triple integrator in Figure 5.9. The frame encircles the region of attraction, illustrating the area before the process (in cyan). Notably, its size is minimal, considering the solver's error magnitude of 10^{-6} concerning ξ.

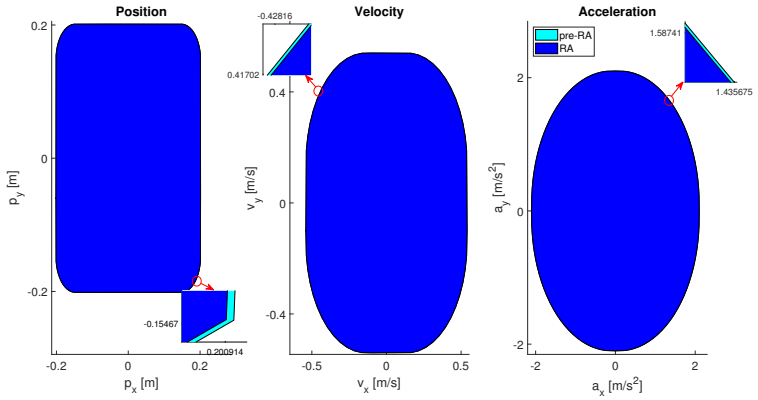


Figure 5.9: Technique to reduce numerical errors applied to the projected region of attraction in x and y directions.

In Figure 5.10, the same technique is applied in this dissertation's case employing the quadrotor.

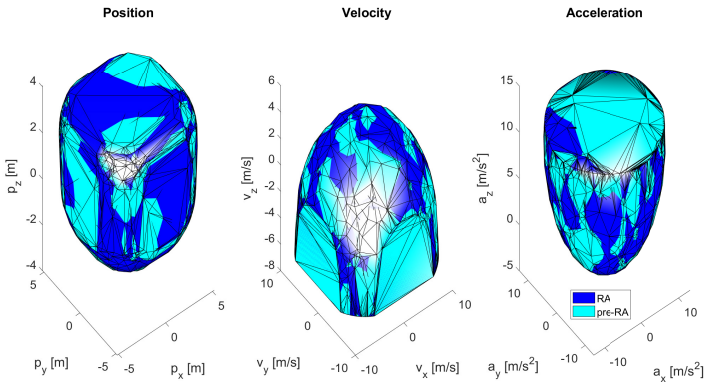


Figure 5.10: Technique to reduce numerical errors applied to the projected region of attraction for DJI Phantom System.

Because of the non-uniform shape of this region of attraction, the process does not follow a linear pattern. One possible cause is that certain irregularities cannot be accommodated within the bounds of the prior region of attraction, resulting in some minor imperfections.

When working with predefined solvers in the MATLAB toolbox, it is common for numerical issues to emerge, making it challenging to eliminate them. However, this strategy aims to mitigate these issues to the greatest extent possible. Performing a Monte Carlo simulation with 500 runs for the quadrotor results in a substantial decrease in numerical issues during the MPC process. Specifically, it decreases the percentage of issues from 4.29% to a significantly improved 0.2%.

5.5 MPC Regulator

To explain the controller's behaviour, the following simulation focuses on the x and y directions depicted in 2D and 3D (with discrete-time instants indicated on the x-axis) for different perspectives. The 2D representation is displayed in Figure 5.11, and the 3D is provided in Figure 5.12. The disturbances only manifest in acceleration to illustrate the context explored in this thesis.

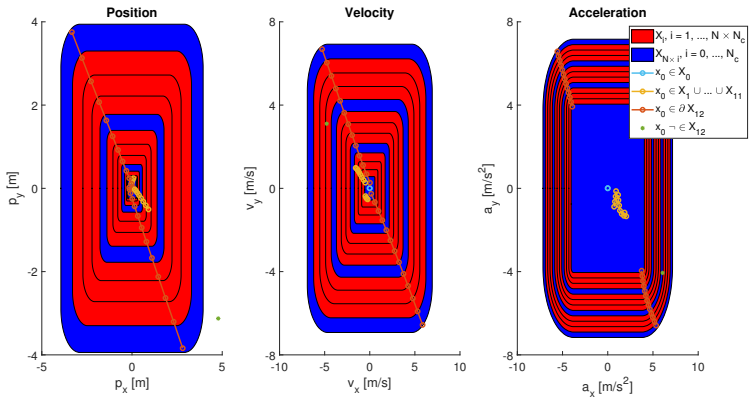


Figure 5.11: Projection of sequence sets $\mathcal{X}_i,\ i = 0, \ldots, N_c \times N$ in x and y directions. The MPC trajectories from different initial conditions.

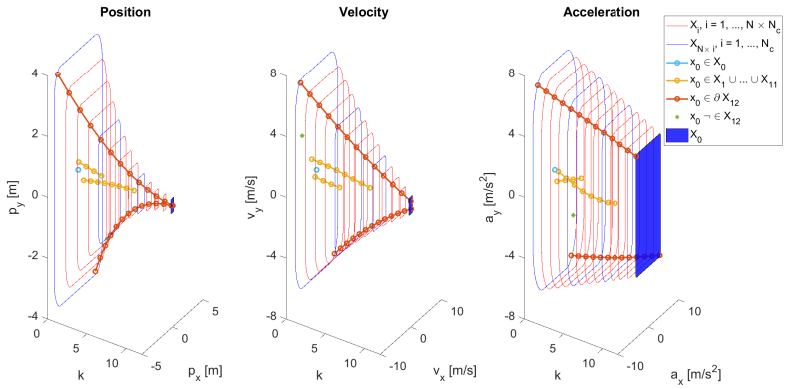


Figure 5.12: Projection of sequence sets $\mathcal{X}_i,\ i = 0, \ldots, N_c \times N$ in x and y directions over discrete-time instants. The MPC trajectories from different initial conditions.

Several observations are derived from these visual representations:

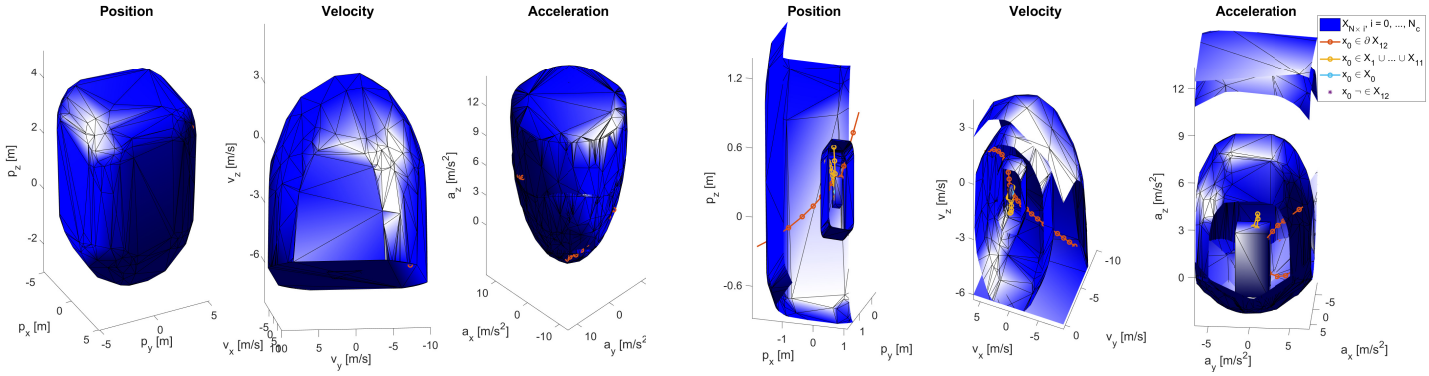
- The blue trajectories represent cases where the initial condition belongs to the mRPI, denoted as $\mathcal{X}_0$. Consequently, these trajectories consist of only one state.

- The yellow trajectories illustrate scenarios in which the initial condition falls between the penultimate set $\mathcal{X}_1$ and the second set $\mathcal{X}_{11}$. From the scenario where the initial condition resides within the mRPI in some parameter (e.g., position, velocity and acceleration), the trajectory tends to progress towards the mRPI in the dimensions that have yet to be reached. Thus, these yellow trajectories demonstrate that when the initial condition is further from the mRPI, it necessitates more discrete-time instants to reach the mRPI.

- To test the MPC under maximum circumstances, the initial conditions are placed on the boundary of the region of attraction. These trajectories, depicted in orange, indicate that the MPC operates at its limits and requires maximum actuation in particular directions, as detailed in Section 4.2.3. Notably, when the initial state is positioned at the edge of the region of attraction, the final condition also lies along the boundary of the mRPI, underscoring the calculation's accuracy.

- When an initial condition lies outside the region of attraction, the MPC is infeasible. In such cases, the initial condition is represented by an asterisk.

The visual representation in Figure 5.13 illustrates the MPC trajectories within the DJI Phantom System scenario. Only the sets $\mathcal{X}_{i\times N}$, $i = 0, \ldots, N_c$ are displayed to simplify trajectory visualisation. Figure 5.13a is included to confirm the start of particular trajectories from the boundary of the region of attraction. Figure 5.13b showcases a selection of trajectories derived from a Monte Carlo simulation, converging towards the mRPI (depicted as the smaller blue set).



(a) (b)

Figure 5.13: Projection of sequence sets $\mathcal{X}_{i\times N}$, $i = 0, \ldots, N_c$ in x, y and z directions for DJI Phantom System. The MPC trajectories from different initial conditions. (a) Zoom-out. (b) Zoom-in.

The controller's performance reflects the previous observations for the x and y directions (Figure 5.11 and 5.12). It reaffirms the accuracy and robustness of the proposed MPC.

Finally, during a simulation employing the region of attraction with $30\ 147$ generators (i.e., regarding Table 5.2), it requires over one hour for pre-compiling the optimisation problem before further integration to execute the MPC. In contrast, when performing with the proposed mRPI, it takes approximately 1.39

s to pre-compile the problem and 1.41 s to generate the MPC trajectory. It underscores the computational impact of performing with an mRPI with few generators on the MPC process, confirming that the computational times are suitable for a real-time application.

5.6 Obstacles on the Trajectory

The controller's performance depends on different factors. A series of simulations are conducted to assess the result of these factors, followed by a meticulous analysis.

Considering a simulation with $\mathcal{X}_i,\ i = 0, \ldots, 12$ sets, with $N = 4$ and $N_c = 3$. The obstacle is introduced into the trajectory at set $\mathcal{X}_8$. This obstacle remains stationary, primarily affecting the position. To illustrate the formulation discussed in Section 4.4, Figure 5.14 is presented. Depicting the differences within this set, both before $\mathcal{X}_8$ and after the obstacle $\mathcal{O}_8$.

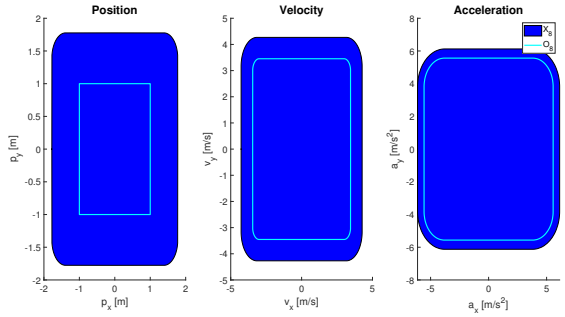


Figure 5.14: Projection of set $\mathcal{X}_8$ in each direction, before and after the obstacle's introduction.

In Figure 5.15, the impact among all the sets within the sequence before and after the obstacle is presented. The various trajectories along this sequence of sets with the obstacle are highlighted, each initiated from different initial conditions.

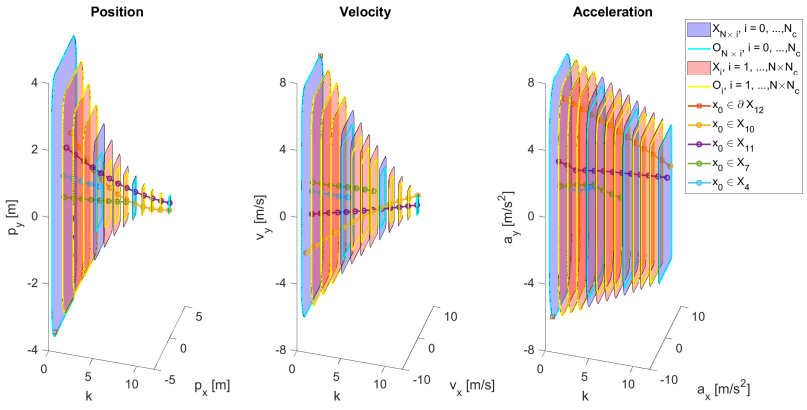


Figure 5.15: Projection of sequence sets in x and y directions over discrete-time instants and the MPC trajectories from different initial conditions. The obstacle is situated in $\mathcal{X}_8$, leading to the creation of the resulting sequence $\mathcal{O}_i,\ i = 0, ..., N_c \times N$, with $N = 4$.

Several observations are derived from the results in Figure 5.15:

- Due to the intersection of the obstacle set and $\mathcal{X}_8$, when the obstacle assumes a more extensive set, the potential manoeuvring space for circumventing the object increases.

- As mentioned earlier, when considering other factors, it becomes evident that the initial state significantly influences whether the solution is feasible or infeasible. Considering that the initial condition lies in the sets before the obstacle ($\mathcal{X}_{12}, \ldots, \mathcal{X}_7$):

  - Initial conditions located at the boundary of $\mathcal{X}_{12}$ already demand maximum actuation to reach the edge of each terminal set in the sequence. Consequently, when considering the introduction of obstacles, the MPC immediately determines infeasibility, as the last state of the prediction $\boldsymbol{x}(N)$ cannot reside within $\mathcal{O}_8$. This example is illustrated with the orange square.

  - The initial condition is positioned in a manner that allows reaching the new mRPI. This case is represented in yellow and purple trajectories.

- The behaviour differs when the initial condition lies in the sets after the obstacle ($\mathcal{X}_7, \ldots, \mathcal{X}_0$):

  - The MPC fails to adapt to the new sequence of sets due to the location of the initial condition near the boundary. Similar performance to the one at the boundary of the region of attraction. It is noteworthy to realise that this scenario is far less common.

  - The state encounters no obstacle, allowing the trajectory to progress as expected. These scenarios are depicted in blue and green.

Highlighting some of the previously discussed insights, a simulation featuring the obstacle set results in Figure 5.16 represented in Figure 5.17. The prediction horizon is set to $N = 3$ to illustrate the effect of this factor. Additionally, to test the obstacle's impact at a different point in the trajectory, it is placed within $\mathcal{X}_4$.

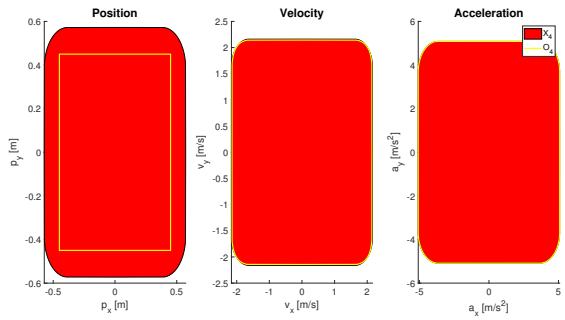


Figure 5.16: Projection of set $\mathcal{X}_4$ in each direction, before and after the obstacle's introduction.

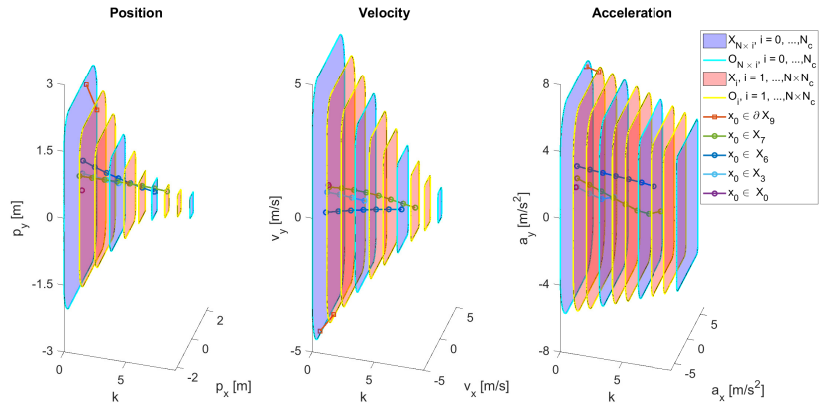


Figure 5.17: Projection of sequence sets in x and y directions over discrete-time instants and the MPC trajectories from different initial conditions. The obstacle is situated in $\mathcal{X}_4$, leading to the creation of the resulting sequence $\mathcal{O}_i,\ i = 0, ..., N_c \times N$, with $N = 3$.

The obstacle's location along the trajectory may significantly affect the performance. An obstacle at the beginning of the trajectory might allow more instants to adjust the course, while an obstacle closer to the end could have more impact.

Furthermore, the trajectories started in the boundary of the region of attraction terminated prematurely after just three states. It occurs because $x_0 \in \mathcal{X}_9$ (region of attraction) and at the third MPC it means $\mathbb{X}_f := \mathcal{O}_5$, which coincides with the obstacle location. This case is illustrated in the orange trajectory. The dark blue and green examples represent initial conditions placed within sets where the obstacle does not affect the trajectory. Finally, the light blue and purple example features a trajectory that does not encounter the obstacle. In the purple case, the initial condition is already within the mRPI.

In conclusion, obstacle avoidance is a crucial aspect of autonomous system control. This section has provided insights into the diverse behaviours of the MPC controller when confronted with obstacles. These considerations are summarised as follows:

- **Obstacle Location**: Placing an obstacle at the beginning of the trajectory (i.e., after a minimum of N steps to guarantee that MPC can detect) tends to yield better performance.
- **Obstacle set size**: When the obstacle set occupies a larger area, it expands the available space for manoeuvring around it. Thus, the likelihood of encountering infeasibility is greater with less extensive sets of obstacles.
- **Initial condition**:
  - When lies on the boundary of the region of attraction, no control action can be applied to *manoeuvre* around the obstacle, thus rendering the problem infeasibility. A Monte Carlo simulation with 500 runs yields 100% infeasible trajectories.
  - Specific conditions outside the centre of the region of attraction can result in infeasibility, as the trajectory's terminal set (for that prediction) fails to align with the newly introduced set.
  - Conversely, if it is close to the mRPI, the state might reach it without encountering the obstacle.

  - For the ones that are neither close to the mRPI nor on the boundary of the region of attraction, the problem's feasibility depends on specific circumstances.

- **Prediction Horizon**: The size of the control horizon influences performance. A shorter horizon may restrict the controller's capacity to navigate obstacles, reflecting poorer performance. However, it is noteworthy to acknowledge the existence of a trade-off, as significantly increasing the control horizon can lead to heightened computational complexity.

The same behaviour is observed when considering the drone case, although perception becomes more challenging due to the three-dimensional character of the problem. The analysis and observations align with the previously discussed for the two-dimensional case.

Chapter 6

Conclusion

6.1 Conclusions

The dissertation's goal was to analyse the stability of the predictive controllers by the innovative application of CCGs as the primary method for characterising the reachable sets. The CCGs have revealed the capability to reduce the conservatism traditionally linked with set representations whilst improving computational efficiency. Underscoring the practicality of such set-based methodologies to estimate the sets in quadrotor control to enhance feasibility, ultimately strengthening controller robustness.

A comprehensive investigation is conducted to analyse the accuracy and reduced conservatism achieved with CCGs. This study compares CCGs with CZs, which tend to overestimate the actual set, resulting in more extensive and less precise representations. This divergence in conservatism becomes particularly noted when searching for initial conditions. CZs often yield more infeasible solutions due to their overestimation of the feasible region. Therefore, the region of attraction represented through CZ for the quadrotor model encompasses 23% of states that are, in reality, infeasible states. Consequently, it can result in infeasibility of the control system, posing a risk to aerospace missions. In other scenarios, it may lead to sub-optimal control decisions that might not fully exploit the system's potential.

The novel approach of integrating CCGs into the design of a predictive controller hinges on the MPC with a variable terminal condition. In this framework, each MPC iteration has a unique terminal set precisely defined by CCGs. This process forces the system to enter into an invariant set represented as CCG, which reduces the related conservatism. Moreover, The mRPI representation defines the feasible region where system states converge and remain invariant, even when subjected to disturbances. Therefore, when the system's state falls within the mRPI, it can seamlessly transition to the LQR controller and operate within this region. It confers a distinct advantage to autonomous vehicles. With this controller transition, the new controller requires fewer computational resources and maintains an adherence to the system's constraints.

For the effective operation of the MPC, an essential factor to consider is the search within the CCGs, where the number of generators plays a pivotal role in determining its performance. When this number becomes excessively high, it is challenging for the MPC to predict the control inputs (e.g., the scenario

with 30 147 generators to describe the region of attraction requires more than an hour to formulate the optimisation problem). In this case, it can lead to a situation where, over time, numerical issues or even infeasibility arise. These results underline the significance of the proposed method, which presents an optimal representation of the mRPI with fewer generators (e.g., 39 instead of 30 003) and an excellent computational time (e.g., 3.04 $\times 10^{-3}$ s instead of 3.98 s). A reduced number of generators in the mRPI facilitates the backward propagation and creates a region of attraction with fewer generators (e.g., 183 instead of 30 147).

Recognising perturbations is an inevitable facet of real-world scenarios. This dissertation delves into the theoretical exploration of their effects and outputs. The introduction of obstacles, situated in diverse locations and varying sizes, assumes central significance. Maintaining the convexity of the new estimations is imperative to facilitate precise simulations. These obstacles, treated as CCGs, ensure that the investigation remains aligned with the previous rigour and methodology employed during MPC testing in an obstacle-free environment. The results demonstrate the controller's capability to navigate around obstacles under such circumstances. However, it is important to note that if the trajectory initiates within the boundary of the region of attraction, obstacle avoidance becomes unfeasible, underscoring the precision of the region.

UAVs demand more robust control strategies to guarantee that the vehicles can effectively interact with external environments and execute complex manoeuvres. Incorporating CCGs into MPC to represent the control invariant sets significantly enhances the feasibility of the control system. It reduces conservatism and enables operations near the set boundaries. Consequently, the rising demand for robustness, adaptability, and improved performance in quadrotor control is addressed.

In summary, this research precisely defines the region of attraction for aerospace vehicles subject to different constraints and disturbances, ensuring mission feasibility from the outset. It ensures that if any initial condition is located within the region of attraction, the MPC successfully reaches the terminal set 100% of the time with a computational time suitable to a real-time application (e.g., 1.41 s to generate an MPC trajectory). The outcome of this study stands as a validation of a concept, underscoring the potential of advanced control methods in shaping the trajectory of autonomous vehicles in the future.

6.2 Future Work

The present research represents a step forward in set representation for control systems stability, yet there is ample room for future exploration and improvement.

One of the notable limitations in the application of MPC is its considerable computational demand. Overcoming this is crucial to expand the proposed solution to larger trajectories and more complex systems. A promising approach involves over-approximating the sets within each backward sequence. Moreover, the paving techniques divide these sets into smaller and more manageable subsets, facilitating efficient representation and computation, even in the presence of uncertainty or disturbances. One of the techniques employed in paving is via inner approximations, which ensures that the accuracy of the sets is preserved.

To augment the foundation of this dissertation, practical testing of the proposed theory in a real-world setting would be a valuable addition. Nonetheless, before embarking on such empirical tests, it is imperative to construct extensive simulation environments to facilitate the assessment and visualisation of aerospace vehicle behaviours on a broader scale. One effective method is to undertake Hardware-in-the-Loop (HiL) testing, which involves integrating UAVs components like thrusts, sensors, and onboard computers into the simulation.

The spotlight on autonomous vehicles has grown considerably in recent years. There are possible lines of further research, such as expanding the application of the introduced control strategies to encompass various industries where UAVs are typically employed, like surveillance, photography, inspection, search, and rescue operations. Furthermore, exploring the intricacies of coordination and collaboration mechanisms among the vehicles engaged in complex missions offers substantial potential. The insights from studying these dynamics can be readily transferred and customised for other autonomous vehicle domains.

Finally, for these simulations to be more robust and ensure the feasibility of reference trajectories for performing aggressive manoeuvres, it is crucial to incorporate more robust disturbances models (e.g., gravitational perturbations and atmospheric drag). While current research has made substantial progress in avoiding obstacles, dynamic obstacles present a unique challenge. Consequently, future work in obstacle avoidance should expand its focus to address obstacles that are not static.

Beyond these focused research directions, it is crucial to emphasise the importance of collaboration and interdisciplinary. This work strives to bridge the gap between academic research and practical application. Implementing techniques like MPC in real-world aerospace systems demonstrates the potential for academic findings to drive practical innovations. Collaborative efforts involving industry stakeholders and academia are pivotal in furthering the theory and application of aerospace control. These collaborations foster innovative ideas and real-world solutions, propelling advancements in this domain.

References

[1] J. Hanson. A plan for advanced guidance and control technology for 2nd generation reusable launch vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–9, 2002. doi: 10.2514/6.2002-4557. URL https://arc.aiaa.org/doi/abs/10.2514/6.2002-4557.

[2] C. Ding and L. Lu. A tilting-rotor unmanned aerial vehicle for enhanced aerial locomotion and manipulation capabilities: Design, control, and applications. *IEEE/ASME Transactions on Mechatronics*, 26(4):2237–2248, 2021. doi: 10.1109/TMECH.2020.3036346.

[3] S. Foundation. The space report 2022 q2. *space report*, 2022. URL https://www.spacefoundation.org/2022/07/27/the-space-report-2022-q2/. Accessed: October 26, 2022.

[4] Y.-c. Xie, H. Huang, Y. Hu, and G.-q. Zhang. Applications of advanced control methods in spacecrafts: progress, challenges, and future prospects. *Frontiers of Information Technology & Electronic Engineering*, 17:841–861, 09 2016. doi: 10.1631/FITEE.1601063.

[5] K. M. Ormiston. *A Comparison of PID and Sliding Mode Controllers When Applied to the Orbit Raising of a Satellite Using Solar Sail Propulsion*. PhD thesis, Embry-Riddle Aeronautical University, 2020.

[6] J. Su and K.-Y. Cai. Globally stabilizing proportional-integral-derivative control laws for rigid-body attitude tracking. *Journal of Guidance, Control, and Dynamics*, 34(4):1260–1264, 2011. doi: 10.2514/1.52301.

[7] S. J. Paynter and R. H. Bishop. Adaptive nonlinear attitude control and momentum management of spacecraft. *Journal of Guidance, Control, and Dynamics*, 20(5):1025–1032, 1997. doi: 10.2514/2.4150.

[8] D. Malyuta, Y. Yu, P. Elango, and B. Açıkmeşe. Advances in trajectory optimization for space vehicle control. *Annual Reviews in Control*, 52:282–315, 2021. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2021.04.013. URL https://www.sciencedirect.com/science/article/pii/S1367578821000377.

[9] R. Pérez-Alcocer, J. Moreno-Valenzuela, and R. Miranda Colorado. A robust approach for trajectory tracking control of a quadrotor with experimental validation. *ISA Transactions*, 65, 09 2016. doi: 10.1016/j.isatra.2016.08.001.

[10] M. S. Kamel, T. Stastny, K. Alexis, and R. Siegwart. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. *Studies in Computational Intelligence*, 05 2017. doi: 10.1007/978-3-319-54927-9_1.

[11] B. Guerreiro, R. Cunha, P. Oliveira, and C. Silvestre. Replace project, 2023. URL `https://replace.isr.tecnico.ulisboa.pt/`. Accessed: October 12, 2023.

[12] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017. doi: 10.2514/1.G002507.

[13] M. Islam, M. Okasha, and E. Sulaeman. A model predictive control (mpc) approach on unit quaternion orientation based quadrotor for trajectory tracking. *International Journal of Control, Automation and Systems*, 17, 07 2019. doi: 10.1007/s12555-018-0860-9.

[14] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Model predictive control: Theory, computation, and design. *Automatica*, 36:789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9.

[15] D. Silvestre. Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements. *IEEE Control Systems Letters*, 6:1610–1615, 2022. doi: 10.1109/LCSYS.2021.3129729.

[16] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz. Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69:126–136, 2016. doi: https://doi.org/10.1016/j.automatica.2016.02.036.

[17] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano. Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance. *Aerospace Science and Technology*, 77:585–594, 2018. ISSN 1270-9638. doi: https://doi.org/10.1016/j.ast.2018.04.009. URL `https://www.sciencedirect.com/science/article/pii/S1270963817321223`.

[18] L. Breger and J. How. Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, 31, 09 2008. doi: 10.2514/1.29590.

[19] J. D. B. da Silva Pinto, B. J. N. Guerreiro, and R. M. M. de Almeida Correia da Cunha. Model predictive control strategies for aggressive parcel relay maneuvers using drones. Master's thesis, Universidade de Lisboa - Instituto Superior Técnico, September 2021.

[20] T. P. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Açıkmeşe, and J. M. Carson. Dual quaternion-based powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(9):1584–1599, 2020. doi: 10.2514/1.G004536.

[21] M. A. Mousavi, Z. Heshmati, and B. Moshiri. Ltv-mpc based path planning of an autonomous vehicle via convex optimization. In *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pages 1–7, 2013. doi: 10.1109/IranianCEE.2013.6599610.

[22] A. Botelho, B. Parreira, P. N. Rosa, and J. M. Lemos. *Predictive Control for Spacecraft Rendezvous*. SpringerBriefs in Applied Sciences and Technology. Springer Cham, 2021. doi: 10.1007/978-3-030-75696-3. URL https://doi.org/10.1007/978-3-030-75696-3.

[23] P. O. M. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8):1649–1659, 1999. doi: https://doi.org/10.1002/aic.690450805. URL https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690450805.

[24] L. Yang, A. Katnik, and N. Ozay. Quickly finding recursively feasible solutions for mpc with discrete variables. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 374–381, 2019. doi: 10.1109/CCTA.2019.8920443.

[25] J. Hanema, M. Lazar, and R. Tóth. Stabilizing tube-based model predictive control: Terminal set and cost construction for lpv systems. *Automatica*, 85:137–144, 2017. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2017.07.046. URL https://www.sciencedirect.com/science/article/pii/S0005109817303916.

[26] D. Limon, T. Alamo, and E. Camacho. Enlarging the domain of attraction of mpc controllers. *Automatica*, 41(4):629–635, 2005. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2004.10.011. URL https://www.sciencedirect.com/science/article/pii/S0005109804003085.

[27] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer. An offline-sampling smpc framework with application to autonomous space maneuvers. *IEEE Transactions on Control Systems Technology*, 28(2):388–402, 2020. doi: 10.1109/TCST.2018.2879938.

[28] 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. doi: https://doi.org/10.1016/j.ifacol.2015.08.092. 1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles ACNAAV'15.

[29] R. Sandeepkumar and R. Mohan. Flatness-based reduced hessian method for optimal control of aircraft. *Journal of Guidance, Control, and Dynamics*, 45(5):921–934, 2022. doi: 10.2514/1.G006331.

[30] 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. doi: https://doi.org/10.1002/rnc.2827.

[31] M. Schwenzer, M. Ay, T. Bergs, and D. Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5), 2021. ISSN 1433-3015. doi: 10.1007/s00170-021-07682-3. URL https://doi.org/10.1007/s00170-021-07682-3.

[32] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719, 1952. doi: 10.1073/pnas.38.8.716. URL https://www.pnas.org/doi/abs/10.1073/pnas.38.8.716.

[33] A. Lyapunov. Sur les fonctions de deux variables indépendantes dont les dérivées partielles sont de signe constant. *Annales scientifiques de l'École Normale Supérieure*, 1907. URL https://afst.centre-mersenne.org/item/AFST_1907_2_9__203_0.pdf.

[34] I. Mascolo. Recent developments in the dynamic stability of elastic structures. *Frontiers in Applied Mathematics and Statistics*, 5:51, 10 2019. doi: 10.3389/fams.2019.00051.

[35] M. Althoff. On computing the minkowski difference of zonotopes, 2022.

[36] A. Gibiansky. Quadcopter dynamics and simulation, November 2012. URL https://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/.

[37] T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor uav on se(3). In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, 2010. doi: 10.1109/CDC.2010.5717652.

[38] J. Lofberg. Yalmip : a toolbox for modeling and optimization in matlab. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, pages 284–289, 2004. doi: 10.1109/CACSD.2004.1393890.

[39] MATLAB. *version 7.10.0 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[40] D. Silvestre. Reachtool, March 2023. URL https://github.com/danielmsilvestre/ReachTool.

[41] V. Raghuraman and J. P. Koeln. Set operations and order reductions for constrained zonotopes. *Automatica*, 139:110204, 2022. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2022.110204. URL https://www.sciencedirect.com/science/article/pii/S0005109822000498.

[42] S. V. Raković, E. Kerrigan, K. Kouramas, and D. Mayne. Invariant approximations of the minimal robust positively invariant set. *Automatic Control, IEEE Transactions on*, 50:406 – 410, 03 2005. doi: 10.1109/TAC.2005.843854.

[43] YALMIP Developers. Yalmip - optimizer command, 2021. URL https://yalmip.github.io/command/optimizer/. Accessed: November 10, 2022.

[44] Mosek documentation. https://docs.mosek.com/latest/intro/index.html, 1994. Accessed: November 10, 2022.

[45] Gurobi Optimization. https://www.gurobi.com/, 2008. Accessed: November 10, 2022.

[46] R. N. Palma. Stability-guarantees-for-mpc, 2023. URL https://github.com/ritapalmaa/Stability-Guarantees-for-MPC.

[47] DJI. Dji phantom 4 pro v2, 2023. URL https://www.dji.com/pt/phantom-4-pro-v2.

[48] S. Boyd and L. Vandenberghe. *Convex Optimization*, chapter 4. Cambridge University Press, March 2004. ISBN 0521833787. URL https://web.stanford.edu/~boyd/cvxbook/.

Appendix A

Convex optimisation

This appendix delves into the critical field of convex optimisation, a basis for understanding the mathematical foundations and computational aspects that drive the control strategies presented in this research.

An optimisation problem is a mathematical object of the following form:

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & f(x), \\ \text{subject to} \quad & h_i(x) = 0,\ i = 1, \ldots, p, \\ & g_j(x) \leq 0,\ j = 1, \ldots, m, \end{aligned} \tag{A.1}$$

where $x \in \mathbb{R}^n$ is the optimisation variable, $f : \mathbb{R}^n \to \mathbb{R}$ is the objective or cost function to minimise, $h_1, \ldots, h_p, g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ are constraint functions.

Convex optimisation aims to minimise a convex objective function whilst satisfying a set of convex constraints. The succeeding rules must be satisfied for the problem to be considered convex: f is a convex function; h_i is an affine function for $1 \leq i \leq p$ $\big($that is, $h_i(x) = {a_i}^T x - b_i\big)$; g_j is a convex function for $1 \leq j \leq m$. These impositions ensure the feasible set is always convex, as demonstrated by the veracity definitions below.

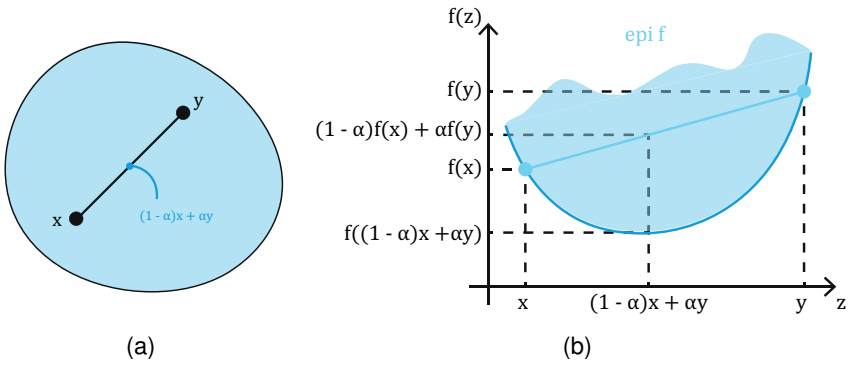
Definition A.0.1 (Convex set, [48])**.** A set $\mathcal{A} \subseteq \mathbb{R}^n$ is convex if, and only if, contains a line segment between any two points in the set $x, y \in \mathcal{A}, 0 \leq \alpha \leq 1$, then

$$(1 - \alpha)x + \alpha y \in \mathcal{A}. \tag{A.2}$$

Definition A.0.2 (Convex function, [48])**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ if

$$f((1 - \alpha)x + \alpha y) \leq (1 - \alpha)f(x) + \alpha f(y),\ \forall x, y \in \mathbb{R}^n \wedge 0 \leq \alpha \leq 1. \tag{A.3}$$

It means the function exists only below the chord connecting any two points on the function graph. The visual representation of this concept can be observed in Figure A.1.



(a) (b)

Figure A.1: A notional convex set and convex function (adapted from [48]). On both, the variable $(1-\alpha)x+\alpha y$ generates a line segment between two points. The epigraph (epi) $f \subseteq \mathbb{R}^n \times \mathbb{R}$ is the set of points that lie above the function and itself defines a convex set. (a) Convex set. (b) Convex function.

One of the main advantages of convex functions is that some specific operations preserve the system's convexity. These succeeding properties aid in the recognition of complex convex functions.

Definition A.0.3 (Non-negative weighted sums of convex functions, [48])**.** Considering $f_i : \mathbb{R}^n \to \mathbb{R}$ is convex for $i = 1, 2, \ldots, m$ and $t_i \geq 0$, then $t_m f_m$ is convex and

$$t_1 f_1 + t_2 f_2 + \ldots + t_m f_m : \mathbb{R}^n \to \mathbb{R} \text{ is convex.} \tag{A.4}$$

Definition A.0.4 (Composition of a convex function with an affine mapping, [48])**.** Assuming $f : \mathbb{R}^n \to \mathbb{R}$ is convex and $g : \mathbb{R}^m \to \mathbb{R}^n,\ g(x) = Ax + b$, then

$$f \circ g : \mathbb{R}^m \to \mathbb{R} \text{ is convex.} \tag{A.5}$$

Definition A.0.5 (Composition with non-decreasing function, [48])**.** $f : \mathbb{R} \to \mathbb{R}$ is convex and non-decreasing and $g : \mathbb{R}^n \to \mathbb{R}$ is convex, then

$$f \circ g : \mathbb{R}^n \to \mathbb{R} \text{ is convex.} \tag{A.6}$$

Definition A.0.6 (Maximum of convex functions, [48])**.** $f_i : \mathbb{R}^n \to \mathbb{R}$ is convex for $i = 1, 2, \ldots, m$, then

$$\max\{f_1, f_2, \ldots, f_m\} : \mathbb{R}^n \to \mathbb{R} \text{ is convex.} \tag{A.7}$$

The convex formulation is appealing due to its ability to capture a broad spectrum of trajectory generation and control applications. The following criteria apply to the available solution algorithms: If a feasible solution exists globally optimal solution is found; when there is no solution, a certificate of infeasibility is issued; the run-time complexity of the problem is polynomial in size; and the algorithms can self-initialise, eliminating the need for an expert initial guess. These characteristics are inherited by convex optimisation-based trajectory generation and control algorithms. Convex programming is thus safer and faster than other optimisation methods for autonomous applications.