Reproducible Hierarchical Model Predictive Control for Autonomous and Safe Mobile Robot Navigation

Dennis Benders

Reproducible Hierarchical Model Predictive Control for Autonomous and Safe Mobile Robot Navigation

Reproducible Hierarchical Model Predictive Control for Autonomous and Safe Mobile Robot Navigation

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus,
Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates,
to be defended publicly on
Tuesday, 21 April 2026, 12:30

by

Dennis Benders

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Prof. dr. J. Alonso-Mora,	Delft University of Technology, *promotor*
Prof. dr. R. Babuška,	Delft University of Technology, *promotor*
Dr. L. Ferranti,	Delft University of Technology, *promotor*
Independent members:	
Prof. dr. G.C.H.E. de Croon,	Delft University of Technology
Prof. dr. ir. M. Wisse,	Delft University of Technology
Prof. dr. T. Faulwasser,	Hamburg University of Technology, Germany
Dr. J. Knotter,	Police Academy of the Netherlands
Prof. Dr.-Ing. J. Kober,	University of Stuttgart, Germany, *reserve member*

This research was supported by the National Police of the Netherlands. All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Keywords:	Real-Time Embedded Autonomous Mobile Robots, Motion Planning and Control, Collision Avoidance, Hierarchical Model Predictive Control, Robust Hierarchical Model Predictive Control, Reproducibility
Printed by:	Gildeprint
Cover by:	Fae van Norden

ISBN 978-94-6518-300-8

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

In theory, theory and practice are the same.
In practice, they are not.

Albert Einstein

CONTENTS

SUMMARY

Autonomous mobile robots have become increasingly capable over the past decades, enabling their use in domains such as logistics, agriculture, and healthcare. This thesis contributes to a collaborative project between the Dutch National Police and the Delft University of Technology, focusing on search and rescue operations in the security domain. In these safety-critical missions, robots are expected to support human first responders by exploring unknown and potentially hazardous environments, including buildings and natural terrains. To reduce the cognitive and operational burden on human operators, robots must navigate cluttered spaces both *autonomously* and *safely*. In the context of this thesis, autonomy refers to onboard decision-making and safety to avoiding collisions despite uncertainties like sensor noise or environmental effects.

This thesis explores autonomous and safe navigation using *model predictive control (MPC)*, a planning and control strategy that predicts future behavior based on a system model and optimizes actions accordingly. MPC's strength lies in its ability to handle constraints such as actuator limits and obstacle avoidance. However, applying MPC in real-world navigation presents several challenges, which this thesis addresses through the following contributions.

Navigating a mobile robot through cluttered environments requires it to follow dynamic trajectories while avoiding obstacles. Although MPC is well-suited for this task, recent methods for collision-free trajectory tracking often rely on complex mathematical formulations that can be difficult to interpret and apply. Chapter 2 aims to bridge this gap by offering a *structured, step-by-step guide*. It explains how to model the robot's navigation problem, formulate the corresponding MPC problem, and establish performance and safety guarantees. The chapter presents three formulations: a nominal MPC approach for ideal conditions, a robust MPC formulation that accounts for bounded disturbances, and a robust output-feedback MPC formulation that additionally handles measurement noise. Each formulation is supported by theoretical insights and practical considerations. While not exhaustive, the guide is intended to support researchers and practitioners in implementing MPC-based navigation under varying levels of uncertainty.

To enable *autonomy*, the MPC formulations introduced in Chapter 2 must operate in real time. This is challenging because MPC relies on solving an optimization problem at each time step, which can be computationally demanding. Planning long-term trajectories and computing control commands at high frequency on embedded hardware is especially difficult. Chapter 3 addresses this by introducing a hierarchical MPC (HMPC) framework that separates planning and control into two layers. The planning MPC handles long-term trajectory generation at a lower frequency, while the tracking MPC focuses on short-term execution at a higher frequency. This separation allows the use of complex nonlinear models in both layers without compromising real-time performance. The tracking MPC layer, freed from long-term planning, can focus on precise

tracking, improving stability and responsiveness. The HMPC framework also includes a method for generating consistent collision avoidance constraints. Its effectiveness is demonstrated through simulations and lab experiments, showing safer and approximately four times faster goal-reaching compared to a single-layer MPC approach.

While the HMPC framework improves autonomy, it leverages a nominal MPC formulation that assumes perfect models and accurate state data. In practice, mobile robots operate in the presence of model uncertainties and noisy measurements, which can lead to constraint violations such as collisions. Chapter 4 extends the HMPC framework to address these issues by incorporating robust output-feedback MPC into the tracking layer. This extension, called robust output-feedback hierarchical MPC (ROHMPC), provides formal *safety* guarantees even in the presence of disturbances and measurement noise. Synthesizing the ROHMPC scheme requires knowledge of uncertainty bounds, which are typically unknown. To overcome this, the chapter introduces an efficient and modular pipeline that estimates these bounds from experimental data, performs necessary offline computations, calibrates constraint tightening to reduce conservatism, and implements the complete control scheme. The pipeline is released as an open-source software package to support reproducibility and future research. Using this pipeline, the chapter demonstrates the successful validation of the ROHMPC framework properties on a simulated quadcopter platform in Gazebo, with reproducible results.

The HMPC and ROHMPC frameworks address autonomy and safety respectively, but successful deployment in real-world scenarios also depends on the reliability of the system. A key aspect of reliability is *reproducibility*: the ability to consistently generate similar results. Chapter 5 explores this concept in the context of robotics, defining reproducibility and analyzing how it applies to the hardware-software setups used in earlier chapters. The HMPC framework satisfies method reproducibility, meaning its implementation can be consistently reproduced. However, due to asynchronous processes and non-deterministic code components, it does not fully achieve results reproducibility. In contrast, the ROHMPC framework satisfies both method and results reproducibility, reinforcing the credibility of the framework. By raising awareness of reproducibility challenges and offering practical insights, this chapter aims to support the development of more robust and trustworthy robotic systems.

All results presented in this thesis have been made publicly accessible through submissions to peer-reviewed venues, an open-access preprint server, and the release of open-source software packages. These results highlight the effectiveness of hierarchical MPC in both simulation and laboratory settings, and demonstrate how formal safety guarantees can be achieved under uncertainty. To support future research, Chapter 6 summarizes the key contributions and outlines several promising directions for further exploration. These include extending the proposed algorithms to 3D environments, integrating onboard sensing for autonomous outdoor navigation, and incorporating data-driven methods to reduce conservatism.

SAMENVATTING

Autonome mobiele robots zijn de afgelopen decennia steeds capabeler geworden, waardoor hun inzet in domeinen zoals logistiek, landbouw en gezondheidszorg mogelijk is. Dit proefschrift draagt bij aan een samenwerkingsproject tussen de Nederlandse Nationale Politie en de Technische Universiteit Delft, gericht op zoek- en reddingsoperaties binnen het veiligheidsdomein. In deze veiligheidskritische missies wordt van robots verwacht dat zij menselijke hulpverleners ondersteunen door onbekende en mogelijk gevaarlijke omgevingen te verkennen, waaronder gebouwen en natuurlijke terreinen. Om de cognitieve en operationele belasting van menselijke operators te verminderen, moeten robots zowel *autonoom* als *veilig* door complexe ruimtes navigeren. In de context van dit proefschrift verwijst autonomie naar besluitvorming aan boord en veiligheid naar het vermijden van botsingen ondanks onzekerheden, zoals sensorruis of omgevingsinvloeden.

Dit proefschrift onderzoekt autonome en veilige navigatie met behulp van *model predictive control (MPC)*, een plannings- en regelstrategie, die toekomstig gedrag voorspelt op basis van een systeemmodel, en bijbehorende acties optimaliseert. De kracht van MPC ligt in het vermogen om beperkingen te verwerken, zoals actuatorlimieten en obstakelvermijding. Het toepassen van MPC in navigatie in de echte wereld brengt echter verschillende uitdagingen met zich mee, die in dit proefschrift worden aangepakt via de volgende bijdragen.

Het navigeren van een mobiele robot door complexe omgevingen vereist het volgen van dynamische trajecten terwijl obstakels worden vermeden. Hoewel MPC hiervoor geschikt is, steunen recente methoden voor botsingsvrije trajectvolging vaak op complexe wiskundige formuleringen, die moeilijk te interpreteren en toe te passen zijn. Hoofdstuk 2 beoogt deze kloof te overbruggen door een *gestructureerde, stapsgewijze handleiding* te bieden. Het legt uit hoe het navigatieprobleem van de robot gemodelleerd kan worden, hoe het bijbehorende MPC-probleem geformuleerd wordt en hoe prestatie- en veiligheidsgaranties kunnen worden vastgesteld. Het hoofdstuk presenteert drie formuleringen: een nominale MPC-aanpak voor ideale omstandigheden, een robuuste MPC-formulering die rekening houdt met begrensde verstoringen, en een robuuste output-feedback MPC-formulering die bovendien meetruis verwerkt. Elke formulering wordt ondersteund door theoretische inzichten en praktische overwegingen. Hoewel niet allesomvattend, is de handleiding bedoeld om onderzoekers en vakmensen te ondersteunen bij het implementeren van MPC-gebaseerde navigatie bij verschillende niveaus van onzekerheid.

Om *autonomie* mogelijk te maken, moeten de in Hoofdstuk 2 geïntroduceerde MPC-formuleringen in real time functioneren. Dit is uitdagend omdat MPC vereist dat bij elke tijdstap een optimalisatieprobleem wordt opgelost, wat veel rekenkracht kost. Het plannen van langetermijntrajecten en het berekenen van regelcommando's met hoge frequentie op embedded hardware is bijzonder moeilijk. Hoofdstuk 3 pakt dit aan door een

hiërarchisch MPC (HMPC) raamwerk te introduceren dat planning en regeling opsplitst in twee lagen. De plannings-MPC behandelt langetermijntrajectgeneratie met een lagere frequentie, terwijl de tracking-MPC zich richt op kortetermijnuitvoering met een hogere frequentie. Deze scheiding maakt het mogelijk om complexe niet-lineaire modellen in beide lagen te gebruiken zonder concessies te doen aan real-time prestaties. De tracking-MPC-laag, bevrijd van langetermijnplanning, kan zich concentreren op nauwkeurige tracking, wat stabiliteit en responsiviteit verbetert. Het HMPC-raamwerk bevat ook een methode voor het genereren van consistente obstakelvermijdingsbeperkingen. De effectiviteit ervan wordt aangetoond via simulaties en labexperimenten, waarbij veiliger en ongeveer vier keer sneller doelen worden bereikt in vergelijking met een enkellaags MPC-aanpak.

Hoewel het HMPC-raamwerk de autonomie verbetert, maakt het gebruik van een nominale MPC-formulering die perfecte modellen en nauwkeurige toestandsdata veronderstelt. In de praktijk werken mobiele robots in omstandigheden met modelonzekerheden en meetruis, wat kan leiden tot schendingen van beperkingen zoals botsingen. Hoofdstuk 4 breidt het HMPC-raamwerk uit om deze problemen aan te pakken door robuuste output-feedback MPC in de trackinglaag op te nemen. Deze uitbreiding, robuuste output-feedback hiërarchische MPC (ROHMPC) genoemd, biedt formele *veiligheids*garanties, zelfs in aanwezigheid van verstoringen en meetruis. Het synthetiseren van het ROHMPC-schema vereist kennis van onzekerheidsgrenzen, die doorgaans onbekend zijn. Om dit te overwinnen, introduceert het hoofdstuk een efficiënte en modulaire pijplijn die deze grenzen schat op basis van experimentele data, noodzakelijke offline berekeningen uitvoert, constraint-tightening kalibreert om conservatisme te verminderen en het volledige regelschema implementeert. De pijplijn wordt vrijgegeven als een open-source softwarepakket om reproduceerbaarheid en toekomstig onderzoek te ondersteunen. Met behulp van deze pijplijn toont het hoofdstuk de succesvolle validatie van de ROHMPC-raamwerkeigenschappen op een gesimuleerd quadcopterplatform in Gazebo, met reproduceerbare resultaten.

De HMPC- en ROHMPC-raamwerken behandelen respectievelijk autonomie en veiligheid, maar succesvolle inzet in realistische scenario's hangt ook af van de betrouwbaarheid van het systeem. Een belangrijk aspect van betrouwbaarheid is *reproduceerbaarheid*: het vermogen om consequent vergelijkbare resultaten te genereren. Hoofdstuk 5 verkent dit concept in de context van robotica, definieert reproduceerbaarheid en analyseert hoe dit van toepassing is op de hardware-softwareconfiguraties die in eerdere hoofdstukken zijn gebruikt. Het HMPC-raamwerk voldoet aan methodereproduceerbaarheid, wat betekent dat de implementatie consistent kan worden gereproduceerd. Vanwege asynchrone processen en niet-deterministische codecomponenten wordt resultatenreproduceerbaarheid echter niet volledig bereikt. Daarentegen voldoet het ROHMPC-raamwerk aan zowel methode- als resultatenreproduceerbaarheid, wat de geloofwaardigheid van het raamwerk versterkt. Door bewustzijn te creëren rond reproduceerbaarheidsuitdagingen en praktische inzichten te bieden, beoogt dit hoofdstuk de ontwikkeling van robuustere en betrouwbaardere robotsystemen te ondersteunen.

Alle resultaten die in dit proefschrift worden gepresenteerd zijn openbaar toegankelijk gemaakt via inzendingen naar gepeerreviewde publicatiekanalen, een open-access preprintserver en de vrijgave van open-source softwarepakketten. Deze resultaten bena-

drukken de effectiviteit van hiërarchische MPC in zowel simulatie- als labomgevingen en tonen aan hoe formele veiligheidsgaranties kunnen worden bereikt onder onzekerheid. Ter ondersteuning van toekomstig onderzoek vat Hoofdstuk 6 de belangrijkste bijdragen samen en schetst verschillende veelbelovende richtingen voor verdere verkenning. Deze omvatten het uitbreiden van de voorgestelde algoritmen naar 3D-omgevingen, het integreren van onboard sensing voor autonome buitennavigatie, en het opnemen van data-gedreven methoden om conservatisme te verminderen.

ACRONYMS

AI artificial intelligence. 140, 180

ASV autonomous surface vessel. 15, 16, 136, 178

CBF control barrier function. 13, 78, 108, 140

CCM control contraction metric. 52, 55, 78

CRAMPC contingency robust adaptive MPC. 16, 136

GO-MPC goal-oriented MPC. 87, 95, 96

GP Gaussian Process. 15, 36, 108, 140, 179

GP-MPC Gaussian Process MPC. 15

GPS global positioning system. 123, 139

HJ Hamilton-Jacobi. 78, 140

HMPC hierarchical MPC. 13–16, 75–77, 79–82, 89–93, 95, 98–102, 104–107, 109, 111, 121, 136, 137, 139, 142

IO input-output. 109, 110, 117, 137

LiDAR Light Detection And Ranging. 106, 123, 139

LMI linear matrix inequality. 31–33, 54–58, 69–73, 83, 86, 92, 113, 115, 146–156

MHE moving horizon estimation. 108–110, 137

MPC model predictive control. 12–16, 19–27, 29, 31, 34, 36, 37, 39, 63, 76–81, 93, 94, 96, 98–100, 108, 117, 118, 121, 128, 136–141, 143, 178

MVT mean value theorem. 55, 69, 86, 152, 153

NLP nonlinear programming. 80, 97, 117

NMPC nonlinear MPC. 12, 14, 16, 19, 20, 75, 136, 138, 141

PMPC planning MPC. 13, 14, 76, 77, 79–82, 84–96, 99–105, 111, 115, 117, 119, 128, 129, 136, 138, 159

RK4 fourth-order Runge–Kutta method. 94, 100

RMPC robust MPC. 13–16, 34–37, 39, 47, 59, 61, 63, 70, 108–111, 120, 136, 139

ROHMPC robust output-feedback HMPC. 15, 107, 109–111, 115, 119–121, 137, 139, 142

ROMPC robust output-feedback MPC. 14, 16, 20, 59–61, 63, 73, 74, 107, 109, 111, 113, 117, 136, 137

ROS Robot Operating System. 80, 124, 128–130, 132

RPI robust positive invariant. 56, 57, 69, 70, 72, 73, 146, 151–153

SDP semidefinite program. 31–33, 54, 56–58, 70–73, 82, 83, 85, 92, 94, 95, 112, 113, 115, 117, 118

SMPC single-layer MPC. 13, 75, 76, 80, 81, 93–96, 99–105, 137

TMPC tracking MPC. 13, 14, 22–25, 31, 34–36, 39, 40, 43, 44, 47, 56, 60, 76, 77, 79–82, 84–86, 88–96, 100, 101, 104–107, 111, 114, 115, 117–119, 128, 129, 136, 158

UQ uncertainty quantification. 113, 117

1

INTRODUCTION

1

Imagine the following scenario: a drug dealer has just been apprehended, leaving behind an abandoned drug laboratory. The police must now enter the lab to retrieve relevant materials such as drugs and weapons. This task involves not only object retrieval but also ensuring officer safety and avoiding detection by potential accomplices.

This scenario is not purely hypothetical; it reflects a real emergency reconnaissance operation broadcasted on Dutch television in 2025 [1]. The program illustrates the current approach: officers drive to the lab, identify the easiest entry point, and enter without knowing what awaits them. They search each room, collect relevant items, and exit as quickly as possible. Figure 1.1 shows a snapshot from the broadcast to help visualize the situation.

Figure 1.1: Snapshot taken from [1]: police officers searching for objects of interest in a drug lab, risking their own safety.

Now, consider an alternative scenario: as one team drives toward the lab, another deploys an aerial vehicle to inspect the site and its surroundings. Equipped with cameras and sensors, the drone identifies potential threats and suggests optimal entry points. After breaching the door, a second aerial vehicle enters the lab to detect booby traps and hazardous gases, and maps the interior, including the locations of relevant objects. With a clear understanding of the environment, the officers enter while the drone continues exploring the remaining areas.

You may find that the second scenario appears safer and more efficient than the first. More importantly, it illustrates a concrete real-world example of how mobile robots can collaborate with humans to enhance safety and efficiency in future emergency reconnaissance operations. This is particularly relevant to the context of this thesis: a collaborative research project between the Delft University of Technology and Dutch National Police. The project aims to develop mobile robots that assist police forces during emergency reconnaissance missions, such as the one described above.

Emergency reconnaissance is just one of many domains where mobile robots can make valuable contributions. In fact, mobile robots are increasingly playing important roles in society [2]. Other example domains include urban transportation [3], warehouse automation [4], manufacturing [5], education [6], space exploration [7], and disaster response [8]. As a result, there is growing momentum for mobile robot development across a diverse range of societal sectors.

A key aspect in robot development is their ability to contribute to missions by performing tasks without human intervention, that is, to act autonomously. Although autonomy is a complex and debated concept [9], this thesis defines it as the robot's ability to perform tasks independently of human input. Autonomous behavior can relieve humans from dull, repetitive, or time-consuming tasks, thereby contributing positively to their well-being [10]. Enhancing human well-being, along with building trust, is essential for societal acceptance of robots, as demonstrated in a recent study conducted in Bremen [11]. Consequently, designing algorithms that enable robots to act both *autonomously* and *safely* is a crucial research direction.

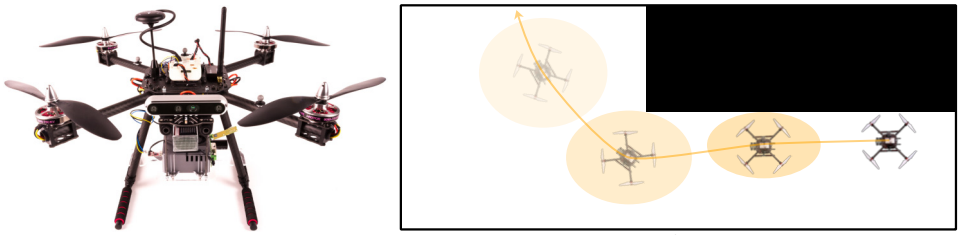
To illustrate autonomous behavior, consider the aerial vehicle tasked with exploring and mapping the drug lab. Its responsibilities include:

- *High-level decision-making*: The robot must determine which room to explore next, based on its current knowledge of the environment and the objects already identified.
- *Perception*: To make informed decisions, the robot must perceive its surroundings. This includes estimating its own state (e.g., position and orientation), detecting obstacles, and identifying objects of interest.
- *Motion planning*: The robot must plan a trajectory to the next location of interest, avoiding obstacles such as walls and furniture, while accounting for its dynamics and actuator limitations.
- *Control*: The robot must accurately follow the planned trajectory by controlling its actuators.

These tasks are not specific to emergency reconnaissance operations; they are common across many mobile robot applications. This thesis focuses on the motion planning and control aspects. For further insights into high-level decision-making and perception, the reader is referred to works such as [12]–[14]. A study that specifically addresses high-level decision-making in emergency reconnaissance is [15], which proposes a frontier-based exploration algorithm that generates greedy waypoints to guide efficient exploration.

Designing motion planning and control algorithms is essential for safe navigation. However, real-world uncertainties such as sensor noise and aerodynamic effects present significant challenges [8]. These uncertainties can cause deviations from planned trajectories and may lead to collisions. Therefore, ensuring *safety* under uncertainty, also referred to as robustness, is a critical research focus.

Research in this direction remains relatively limited, as it is a challenging topic and often involves trade-offs with performance [8]. An important question to consider is:



(a) (b)

Figure 1.2: (a) Quadrotor platform used in Chapter 3, and (b) the problem of safe motion planning and control. (b) illustrates the worst-case uncertainty regions of the predicted robot movement over time and how a safe motion planning and control scheme accounts for these uncertainties by constructing a plan with increased clearance around obstacles.

How safe is safe enough? [16]. The authors of [16] mention that "the importance of safety cannot be over-emphasized" and argue that researchers have an active role to play in addressing this issue.

Motivated by this perspective, and by the belief that all available knowledge should be used to improve robot safety, this thesis aims to contribute to robust planning and tracking guarantees in real-world settings. Specifically, the focus lies on the design of model predictive control (MPC) schemes, in particular *nonlinear MPC (NMPC)* schemes, that enable autonomous and safe navigation of mobile robots in the presence of model and measurement uncertainty.

The next section provides a more detailed overview of the challenges involved in designing NMPC schemes for mobile robots, along with the knowledge gaps that this thesis seeks to address.

1.1. RELATED WORK

Before addressing the challenges of NMPC for mobile robots, it is important to recognize the wide variety of existing motion planning and control algorithms. Motion planning methods can be broadly categorized into graph-search [17], sampling-based [18], and optimization-based [19] approaches. Examples of control, or tracking, algorithms include proportional-integral-derivative [20], linear-quadratic regulator [21], and MPC [22]. For a comprehensive overview of these algorithms, refer to the review works in [23]–[27].

In addition to these classical approaches, learning-based methods have been proposed for both motion planning and control, as discussed in [24], [28]. These methods are particularly effective in scenarios where modeling is difficult. However, proving their safety remains a significant challenge. For this reason, learning-based algorithms are considered out of scope in this thesis. They may offer additional capabilities, but these can be integrated on top of the MPC-based methods presented here.

MPC is an optimization-based method that computes control actions by minimizing a cost function, given the robot's dynamical model [22]. The objective typically includes minimizing the distance to a goal or a time-varying reference trajectory. The model is used to predict future robot states and is incorporated in the form of a constraint in the

optimization problem, along with other constraints that enforce state and actuator limits and ensure obstacle avoidance. At each iteration, MPC computes the optimal control actions and the corresponding future trajectory, then applies the first control action. By repeating this process continuously, MPC can adapt to changes in the robot's state or environment. This adaptability makes MPC suitable for both planning and tracking.

Whereas MPC can be used for planning and tracking in multi-robot settings in the form of decentralized MPC [29] or distributed MPC [30], this thesis focuses on single-robot applications. In this context, MPC can be used in two configurations: single-layer MPC (SMPC), as in [31], or *hierarchical MPC (HMPC)*, as in [32]. In SMPC, a single MPC layer is responsible for both planning the trajectory and computing the control actions to track it. The advantage of this approach is its simplicity, as only one algorithm is required. However, it involves a trade-off between planning horizon and control frequency to ensure real-time feasibility. To address this limitation, HMPC introduces two layers: a planning MPC (PMPC) layer that plans long-horizon trajectories at a slower rate, and a tracking MPC (TMPC) layer that tracks the planned trajectory at a higher rate. This separation has the potential to improve both planning and tracking performance.

Despite the potential of HMPC, existing schemes face specific limitations. One major challenge is the computational cost, particularly in the PMPC layer. Solving the optimization problem can be *time-consuming* due to the use of nonlinear robot models or the non-convex nature of typical planning problems. Existing work has addressed this issue by either using a low-fidelity planning model [33] or by optimizing over a discrete set of motion primitives [34]. The first approach reduces computational complexity but may result in suboptimal plans. The second allows for more expressive plans but restricts the robot's maneuverability due to the discrete nature of the motion primitives. This leads to an open research question: *How can we design a real-time feasible HMPC scheme that enables expressive planning?*

In addition to computational challenges, MPC schemes can suffer from *infeasibility* in certain scenarios. Infeasibility means that no solution exists that satisfies all constraints. In practice, this can also occur when the problem is poorly scaled, causing numerical issues for the optimizer. In either case, the solver may return no usable solution, and executing the first control action could be unsafe. Therefore, it is essential to ensure that both PMPC and TMPC layers remain feasible at all times; a property known as recursive feasibility.

A third challenge involves *model uncertainty*, which includes measurement noise and unmodeled effects such as aerodynamics and friction. These effects are often difficult to capture accurately and are typically excluded from the MPC formulation. As a result, the predicted trajectory may deviate from the actual robot behavior, increasing the risk of collisions. Measurement noise further contributes to this issue by introducing errors in the state estimate. Common approaches to address these challenges include control barrier functions (CBFs) [35], reachability-based methods [36], and robust MPC (RMPC) techniques [37]. While these methods differ in how they handle uncertainty, they all aim to bound its effect on the system behavior. This thesis focuses on RMPC methods, with particular emphasis on obtaining realistic uncertainty bounds for mobile robots.

Solving the three challenges outlined above is essential for the successful deploy-

ment of MPC-based motion planning and tracking schemes on mobile robot platforms in real-world environments. Recent advances in MPC control theory have shown promising results in this direction, for example the contributions presented in [38], [39]. However, applying these theoretical developments to robotics remains a nontrivial task. The aim of this thesis is to help bridge this gap and provide insights that support future research in this area.

In addition to the research contributions related to MPC, this thesis also addresses the important topic of *reproducibility* in robotics research. Reproducibility is a fundamental aspect of the scientific method, as it enables researchers to verify results and build upon prior work. Achieving reproducibility in robotics, however, is particularly challenging due to the complexity of robotic systems, which typically involve the integration of diverse hardware and software components. This thesis aims to contribute to a better understanding of reproducibility in robotics research. Furthermore, it analyzes the proposed frameworks and how they take a step towards more reproducible results.

1.2. CONTRIBUTIONS

The contributions of this thesis are:

(A) **A Step-by-Step Guide on NMPC for Safe Mobile Robot Navigation**
This thesis first presents a step-by-step guide on NMPC for safe mobile robot navigation, introduced in Chapter 2. The guide serves as a theoretical foundation for Chapters 3 and 4, and is intended for early-stage researchers and robotics practitioners. It explains the principles underlying NMPC, translates them into mathematical formulations, and demonstrates how these can be used to prove recursive feasibility and obstacle avoidance for nominal MPC, RMPC, and robust output-feedback MPC (ROMPC) schemes.

(B) **A Novel Real-Time Embedded HMPC Scheme with Recursive Feasibility and Collision Avoidance Guarantees**
The second contribution, presented in Chapter 3, addresses the design of an HMPC scheme that ensures recursive feasibility and collision avoidance in both the PMPC and TMPC layers, under the assumption of no model mismatch. The scheme builds on the nominal MPC theory from the guide and uses an expressive nonlinear quadrotor model in both layers, overcoming limitations of other approaches discussed in Section 1.1. Experimental results show that the scheme is practically recursively feasible and capable of running in real time on an NVIDIA Jetson Xavier NX embedded computer. The implementation and documentation are available in the corresponding GitHub repository at `https://github.com/dbenders1/hmpc`.

(C) **An Efficient and Modular RMPC Design Pipeline with Safety Guarantees in the Presence of Model Uncertainty**
The third contribution of this thesis, presented in Chapter 4, introduces an efficient and modular design pipeline for constructing RMPC schemes with safety guarantees in the presence of model uncertainty. The pipeline consists of several stages, including uncertainty quantification, robust offline MPC design, constraint

tightening, and the synthesis of a robust output-feedback HMPC (ROHMPC) scheme. The ROHMPC scheme extends the HMPC approach from Chapter 3 and provides recursive feasibility and obstacle avoidance guarantees. The complete design pipeline is implemented and made available as open-source code in the corresponding GitHub repository at `https://github.com/dbenders1/rohmpc`.

(D) **An Interpretation of Reproducibility in Robotics Research and the Level of Reproducibility of the Frameworks Used in this Thesis**
The final contribution of this thesis focuses on reproducibility in robotics research. Chapter 5 begins by defining reproducibility and interpreting its meaning in the context of robotics, where it is often hindered by diverse hardware-software setups and limited access to (well-documented) code. Given this context, the chapter analyzes the framework used for the HMPC contribution in Chapter 3, including an assessment of its level of reproducibility. The results are based on a self-developed quadrotor platform and corresponding software setup. Lessons learned from this project informed the framework used in Chapter 4, which adopted the open-source Agilicious stack [40] and emphasized deterministic software behavior. The chapter also evaluates the level of reproducibility of this second framework.

1.3. RESEARCH BEYOND THIS THESIS

Besides the content described in this work, the work executed as part of the project has also contributed to other research directions beyond the scope of this thesis. These research directions include:

(a) **A Novel Gaussian Process MPC (GP-MPC) Scheme with Probabilistic Safety Guarantees for Outdoor Quadrotor Flight Subject to Wind Disturbances**
To address the conservativeness of RMPC schemes in the presence of highly variable wind disturbances, this work explored the use of a Gaussian Process (GP) [41] to estimate the effective wind force acting on a quadrotor. The GP-MPC scheme incorporates both the estimated wind force and its uncertainty into chance constraints within the MPC formulation. This allows the robot to fly closer to obstacles under low-wind conditions and maintain greater distances under high-wind conditions, providing probabilistic safety guarantees. The framework was developed in two stages: initially with offline GP-based wind field learning followed by online GP-MPC control [42], and later with fully online GP-based learning and control [43], enabling adaptation to changing wind conditions during flight.

(b) **Rule-aware Trajectory Optimization for autonomous surface vessels (ASVs)**
Building on the codebase and insights developed during this thesis, a collaboration was established on rule-aware trajectory optimization for ASVs [44]. For ASVs, compliance with maritime traffic regulations, known as the International Regulations for Preventing Collisions at Sea, is essential. This work focused on embedding these rules as constraints in the trajectory optimization problem, allowing ASVs to safely overtake and cross other vessels while adhering to the international regulations.

(c) **A Novel contingency robust adaptive MPC (CRAMPC) Scheme for MPC-based Contingency Planning of ASVs Subject to Actuator Faults**
Extending both the rule-aware trajectory optimization and the RMPC work in Chapter 4, this contribution introduces a CRAMPC scheme for ASVs operating under disturbances and actuator faults [45, Chap. 6]. The scheme uses a robust adaptive MPC formulation based on [46] to compute trajectories that are safe under worst-case disturbances and parameter uncertainty. Additionally, it includes a contingency planning component that generates conservative fallback trajectories accounting for potential actuator faults. Together, these elements ensure collision avoidance for a system subject to uncertain parameters, disturbances, and actuator faults.

(d) **Free energy principle for State and Input Estimation of a Quadcopter Flying in Wind**
Early work in this project contributed to research on brain-inspired state and input estimation using dynamic expectation maximization [47], based on the free energy principle. Dynamic expectation maximization leverages the structured nature of model uncertainty, which implies the existence of higher-order derivatives that provide additional information about the evolution of states and inputs. By exploiting this structure, the method enables more accurate estimation of both states and inputs for a quadcopter operating in windy conditions.

1.4. OUTLINE

Figure 1.3 provides a visual overview of the structure and content of this thesis. As outlined earlier, the main objective is to bridge the gap between MPC control theory and safe mobile robot navigation. To support this goal, Chapter 2 presents a step-by-step guide on NMPC schemes for mobile robot navigation. This theoretical foundation is essential for understanding the concepts and methods introduced in the subsequent chapters. Chapter 3 builds on this foundation and describes the design of a real-time feasible HMPC scheme. The scheme ensures recursive feasibility and is implemented on the embedded computer of a quadrotor platform. Chapter 4 introduces a robust design pipeline that extends the HMPC approach to account for model uncertainty. This pipeline includes uncertainty quantification, RMPC design, constraint tightening, and the synthesis of an ROMPC scheme. The experimental work in Chapters 3 and 4 relies on the robotic hardware and software frameworks described in Chapter 5. This chapter also discusses the broader topic of reproducibility in robotics research and evaluates the reproducibility of the setups used in this thesis. Finally, Chapter 6 summarizes the main findings and contributions of this work and outlines promising directions for future research aimed at advancing the deployment of safe mobile robots.

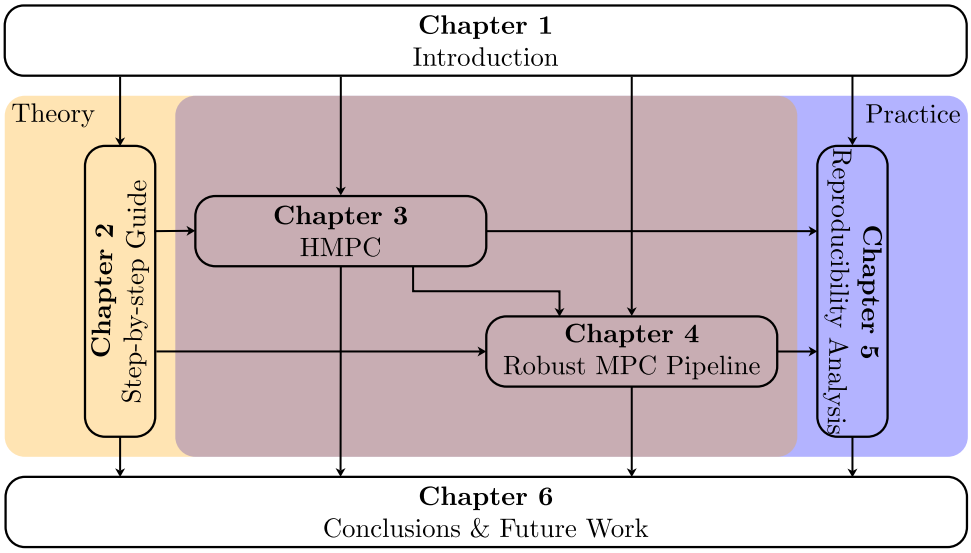


Figure 1.3: Visual overview of the content presented in this thesis.

2

A Step-by-Step Guide on Nonlinear Model Predictive Control for Safe Mobile Robot Navigation

As discussed in Chapter 1, the overarching goal of this thesis is to contribute to bridging the gap between recently developed MPC-based guarantees for dynamic trajectory tracking and their practical applicability in robotics.

To deepen our understanding of this challenge, this chapter presents a comprehensive guide to designing NMPC schemes for safe mobile robot navigation. It is structured to provide a clear exposition of the theoretical principles and practical design considerations required to develop NMPC controllers that ensure safe trajectory tracking through constraint satisfaction. This chapter lays the groundwork for the subsequent contributions, which build upon these concepts to introduce novel hierarchical and robust MPC formulations for mobile robots. The material is intended to be accessible to researchers, engineers, and practitioners aiming to translate theoretical insights into real-world robotic applications.

This chapter is based on and taken in parts literally from:

D. Benders, L. Ferranti, and J. Köhler, "A step-by-step guide on nonlinear model predictive control for safe mobile robot navigation", *arXiv preprint arXiv:2507.17856*, Aug. 2025.
Statement of contributions: DB created the overview and wrote the technical report, LF and JK provided feedback on the report.

2.1. INTRODUCTION

Designing an MPC scheme that enables a mobile robot to safely navigate environments populated with obstacles is a complex yet essential challenge in robotics. Here, safety refers to the robot's ability to satisfy state and input constraints while avoiding collisions, even in the presence of external disturbances and measurement noise.

This chapter presents a step-by-step guide to the design and implementation of NMPC schemes that address these safety requirements. While numerous books and survey papers provide comprehensive overviews of linear MPC [48], [49], NMPC [22], [50]–[53], and their applications in robotics [54]–[57], this chapter does not aim to replicate those exhaustive reviews. Instead, it aims to offer a practical and accessible path from theoretical principles to implementation, with a strong emphasis on safety and performance guarantees.

2.1.1. OVERVIEW

To implement an NMPC (hereafter also simply referred to as MPC) scheme on a mobile robot, several key components must be addressed:

- *System modeling*: define the robot's dynamics, its environment, and task objectives (Section 2.2).
- *Reference tracking*: formulate the MPC problem for trajectory tracking (Section 2.3). This formulation is related to the ones described in [38], [58] and Chapter 3.
- *Robustness to disturbances*: modify the MPC formulation to ensure safety under disturbances (Section 2.4). This formulation is related to the ones described in [37], [59]–[62].
- *Robustness to model uncertainty*: develop a ROMPC scheme to handle model uncertainty consisting of both disturbances and measurement noise (Section 2.5). This formulation is related to the ones described in [46], [63]–[66] and Chapter 4.

Each MPC scheme is introduced in a structured manner. The formulation is presented first, followed by a discussion of its desirable properties. Detailed mathematical proofs are then provided to demonstrate how these properties are satisfied. These proofs rely on specific assumptions, which are shown to hold under the proposed designs.

2.1.2. NOTATION

Throughout this chapter and the remainder of this thesis, vectors are denoted in bold and matrices in capital letters. The sets $\mathbb{R}_{[a,b)}$ and $\mathbb{N}_{[a,b)}$ represent real and natural numbers, respectively, ranging from a (inclusive) to b (exclusive). The interior of a set A is indicated with $\text{int}(A)$ and the Minkowski sum operator is given by $\oplus$. The Euclidean norm for vectors and the induced norm for matrices are denoted by $\|\cdot\|$, while the quadratic norm with respect to a positive (semi-)definite matrix Q is written as $\|\boldsymbol{x}\|_Q = \boldsymbol{x}^\top Q \boldsymbol{x}$, where $Q \succ 0$ or $Q \succeq 0$. Subscripts indicate time steps, indices, or both, and superscripts denote the quantity to which a variable belongs. For example, $g^{\text{r,o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t})$ refers to the

jth obstacle avoidance constraint for the reference trajectory, expressed in terms of the position states $\boldsymbol{p}_{\tau|t}$ at prediction stage $\tau \in [0, T]$ of the MPC problem solved at time t with prediction horizon T. The predicted state-input trajectory at time t is denoted by $(\boldsymbol{x}_{\tau|t}, \boldsymbol{u}_{\tau|t})$, with the optimal solution written as $(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t})$ and a candidate solution as $(\tilde{\boldsymbol{x}}_{\tau|t}, \tilde{\boldsymbol{u}}_{\tau|t})$. Finally, $\boldsymbol{x}_{\cdot|t}$ represents all states over the prediction horizon.

2.2. MOBILE ROBOT, ENVIRONMENT, AND TASK DESCRIPTION

The nominal robot system dynamics are given by

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{2.1}$$

with state $\boldsymbol{x} \in \mathbb{R}^{n^{\mathrm{x}}}$, control input $\boldsymbol{u} \in \mathbb{R}^{n^{\mathrm{u}}}$, and Lipschitz continuous nonlinear function $f(\boldsymbol{x}, \boldsymbol{u})$. The system is subject to polytopic input and state constraints, also called system constraints, described as

$$\mathcal{Z} := \mathcal{U} \times \mathcal{X} = \left\{ (\boldsymbol{u}, \boldsymbol{x}) \in \mathbb{R}^{n^{\mathrm{u}}+n^{\mathrm{x}}} | g^{\mathrm{s}}_j(\boldsymbol{x}, \boldsymbol{u}) \leq 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]} \right\}, \tag{2.2}$$

with $g^{\mathrm{s}}_j(\boldsymbol{x}, \boldsymbol{u}) = L^{\mathrm{s}}_j \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{x} \end{bmatrix} - l^{\mathrm{s}}_j, L^{\mathrm{s}}_j \in \mathbb{R}^{1 \times n^{\mathrm{s}}}, l^{\mathrm{s}}_j \in \mathbb{R}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$. Note that each input and state constraint has a lower and upper bound. Therefore, there exist $n^{\mathrm{s}} = 2(n^{\mathrm{u}} + n^{\mathrm{x}})$ system constraints in total.

The mobile robot operates in an environment with obstacles. To prevent obstacle collisions, we need to ensure that robot region $\mathscr{R}$ does not intersect with obstacle regions $\mathscr{O}$. Correspondingly, we need to formulate obstacle avoidance constraints that the MPC satisfies. In this thesis, the obstacle avoidance constraints are assumed to be constructed as polytopic inner approximations of the possibly non-convex collision-free space. Since the trajectory of the robot progresses over time, these constraints should be defined as a function of time t and prediction stage τ in the MPC horizon:

$$\mathscr{F}_{\tau|t} := \left\{ \boldsymbol{p}_{\tau|t} \in \mathbb{R}^{n^{\mathrm{p}}} \;\middle|\; g^{\mathrm{o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}) \leq 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}\ \tau \in [0, T],\ t \geq 0 \right\}, \tag{2.3}$$

with position $\boldsymbol{p} = M\boldsymbol{x} \in \mathbb{R}^{n^{\mathrm{p}}}$, where M is a matrix selecting the position states from $\boldsymbol{x}$, and $g^{\mathrm{o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}) = L^{\mathrm{o}}_{j,\tau|t}\boldsymbol{p}_{\tau|t} - l^{\mathrm{o}}_{j,\tau|t}, L^{\mathrm{o}}_{j,\tau|t} \in \mathbb{R}^{1 \times n^{\mathrm{p}}}, l^{\mathrm{o}}_{j,\tau|t} \in \mathbb{R}, j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$. Without loss of generality, we set $\|L^{\mathrm{o}}_{j,\tau|t}\| = 1, j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$ by scaling the constraints. For notational convenience, let us denote the obstacle avoidance constraints by $g^{\mathrm{o}}(\boldsymbol{p}) = L^{\mathrm{o}}\boldsymbol{p} - l^{\mathrm{o}}$ if we write about general constraints properties and $g^{\mathrm{o}}_j(\boldsymbol{p}) = L^{\mathrm{o}}_j\boldsymbol{p} - l^{\mathrm{o}}_j,\ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$ if we write constraints properties that need to hold for all constraints explicitly.

Remark 1. *Obstacle avoidance constraints (2.3) form convex polytopic sets that change over the prediction time τ at time t. A method to compute a single convex polytope is described in [67]. Based on this, the work in Chapter 3 proposes a method to construct a sequence of discrete polytopic obstacle avoidance constraints, which are used throughout this thesis. A similar way to represent the obstacle avoidance constraints is recently proposed in [68], in which the constraints are continuously parameterized and differentiable off-centered ellipsoids based on polynomial path segments. This allows them to be more flexible at the cost of increased computation time compared to a discrete sequence of constraints.*

The task of the mobile robot is to track a reference trajectory that satisfies the following property:

Property 1. *Reference trajectory*

$$\boldsymbol{r} = [\boldsymbol{u}^{\mathrm{r}\top}\, \boldsymbol{x}^{\mathrm{r}\top}]^{\top}: \tag{2.4}$$

1. *is dynamically feasible:*

$$\dot{\boldsymbol{x}}^{\mathrm{r}}{}_{t+\tau} = f(\boldsymbol{x}^{\mathrm{r}}_{t+\tau}, \boldsymbol{u}^{\mathrm{r}}_{t+\tau}); \tag{2.5}$$

2. *satisfies a tightened set of the system constraints:*

$$\boldsymbol{r}_{t+\tau} \in \tilde{\mathcal{Z}}, \tag{2.6}$$

where $\tilde{\mathcal{Z}}$ *is defined as*

$$\tilde{\mathcal{Z}} := \tilde{\mathcal{U}} \times \tilde{\mathcal{X}} = \left\{(\boldsymbol{u}, \boldsymbol{x}) \in \mathbb{R}^{n^{\mathrm{u}}+n^{\mathrm{x}}} \;\middle|\; g^{\mathrm{r,s}}_j(\boldsymbol{x}, \boldsymbol{u}) \le 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}\right\} \subseteq \mathrm{int}(\mathcal{Z}), \tag{2.7}$$

with later introduced functions $g^{\mathrm{r,s}}_j(\boldsymbol{x}, \boldsymbol{u}),\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$;

3. *satisfies a tightened set of the obstacle avoidance constraints:*

$$M\boldsymbol{x}^{\mathrm{r}}_{t+\tau} \in \tilde{\mathcal{F}}_{\tau|t}, \tag{2.8}$$

where $\tilde{\mathcal{F}}_{\tau|t}$ *is defined as*

$$\tilde{\mathcal{F}}_{\tau|t} := \left\{\boldsymbol{p}_{\tau|t} \in \mathbb{R}^{n^{\mathrm{p}}} \;\middle|\; g^{\mathrm{r,o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}) \le 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}\right\} \subseteq \mathrm{int}(\mathcal{F}_{\tau|t}),\ \tau \in [0, T],\ t \ge 0, \tag{2.9}$$

with later introduced functions $g^{\mathrm{r,o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}),\ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$.

The tightening of the system and obstacle avoidance constraints is included to ensure that the original constraints are satisfied as long as the real robot tracks the reference trajectory with a bounded error.

In general, generating reference trajectories satisfying Property 1 is nontrivial. Therefore, a method to generate such reference trajectories is proposed in Chapter 3.

2.3. MPC FOR TRAJECTORY TRACKING

This section aims to introduce the MPC scheme that will be used to track the reference trajectory defined in Property 1, the so-called TMPC. To explain TMPC's properties, Section 2.3.1 directly states its formulation. Then, we will discuss the elements in the formulation and the properties it should satisfy to ensure that the mobile robot safely navigates through the environment. This is the topic of Section 2.3.2. After discussing the TMPC design, we will provide some concluding remarks on the TMPC formulation in Section 2.3.3.

2.3.1. TMPC FORMULATION

Consider the following MPC formulation for tracking a reference trajectory defined in Property 1:

$$\mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) = \min_{\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}} \ \mathscr{J}^{\mathrm{f}}(\boldsymbol{x}_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}) + \int_0^T \mathscr{J}^{\mathrm{s}}(\boldsymbol{x}_{\tau|t}, \boldsymbol{u}_{\tau|t}, \boldsymbol{r}_{t+\tau})\, d\tau, \tag{2.10a}$$

$$\text{s.t. } \boldsymbol{x}_{0|t} = \boldsymbol{x}_t, \tag{2.10b}$$

$$\dot{\boldsymbol{x}}_{\tau|t} = f(\boldsymbol{x}_{\tau|t}, \boldsymbol{u}_{\tau|t}), \qquad \tau \in [0, T], \tag{2.10c}$$

$$g^{\mathrm{s}}_j(\boldsymbol{x}_{\tau|t}, \boldsymbol{u}_{\tau|t}) \leq 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{s}}]},\ \tau \in [0, T], \tag{2.10d}$$

$$g^{\mathrm{o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}) \leq 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{o}}]},\ \tau \in [0, T], \tag{2.10e}$$

$$\boldsymbol{x}_{T|t} \in \mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}_{t+T}), \tag{2.10f}$$

with stage cost $\mathscr{J}^{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r}) = \|\boldsymbol{x} - \boldsymbol{x}^{\mathrm{r}}\|^2_Q + \|\boldsymbol{u} - \boldsymbol{u}^{\mathrm{r}}\|^2_R, Q \succ 0, R \succ 0$ and terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{r}}) = \|\boldsymbol{x} - \boldsymbol{x}^{\mathrm{r}}\|^2_P, P \succ 0$. In this formulation, Q and R are tuning matrices, and P should be suitably computed to ensure stability as given in Section 2.3.2. The optimal solution to (2.10) is denoted by $\boldsymbol{x}^*_{\cdot|t}$ and $\boldsymbol{u}^*_{\cdot|t}$, with $\boldsymbol{x}^*_{\cdot|t}$ being the optimal state trajectory and $\boldsymbol{u}^*_{\cdot|t}$ being the optimal control input trajectory.

TMPC problem (2.10) is applied in a receding-horizon fashion, meaning that it is solved every T^{s} seconds and the optimal control input $\boldsymbol{u}^*_{\cdot|t}$ is applied in open loop during each interval with length T^{s}. Mathematically, the open-loop control actions applied during the interval starting at $t = iT^{\mathrm{s}}, i \in \mathbb{N}$, are given by

$$\boldsymbol{u}_{t+\tau} = \boldsymbol{u}^*_{\tau|t}, \ \tau \in [0, T^{\mathrm{s}}]. \tag{2.11}$$

2.3.2. TMPC PROPERTIES

The advantage of running a TMPC on a mobile robot is that if (2.10) can be solved, it is the optimal solution to the problem. The optimal solution is a feasible solution to (2.10), i.e., one that satisfies all constraints (2.10b)-(2.10f). Furthermore, it minimizes objective (2.10a). If the problem is suitably formulated, the robot will efficiently navigate through the environment by accurately tracking the reference trajectory.

Note that MPC is a trajectory optimization approach. Theoretically, one would like to optimize a suitable trajectory for all future times. However, this is not a tractable problem to solve. Therefore, the MPC horizon is finite. To ensure that using a finite horizon does not lead to unavoidable infeasible problems in the future, so-called terminal ingredients are designed [49]. The terminal ingredients include the terminal control law κ^{f}, terminal cost $\mathscr{J}^{\mathrm{f}}$, and terminal set $\mathscr{X}^{\mathrm{f}}$. The idea behind the terminal ingredients is that one could safely execute the open-loop MPC strategy appended with κ^{f}. This might not be the optimal behavior since (2.10) is not solved every T^{s} seconds, but it would be feasible. If designed suitably, the execution of κ^{f} after $\boldsymbol{u}^*_{[0,T]|t}$ renders $\mathscr{J}^{\mathrm{f}}$ an upper bound for the 'infinite-horizon' cost that is not accounted for in the integral term of (2.10a). Furthermore, if $\mathscr{X}^{\mathrm{f}}$ is suitably designed, this set is positive invariant under κ^{f}. This means that the system is guaranteed to remain within this set. As one can imagine, the terminal ingredients are crucial for the safety guarantees of an MPC scheme. Therefore, they are central to the MPC design.

The discussion on terminal ingredients above holds for classical stabilization MPC types that bring the system to an equilibrium state. Recently, this theory has been extended to tracking dynamic reference trajectories, such as the one defined in Property 1, rendering them useful for mobile robot deployment. As a result, the terminal ingredients all have (2.4) as input argument. In particular, their commonly used expressions are

$$\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r}) = \boldsymbol{u}^{\mathrm{r}} + K(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}), \tag{2.12}$$

for the terminal control law,

$$\mathscr{J}^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{x}^{\mathrm{r}}) = \|\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}\|_P^2, \tag{2.13}$$

for the terminal cost and

$$\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}) = \left\{ \boldsymbol{x} \in \mathbb{R}^{n^{\mathrm{x}}} \;\middle|\; \mathscr{J}^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{x}^{\mathrm{r}}) \leq \alpha^2 \right\}, \tag{2.14}$$

for the terminal set. Thus, $\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r})$ consists of a feed-forward reference input term and feedback gain matrix $K \in \mathbb{R}^{n^{\mathrm{u}} \times n^{\mathrm{x}}}$, correcting the error between the reference state $\boldsymbol{x}^{\mathrm{r}}$ and real state $\boldsymbol{x}$. Furthermore, $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ is a sublevel set of the terminal cost with scaling $\alpha > 0$.

Remark 2. *In general, feedback gain matrix K can be state-dependent or time-dependent; see [38] and the discussion therein. For the sake of simplicity, we assume that K is constant in this guide.*

Important to note is that one does not implement $\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r})$. Instead, the MPC scheme is implemented in a receding-horizon fashion as given by (2.11). This means that the control action is recomputed at times $t = iT^{\mathrm{s}}, i \in \mathbb{N}$ and applied during intervals $[iT^{\mathrm{s}}, (i+1)T^{\mathrm{s}}), i \in \mathbb{N}$. A problem that could arise in practice is that TMPC problem (2.10) becomes infeasible at some point. For example, this happens when the solver experiences numerical problems. This means the solver cannot find a solution satisfying all constraints. In this case, the robot has no known control action to take. The TMPC formulation should satisfy the recursive feasibility property to prevent this issue. This property ensures that, given that a feasible solution to (2.10) is found at time t, a feasible solution can also be constructed for time $t + T^{\mathrm{s}}$ without solving the optimization problem, such that the constraints in (2.10) are satisfied at all future times. Recursive feasibility is considered a necessary condition to ensure safe mobile robot deployment.

Remark 3. *In dynamic environments, the constraints might vary over time. This could break the recursive feasibility property since the previously optimal solution appended by $\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r})$ does not necessarily satisfy the updated constraints. This does not mean the receding-horizon MPC implementation will not find an optimal solution at each time step. This means that we cannot guarantee that a feasible solution will always exist. Therefore, to prove safety in such environments, one must make an assumption about the environment. For example, as long as the robot's trajectory can be predicted well, the moving obstacles will avoid the robot. Alternatively, and possibly more standard, is the assumption that there exist known sets that over-approximate the dynamic obstacles over time [60]. The latter assumption is usually limited to structured environments with predictable object movements or results in conservative plans. Less conservative motion plans are constructed in more commonly adopted approaches such as stochastic MPC, e.g., [69].*

These approaches plan safe trajectories in dynamic environments up to a certain collision probability based on future object trajectory predictions. However, the level of safety guarantees decreases for these schemes.

Given the receding-horizon implementation of TMPC, it is also important to show that, even though the predicted state might come closer to the reference trajectory over the prediction horizon, the closed-loop robot state converges to the reference trajectory. We will call this the trajectory tracking property.

Thus, for safe and efficient mobile robot navigation using MPC, we need to ensure that the TMPC formulation (2.10) is recursively feasible and the closed-loop system converges to the reference trajectory. This is formalized in the following theorem:

Theorem 1. *There exist terminal control law κ^{f}, terminal cost $\mathscr{J}^{\mathrm{f}}$, and terminal set $\mathscr{X}^{\mathrm{f}}$ such that, if* (2.10) *is feasible at $t = 0$, the system under control actions* (2.11) *is recursively feasible, satisfies system constraints* (2.2) *and avoids obstacles for all $t \geq 0$, and tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ asymptotically converges to zero.*

Proof. The proof is this theorem is provided throughout the rest of this section and structured as follows: first, the existence of the terminal ingredients is assumed. Together with a specific candidate solution, they enable proving recursive feasibility and asymptotic convergence to zero of the tracking error. Finally, design choices are described that fulfil all assumptions made to complete the recursive feasibility and trajectory tracking proofs, thereby completing the proof of this theorem. □

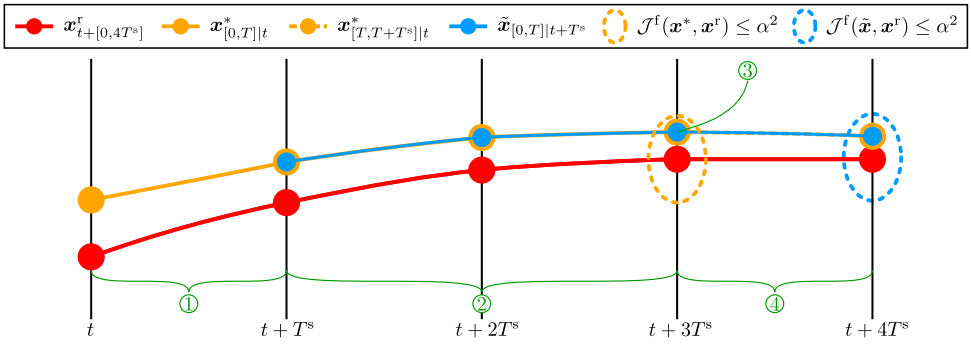


Figure 2.1: Visualization of a 1D reference trajectory $\boldsymbol{x}^{\mathrm{r}}_{t+[0,4T^{\mathrm{s}}]}$, previously optimal solution $\boldsymbol{x}^{*}_{[0,T]|t}$, appended part of previously optimal solution $\boldsymbol{x}^{*}_{[T,T+T^{\mathrm{s}}]|t}$, candidate solution $\tilde{\boldsymbol{x}}_{[0,T]|t+T^{\mathrm{s}}}$, terminal set $\mathscr{J}^{\mathrm{f}}(\boldsymbol{x}^{*},\boldsymbol{x}^{\mathrm{r}}) \leq \alpha^2$ corresponding to $\boldsymbol{x}^{*}_{[0,T]|t}$, and terminal set $\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{x}},\boldsymbol{x}^{\mathrm{r}}) \leq \alpha^2$ corresponding to candidate solution. In this case we set $T = 4T^{\mathrm{s}}$. Note that the terminal sets are visualized as ellipses to clarify their shapes, whereas they should be visualized as vertical lines in this case.

To provide some intuition behind the proofs beforehand, Figure 2.1 shows the optimal solution at time t and the corresponding candidate solution at time $t + T^{\mathrm{s}}$. Thus, the candidate solution covers the prediction horizon of the previously optimal solution shifted by T^{s}. The idea behind the recursive feasibility proof is to show that the candidate satisfies all constraints (2.10b)-(2.10f) at time $t + T^{\mathrm{s}}$. In interval ②, these constraints are

trivially satisfied since the candidate overlaps with the optimal solution at time t. The main thing to show is that once terminal set (2.14) is reached at the end of the prediction horizon at t, ensured by (2.10f), we can append control law (2.12) to the previously optimal solution to obtain a candidate solution that satisfies system and obstacle avoidance constraints (2.10d) and (2.10e), respectively, and terminal set constraint (2.10f). A trivial method to prove recursive feasibility is to show that control law (2.12) renders (2.14) invariant. This guarantees that the candidate satisfies (2.10f). Furthermore, by (2.15b) and (2.15c) in Assumption 1, (2.10d) and (2.10e) are also satisfied in terminal set (2.14).

Given that the candidate is a feasible solution to (2.10) at $t+T^{\mathrm{s}}$, the trajectory tracking proof entails showing that cost (2.10a) is lower for the candidate than for the previously optimal solution, meaning that the robot will converge to the reference trajectory even though (2.10) is not solved again at $t+T^{\mathrm{s}}$. Intuitively, this holds if the cost related to interval ④, shortly cost ④, is lower than terminal cost ③ such that the candidate cost decreases at least by cost ① compared to the cost of the previously optimal solution since cost ② remains the same. This is exactly the condition that is written mathematically in (2.15a) in Assumption 1.

ASSUMPTION ON EXISTENCE TERMINAL INGREDIENTS

Important to note is that the existence of the above-specified terminal ingredients is not guaranteed in general. We will show later how to design them. For now, we assume their existence. This gives the following assumption:

Assumption 1 (Terminal ingredients)**.** *There exist control law* (2.12) *with feedback matrix* $K \in \mathbb{R}^{n^{\mathrm{u}} \times n^{\mathrm{x}}}$*, terminal cost* (2.13) *with terminal cost matrix* $P \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{x}}}$*, and terminal set* (2.14) *with terminal set scaling* $\alpha > 0$*, such that the following properties hold for all* $\boldsymbol{x} \in \mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$*:*

$$\mathscr{J}^{\mathrm{f}}(\boldsymbol{x}^*_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}) \leq -\int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau, \tag{2.15a}$$

$$\left(\boldsymbol{x}, \kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})\right) \in \mathscr{Z}, \tag{2.15b}$$

$$M\boldsymbol{x} \in \mathscr{F}, \tag{2.15c}$$

with sampling time $T^{\mathrm{s}} > 0$*, optimal solution* $(\boldsymbol{x}^*_{\cdot|t}, \boldsymbol{u}^*_{\cdot|t})$ *at* t *for all* $t \geq 0$ *and its extended version using terminal control law* (2.12) $\boldsymbol{x}^*_{T+\tau|t}$ *for* $\tau \in [0, T^{\mathrm{s}}]$*, as defined in Section 2.3.2.*

This assumption is used to complete the trajectory tracking and recursive feasibility proofs in Sections 2.3.2 and 2.3.2. The meaning of the quantities in this assumption will become clearer in the following sections.

CANDIDATE SOLUTION

The goal of defining the candidate solution is to show that there exists a feasible, not necessarily optimal, solution to (2.10) at time $t+T^{\mathrm{s}}$ given that (2.10) is successfully solved at time t. The trivial way to construct such a candidate is to pick the overlapping part with the optimal solution from time t, i.e., the part in prediction interval $\tau \in [T^{\mathrm{s}}, T]$ at t, since this trajectory is known to have satisfied all constraints in the MPC run at time t and overlaps with $\tau \in [0, T-T^{\mathrm{s}}]$ at $t+T^{\mathrm{s}}$. What is left is to define the candidate for

$\tau \in [T - T^{\mathrm{s}}, T]$. This trajectory part satisfies the constraints if we find $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$ with the abovementioned properties.

Let us now mathematically define the candidate. For convenience of notation, we define the appended part of the previously optimal solution, for $\tau \in [T, T + T^{\mathrm{s}}]$, as

$$\boldsymbol{u}^*_{\tau|t} = \kappa^{\mathrm{f}}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}), \tag{2.16}$$

with $\dot{\boldsymbol{x}}^*_{\tau|t}$ given by (2.1) by applying (2.16).

Given these definitions, we can write the candidate for time $t + T^{\mathrm{s}}$ as

$$\tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}} = \boldsymbol{u}^*_{T^{\mathrm{s}}+\tau|t}, \; \tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}} = \boldsymbol{x}^*_{T^{\mathrm{s}}+\tau|t}, \; \tau \in [0, T]. \tag{2.17}$$

The candidate, and thus $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$, is not necessarily implemented in practice. It is only used to prove recursive feasibility and trajectory tracking. One would need this control law if the problem runs infeasible, and one wants the robot to continue executing the open-loop MPC prediction during deployment. However, in that situation, the control action no longer considers any changes in the robot's state or environment, which can be dangerous. In that case, it would be better to stop, check the log to see why the problem was infeasible, and fix it.

Remark 4. *Since the candidate gives the desired recursive feasibility and trajectory tracking properties as shown in the next sections, it would be sufficient to only solve* (2.10) *at* $t = 0$ *and apply the candidate for* $t > 0$*. This reasoning conforms to the arguments presented in [70]: feasibility implies stability. However, the resulting control policy is suboptimal regarding the defined objective function. Therefore, the user can explicitly trade optimality and computational efficiency.*

Although the candidate is not necessarily implemented in practice, it is a suitable warm start for the solver since it is a feasible solution that might be close to the optimal solution. Providing a proper warm start to the solver is vital to speeding up the solve time, as the solver needs fewer iterations to go from the initial point to the optimal solution.

RECURSIVE FEASIBILITY PROOF

To prove recursive feasibility, our task is to show that candidate (2.17) satisfies all constraints (2.10b)-(2.10f) at time $t + T^{\mathrm{s}}$. First, note that, without knowing anything about the terminal ingredients, the candidate solution implies that constraint (2.10b) is satisfied and constraint (2.10c) is satisfied for $\tau \in [0, T]$. That leaves us to show that constraints (2.10d) and (2.10e) are satisfied for $\tau \in [0, T]$ and that the terminal set constraint (2.10f) is satisfied at $\tau = T$.

To show that constraints (2.10d) and (2.10e) are satisfied for $\tau \in [0, T]$, we split this interval up into the half-open interval $\tau \in [0, T - T^{\mathrm{s}})$ and the closed interval $\tau \in [T - T^{\mathrm{s}}, T]$.

System Constraints Satisfaction for $\tau \in [0, T - T^{\mathrm{s}})$

Let us start with the proofs for $\tau \in [0, T - T^{\mathrm{s}})$. For this interval, the following holds for the system constraints $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$g^{\mathrm{s}}_j(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}}) \overset{(2.17)}{=} g^{\mathrm{s}}_j(\boldsymbol{x}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{u}^*_{T^{\mathrm{s}}+\tau|t}) \overset{(2.10d)}{\leq} 0. \tag{2.18}$$

Obstacle Avoidance Constraints Satisfaction for $\tau \in [0, T - T^{\mathrm{s}})$
Similarly, the obstacle avoidance constraints $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$ are satisfied for $\tau \in [0, T - T^{\mathrm{s}})$:

$$g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\tilde{\boldsymbol{p}}_{\tau|t+T^{\mathrm{s}}}) \overset{(2.17)}{=} g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\boldsymbol{p}^{*}_{T^{\mathrm{s}}+\tau|t}) \overset{(2.10e)}{\leq} 0. \tag{2.19}$$

Note that, in contrast to the system constraints, the obstacle avoidance constraints are time-varying. Therefore, we need the following assumption on the obstacle avoidance constraints in order to show that $g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\boldsymbol{p}^{*}_{T^{\mathrm{s}}+\tau|t}) \leq 0$ actually holds:

Assumption 2 (Obstacle avoidance constraints)**.** *The obstacle avoidance constraints should satisfy*

$$\boldsymbol{p}^{*}_{T^{\mathrm{s}}+\tau|t} \in \mathscr{F}_{\tau|t+T^{\mathrm{s}}},\ \tau \in [0, T - T^{\mathrm{s}}], \tag{2.20a}$$

$$\boldsymbol{p}^{*}_{T|t} \in \mathscr{F}_{T|t}. \tag{2.20b}$$

In words, the previously optimal path $\boldsymbol{p}^{*}_{T^{\mathrm{s}}+\tau|t}$ should be contained in the corresponding constraint regions $\mathscr{F}_{\tau|t+T^{\mathrm{s}}}$ applied for $\tau \in [0, T - T^{\mathrm{s}}]$ at $t + T^{\mathrm{s}}$. Similarly, the terminal position $\boldsymbol{p}^{*}_{T|t}$ should be contained in the terminal constraint region $\mathscr{F}_{T|t}$. The latter property ensures that the appended part of the candidate solution, given by (2.16), satisfies the updated obstacle avoidance constraints at time $t + T^{\mathrm{s}}$. This property is used to show terminal constraint satisfaction below. We assume the 'corresponding' constraint regions are constructed using the previously optimal solution.

Terminal Set Constraint Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$
Next, we prove that if (2.10f) is satisfied at time t, the candidate (2.17) satisfies (2.10f) in prediction interval $\tau \in [T - T^{\mathrm{s}}, T]$ for time $t + T^{\mathrm{s}}$. For this proof, we leverage the assumed decrease in $\mathscr{J}^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{x}^{\mathrm{r}})$, given by (2.15a) in Assumption 1. Correspondingly, the following holds for $\tau \in [T - T^{\mathrm{s}}, T]$:

$$\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}) \overset{(2.17)}{=} \mathscr{J}^{\mathrm{f}}(\boldsymbol{x}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}) \overset{(2.15a)}{\leq} \mathscr{J}^{\mathrm{f}}(\boldsymbol{x}_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}) \overset{(2.10f)(2.14)}{\leq} \alpha^{2}. \tag{2.21}$$

System Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$
Satisfaction of system constraints (2.10d) in prediction interval $\tau \in [T - T^{\mathrm{s}}, T]$ is a direct consequence of the combination of terminal set invariance, as shown in the previous section, and invoking (2.15b) in Assumption 1.

Obstacle Avoidance Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$
Similar to the system constraints satisfaction, satisfaction of obstacle avoidance constraints (2.10e) in prediction interval $\tau \in [T - T^{\mathrm{s}}, T]$ is a direct consequence of the combination of terminal set invariance, as shown in the previous section, and invoking (2.15c) in Assumption 1. □

TRAJECTORY TRACKING PROOF

Similar to proving the stability of an autonomous system, proving that nominal system (2.1) tracks reference trajectory (2.4) under closed-loop control law (2.11) follows arguments from Lyapunov theory. A useful analogy to Lyapunov's theory is the energy captured in a mechanical system. In this analogy, the Lyapunov function describes the energy stored in the system. In such a system, the energy decreases over time because of

dissipation effects like friction, meaning that the Lyapunov function value will strictly decrease over time. In the context of MPC, the energy is based on the tracking error, and the dissipation effect that reduces the tracking error is the closed-loop control law (2.11). Thus, a logical choice for the Lyapunov function is to pick the optimal cost $\mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t)$ in (2.10) since this cost indicates how close the predicted trajectory is to the reference trajectory. Ideally, we would like the optimal cost to decrease over time, meaning the tracking error converges to zero.

To derive a bound on the optimal cost, we follow similar steps to the standard descent proof for the candidate cost in [22]:

$$\mathscr{J}_{t+T^\mathrm{s}}^*(\boldsymbol{x}_{t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}}) \overset{\text{(a)}}{\leq} \int_0^T \mathscr{J}^\mathrm{s}(\tilde{\boldsymbol{x}}_{\tau|t+T^\mathrm{s}}, \tilde{\boldsymbol{u}}_{\tau|t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) \tag{2.22a}$$

$$\overset{\text{(b)}}{=} \int_0^{T-T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\tilde{\boldsymbol{x}}_{\tau|t+T^\mathrm{s}}, \tilde{\boldsymbol{u}}_{\tau|t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}+\tau}) d\tau + \int_{T-T^\mathrm{s}}^T \mathscr{J}^\mathrm{s}(\tilde{\boldsymbol{x}}_{\tau|t+T^\mathrm{s}}, \tilde{\boldsymbol{u}}_{\tau|t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) \tag{2.22b}$$

$$\overset{\text{(c)}}{=} \int_{T^\mathrm{s}}^T \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \int_T^{T+T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) \tag{2.22c}$$

$$\overset{\text{(d)}}{=} \int_0^T \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau - \int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \int_T^{T+T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) \tag{2.22d}$$

$$\overset{\text{(e)}}{=} \mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) - \mathscr{J}^\mathrm{f}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^\mathrm{r}_{t+T}) - \int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \int_T^{T+T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) \tag{2.22e}$$

$$\overset{\text{(f)}}{=} \mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) - \int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\tilde{\boldsymbol{x}}_{T|t+T^\mathrm{s}}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) - \mathscr{J}^\mathrm{f}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^\mathrm{r}_{t+T}) + \int_T^{T+T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau \tag{2.22f}$$

$$\overset{\text{(g)}}{=} \mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) - \int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau + \mathscr{J}^\mathrm{f}(\boldsymbol{x}^*_{T^\mathrm{s}+T|t}, \boldsymbol{x}^\mathrm{r}_{t+T^\mathrm{s}+T}) - \mathscr{J}^\mathrm{f}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^\mathrm{r}_{t+T}) + \int_T^{T+T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau \tag{2.22g}$$

$$\overset{\text{(h)}}{\leq} \mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) - \int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau, \tag{2.22h}$$

where (a) is obtained by filling in (2.10a) with the feasible, but not necessarily optimal candidate (2.17), the so-called candidate cost, (b) by splitting the integral term at $T - T^\mathrm{s}$,

(c) by filling in candidate (2.17), (d) by adding and subtracting $\int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau$, (e) by replacing $\int_0^{T} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau$ with $\mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t) - \mathscr{J}^\mathrm{f}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^\mathrm{r}_{t+T})$, (f) by conveniently re-arranging the terms, (g) by (2.17), and (h) by (2.15a) in Assumption 1.

In simple words, (2.15a) in Assumption 1 states that the added candidate cost in prediction interval $\tau \in [T - T^\mathrm{s}, T]$ at $t + T^\mathrm{s}$ should be less than the terminal predicted cost at predicted time T at t. If this is the case, the candidate cost $\mathscr{J}_{t+T^\mathrm{s}}(\boldsymbol{x}_{t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}})$ is at least the stage cost for $\tau \in [0, T^\mathrm{s}]$ at t smaller than the previously optimal cost $\mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t)$. Thus, even without optimizing (2.10) at time $t + T^\mathrm{s}$ and instead applying candidate (2.17), the cost strictly decreases. Given that the optimal cost $\mathscr{J}^*_{t+T^\mathrm{s}}(\boldsymbol{x}_{t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}})$ is upper-bounded by the candidate cost, the optimal cost also strictly decreases.

Thus, (2.15a) in Assumption 1 implies

$$\mathscr{J}^*_{t+T^\mathrm{s}}(\boldsymbol{x}_{t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}}) - \mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t) \leq -\int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau. \tag{2.23}$$

It is not trivial to find a class-$\mathcal{K}_\infty$ function that forms a lower bound to $\int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau$, which would allow us to show that $\int_0^{T^\mathrm{s}} \mathscr{J}^\mathrm{s}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{u}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau > 0$. Hence, we cannot prove stability utilizing standard Lyapunov arguments outlined in [22]. As a convergence proof does not require this bound, we prove convergence instead.

The descent property of the optimal cost (2.23) implies

$$\mathscr{J}^*_{t+T^\mathrm{s}}(\boldsymbol{x}_{t+T^\mathrm{s}}, \boldsymbol{r}_{t+T^\mathrm{s}}) - \mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t) \leq -c^{\mathscr{J},\mathrm{d}} \int_t^{t+T^\mathrm{s}} \|\boldsymbol{x}_\tau - \boldsymbol{x}^\mathrm{r}_\tau\|^2_Q d\tau, \tag{2.24}$$

with constant $c^{\mathscr{J},\mathrm{d}} > 0$. Reversing the sign in (2.24), iterating this inequality from 0 to t with $t \to \infty$, and using the fact that $\mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t)$ is uniformly bounded yields

$$\lim_{t\to\infty} \int_0^t \|\boldsymbol{x}_\tau - \boldsymbol{x}^\mathrm{r}_\tau\|^2_Q d\tau \leq \mathscr{J}^*_t(\boldsymbol{x}_0, \boldsymbol{r}_0) - \lim_{t\to\infty} \mathscr{J}^*_t(\boldsymbol{x}_t, \boldsymbol{r}_t) \leq \mathscr{J}^*_t(\boldsymbol{x}_0, \boldsymbol{r}_0) < \infty. \tag{2.25}$$

By applying Barbalat's lemma [71], we can conclude that the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}^\mathrm{r}_t\|$ asymptotically converges to zero. □

DESIGN TO SATISFY THE ASSUMPTIONS

This section will detail the design of:

- the obstacle avoidance constraints, which need to satisfy Assumption 2;
- the terminal ingredients, which need to satisfy (2.15a) in Assumption 1;
- the reference trajectory design, which needs to satisfy (2.15b) and (2.15c) in Assumption 1.

Obstacle Avoidance Constraints Design

To prove recursive feasibility, we need the obstacle avoidance constraints to satisfy Assumption 2. A trivial way to satisfy this assumption is to generate the obstacle avoidance constraints based on the previously optimal path $\boldsymbol{p}^*_{\cdot|t}$ and ensure that this path is also contained in the corresponding constraint regions.

Remark 5. *Since TMPC (2.10) will be implemented using multiple shooting with discretized dynamics [22], we also define the obstacle avoidance constraints based on piecewise-constant segments between MPC stages. This means that the obstacle avoidance constraints are also piecewise defined according to*

$$\mathscr{F}_{iT^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}} = \mathscr{F}_{iT^{\mathrm{s}}|t+T^{\mathrm{s}}},\ \tau \in (0, T^{\mathrm{s}}],\ i \in \mathbb{N}_{[0,N-1]}, \tag{2.26}$$

where N is the number of discrete predicted stages.

Terminal Ingredients Design

The terminal ingredients design involves computing a suitable combination of terminal control law (2.12), terminal cost 2.13, and terminal set (2.14) that satisfy terminal cost decrease (2.15a) in Assumption 1. This design involves finding a suitable combination of matrices P and K and constant $\alpha > 0$.

To design P and K, recent work [38] proposes a linear matrix inequality (LMI) to ensure direct satisfaction of Assumption 1:

$$\big(A(\boldsymbol{r}) + B(\boldsymbol{r})K(\boldsymbol{r})\big)^\top P(\boldsymbol{r}) + P(\boldsymbol{r})\big(A(\boldsymbol{r}) + B(\boldsymbol{r})K(\boldsymbol{r})\big) + \sum_{j=1}^{n^{\mathrm{x}}+n^{\mathrm{u}}} \frac{\partial P(\boldsymbol{r})}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}_j + Q^\epsilon + K(\boldsymbol{r})RK(\boldsymbol{r}) \preceq 0, \tag{2.27}$$

with Jacobians

$$A(\boldsymbol{r}) := \left.\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{r}}, \quad B(\boldsymbol{r}) := \left.\frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right|_{\boldsymbol{r}}, \tag{2.28}$$

$Q^\epsilon := Q + \epsilon I^{n^{\mathrm{x}}}$, and $\epsilon > 0$. See [38] for the corresponding proof.

In particular, this LMI ensures a system property called local incremental stabilizability. See [72], [73] for more details. Note that P and K are parameterized based on local values $\boldsymbol{r}$ in the state-input space of the reference trajectory. This is where the word 'local' comes from. It suffices to show that certain properties hold locally in the state-input space to ensure tracking of reference trajectories in the complete state-input space. However, note that the results only hold for sufficiently small values of terminal set scaling α such that Jacobians (2.28) are accurate enough to describe the nonlinear system behavior. The word 'incremental' refers to the fact that we consider the stabilizability property of the system trajectory that evolves close to the reference trajectory.

As mentioned before, terminal cost (2.13) is an upper bound of the 'infinite-horizon' cost that we would ideally like to minimize. Therefore, the goal is to find the minimum-valued matrix $P(\boldsymbol{r})$ and corresponding matrix $K(\boldsymbol{r})$ satisfying LMI (2.27). Correspondingly, we can formulate the following semidefinite program (SDP):

$$\min_{X,Y} \ -\log\det X_{\min}, \tag{2.29a}$$

$$\text{s.t. } X(\boldsymbol{r}) \succeq 0, \tag{2.29b}$$

$$\begin{bmatrix} A(\boldsymbol{r})X(\boldsymbol{r}) + B(\boldsymbol{r})Y(\boldsymbol{r}) + \big(A(\boldsymbol{r})X(\boldsymbol{r}) + B(\boldsymbol{r})Y(\boldsymbol{r})\big)^\top - \dot{X}(\boldsymbol{r}) & \big(Q^{\epsilon\frac{1}{2}}X(\boldsymbol{r})\big)^\top & \big(R^{\frac{1}{2}}Y(\boldsymbol{r})\big)^\top \\ Q^{\epsilon\frac{1}{2}}X(\boldsymbol{r}) & -I^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}\times n^{\mathrm{u}}} \\ R^{\frac{1}{2}}Y(\boldsymbol{r}) & 0^{n^{\mathrm{u}}\times n^{\mathrm{x}}} & -I^{n^{\mathrm{u}}} \end{bmatrix} \preceq 0, \tag{2.29c}$$

$$X_{\min} \preceq X(\boldsymbol{r}),\ \forall \boldsymbol{r} \in \bar{\mathcal{Z}},,$$

with $\dot{X}(\boldsymbol{r}) \coloneqq \frac{\partial X(\boldsymbol{r})}{\partial \boldsymbol{x}^{\mathrm{r}}} \dot{\boldsymbol{x}}^{\mathrm{r}}$ and LMI (2.29c) being the direct result of taking the Schur complement of LMI (2.27) and applying the following coordinate transform:

$$P(\boldsymbol{r}) = X(\boldsymbol{r})^{-1},\ K(\boldsymbol{r}) = Y(\boldsymbol{r})P(\boldsymbol{r}). \tag{2.30}$$

Note that $\tilde{\mathcal{Z}}$ represents the continuous state-input space of the reference trajectory. Therefore, SDP (2.29) is semi-infinite, and we need to grid or convexify $\tilde{\mathcal{Z}}$ to solve it. For solving the SDP, the states and inputs in $\boldsymbol{r}$ that do not appear in the Jacobians (2.28) can be set to zero, we can take the vertices for the ones that appear linearly, and the ones that appear nonlinearly need to be gridded.

Remark 6. *Since the computation time of generating LMI (2.29c) can become significantly large, a solution is to solve (2.29) with a coarse grid of points and evaluate the result at a denser grid to check the validity of the result.*

Given this method to compute $P(\boldsymbol{r})$ and $K(\boldsymbol{r})$, the only thing left is to design a suitable value for terminal set scaling α. Theoretically, one could set $\alpha = 0$, such that (2.10f) becomes an equality constraint. However, this unnecessarily restricts the solver from finding a solution. This might lead to infeasibility, either because a solution satisfying this equality constraint is non-existent or because the problem is numerically infeasible. Therefore, the idea is to find a non-conservative maximum value for α. Recall that (2.14) is centered around the reference trajectory, and the reference trajectory satisfies tightened system and obstacle avoidance constraints (2.7) and (2.9), respectively. Given this information, α can only grow as large as the tightening applied to the original system and obstacle avoidance constraints.

Let us assume the reference trajectory is only subject to tightened system constraints and is constrained by the following polytopic set: $\tilde{\mathcal{Z}} = \left\{ \boldsymbol{r} \in \mathbb{R}^{n^{\mathrm{u}}+n^{\mathrm{x}}} \;\middle|\; L_j^{\mathrm{r,s}} \boldsymbol{r} \leq l_j^{\mathrm{r,s}},\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]} \right\}$. The tightening equals $l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}, \boldsymbol{r} \in \tilde{\mathcal{Z}}, j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$. As a result, α can be computed by solving the following linear program [38], which is proven in Appendix A.1:

$$\begin{aligned} \alpha^* = & \max_{\alpha} \alpha, && \text{(2.31a)} \\ & \text{s.t. } \left\| P(\boldsymbol{r})^{-\frac{1}{2}} \left[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}} \right] L_j^{\mathrm{r,s}\top} \right\| \alpha \leq l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}, && \text{(2.31b)} \\ & \qquad \forall \boldsymbol{r} \in \tilde{\mathcal{Z}},\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. && \text{(2.31c)} \end{aligned}$$

Naturally, this result can also be extended to include obstacle avoidance constraints.

Note that using this expression, we can either compute the value of α based on the tightening of the system constraints for the reference trajectory or vice versa. Therefore, we need more details on the system and obstacle avoidance constraints before finishing the design of α. This will be described in the next section.

Reference Trajectory Design

The reference trajectory needs to satisfy tightened system constraints (2.7) and tightened obstacle avoidance constraints (2.9). These constraint sets need to be constructed such that the system and obstacle avoidance constraints are satisfied in terminal set (2.14) according to (2.15b) and (2.15c) in Assumption 1.

Given the fact that system constraints (2.2) are continuously differentiable and defined on compact set $\mathcal{Z}$, they are Lipschitz continuous. Therefore, we can express them as

$$g_j^{\mathrm{s}}(\boldsymbol{x}', \boldsymbol{u}') - g_j^{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{u}) \leq c_j^{\mathrm{s}} \alpha, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.32}$$

for some $\boldsymbol{x}' \neq \boldsymbol{x} \in \mathbb{R}^{n^{\mathrm{x}}}$ and $\boldsymbol{u}' \neq \boldsymbol{u} \in \mathbb{R}^{n^{\mathrm{u}}}$ with Lipschitz constants $c_j^{\mathrm{s}}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$. As a result, if we set the tightened system constraints of the reference trajectory as

$$g_j^{\mathrm{r,s}}(\boldsymbol{x}, \boldsymbol{u}) \leq g_j^{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{u}) + c_j^{\mathrm{s}} \alpha, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.33}$$

we get a strict equality in (2.31b) with

$$c_j^{\mathrm{s}} = \| P(\boldsymbol{r})^{-\frac{1}{2}} \left[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}} \right] L_j^{\mathrm{r,s}\top} \|, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{2.34}$$

Equivalently, if we set the upper bound of the tightened obstacle avoidance constraints of the reference trajectory as

$$g^{\mathrm{r,o}}(\boldsymbol{p}) \leq g^{\mathrm{o}}(\boldsymbol{p}) + c^{\mathrm{o}} \alpha, \tag{2.35}$$

we can replace $\left[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}} \right] L_j^{\mathrm{r,s}\top}$ by $M^\top$ to obtain the following expression for c^{o}:

$$c^{\mathrm{o}} = \| P(\boldsymbol{r})^{-\frac{1}{2}} M^\top \|. \tag{2.36}$$

Given this design, we choose a value for α that results in a desired tightening of either of the system constraints or the obstacle avoidance constraints. A natural choice for mobile robots is to decide on the maximum distance between reference trajectory and obstacles. Let us call this quantity $d_{\max}$. Then, we know that $c^{\mathrm{o}} \alpha = d_{\max}$, which results in the following expression for the terminal set scaling:

$$\alpha = \frac{d_{\max}}{c^{\mathrm{o}}}. \tag{2.37}$$

By solving (2.29) some system constraints might be tightened more than others, leading to empty feasible sets. This issue can be avoided by including the system constraints tightening constants in (2.29a) and setting their penalties equal to the inverse of the space in the corresponding system constraint direction $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$. This way, the solver is encouraged to tighten the system constraints equally. Working out Lipschitz inequality on the system constraints (2.33) using $\alpha = \| \boldsymbol{x} - \boldsymbol{x}^{\mathrm{r}} \|_{P(\boldsymbol{r})}$ and taking the Schur complement according to [74, p. 29] results in the following LMI:

$$\begin{bmatrix} c_j^{\mathrm{s}2} & L_j^{\mathrm{s}} \begin{bmatrix} Y \\ X \end{bmatrix} \\ (L_j^{\mathrm{s}} \begin{bmatrix} Y \\ X \end{bmatrix})^\top & X \end{bmatrix} \succeq 0, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{2.38}$$

This LMI is included in the SDP formulation in Chapter 3. In that formulation, the state- and input-dependency of P and K are also removed, so the results hold in the complete state-input space, not just locally. Therefore, the constant ϵ from [38], that

accounts for the Lipschitz continuity of Jacobians (2.28), can be removed. The advantage of including the state- and input-dependency is the reduced conservatism, i.e., the terminal cost is smaller and the terminal set is larger. This results in a smaller tracking error of the closed-loop system. However, the increased conservatism was not a major concern in the work of Chapter 3 and resulted in a simpler formulation.

2.3.3. CONCLUDING REMARKS

In conclusion, the design of the obstacle avoidance constraints, terminal ingredients, and reference trajectory as presented above results in the satisfaction of Assumptions 1 and 2. These assumptions enable the recursive feasibility and trajectory tracking proofs, thereby proving Theorem 4. Thus, TMPC scheme (2.10) is an effective tool for providing safety guarantees for tracking dynamically feasible reference trajectories with a mobile robot described by nominal dynamics (2.1).

Note that the solver will only find a feasible solution to (2.10) if the system starts sufficiently close to the reference trajectory such that terminal set constraint (2.10f) is satisfied.

In practice, a physical robot is subject to dynamical disturbances that could be impossible to model. To ensure safe mobile robot navigation despite these disturbances, we need to know how big the effects of the disturbances on the dynamics are and account for that in the MPC design. This is the topic of Section 2.4 in which the disturbed robot dynamics and a corresponding RMPC design are provided.

2.4. ROBUST MPC FOR TRAJECTORY TRACKING WITH DISTURBANCES

The goal of this section is to introduce the RMPC scheme that will be used to track reference trajectories satisfying Property 1 when the system is subject to disturbances. The description of the system dynamics is provided in Section 2.4.1. Like Section 2.3.1, Section 2.4.2 states the optimization problem and closed-loop control law. Then, Section 2.4.3 builds intuition for RMPC and explains the required elements in the formulation and related properties. Finally, Section 2.4.4 concludes the RMPC formulation.

2.4.1. DISTURBED MOBILE ROBOT DESCRIPTION

The disturbed system dynamics are given by

$$\dot{\boldsymbol{x}} = f^{\mathrm{w}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) := f(\boldsymbol{x}, \boldsymbol{u}) + E\boldsymbol{w}, \tag{2.39}$$

with the same properties as the nominal system (2.1), disturbance $\boldsymbol{w} \in \mathbb{R}^{n^{\mathrm{w}}}$, and disturbance selection matrix $E \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{w}}}$. From now on, to make a distinction between the nominal and disturbed system, let us denote nominal and disturbed states by $\boldsymbol{z}$ and $\boldsymbol{x}$, respectively.

By recording data one can compute a bounding box with lower bound $\underline{\boldsymbol{w}}_i^{\mathrm{rec}}$ and upper bound $\bar{\boldsymbol{w}}_i^{\mathrm{rec}}$ for each dimension $i \in \mathbb{N}_{[1,n^{\mathrm{w}}]}$ of the recorded disturbance values. Let us denote the offset for each dimension i by $\boldsymbol{w}_i^{\mathrm{b}} := \frac{\underline{\boldsymbol{w}}_i^{\mathrm{rec}} + \bar{\boldsymbol{w}}_i^{\mathrm{rec}}}{2}$. Then, the disturbance set

centered around $\mathbf{0}$ is given by

$$\mathcal{W} := \left\{ \boldsymbol{w} \in \mathbb{R}^{n^{\mathrm{w}}} \;\middle|\; \underline{\boldsymbol{w}}_i^{\mathrm{rec},0} \leq \boldsymbol{w}_i \leq \bar{\boldsymbol{w}}_i^{\mathrm{rec},0},\ i \in \mathbb{N}_{[1,n^{\mathrm{w}}]} \right\}, \tag{2.40}$$

with $\underline{\boldsymbol{w}}_i^{\mathrm{rec},0} := \underline{\boldsymbol{w}}_i^{\mathrm{rec}} - \boldsymbol{w}_i^{\mathrm{b}}$ and $\bar{\boldsymbol{w}}_i^{\mathrm{rec},0} := \bar{\boldsymbol{w}}_i^{\mathrm{rec}} - \boldsymbol{w}_i^{\mathrm{b}}$, such that the recorded disturbance values are all contained in set $\boldsymbol{w}^{\mathrm{b}} \oplus \mathcal{W}$. Correspondingly, we write the disturbance values in the next sections as

$$\boldsymbol{w} := \boldsymbol{w}^{\mathrm{b}} + \boldsymbol{w}^{0} \in \boldsymbol{w}^{\mathrm{b}} \oplus \mathcal{W}. \tag{2.41}$$

Model bias $\boldsymbol{w}^{\mathrm{b}}$ can be included in the nominal prediction model to account for the static offset of disturbances as described in the next section.

2.4.2. RMPC FORMULATION

Consider the following RMPC formulation for tracking reference trajectory (2.4):

$$\mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) = \min_{\boldsymbol{z}_{\cdot|t}, \boldsymbol{v}_{\cdot|t}} \ \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}_{T|t}, \boldsymbol{x}_{t+T}^{\mathrm{r}}) + \int_0^T \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}, \boldsymbol{r}_{t+\tau})\, d\tau, \tag{2.42a}$$

$$\text{s.t. } \boldsymbol{z}_{0|t} = \boldsymbol{x}_t, \tag{2.42b}$$

$$s_\tau = 0, \tag{2.42c}$$

$$\dot{\boldsymbol{z}}_{\tau|t} = f(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}) + E\boldsymbol{w}^{\mathrm{b}}, \qquad \tau \in [0, T], \tag{2.42d}$$

$$\dot{s}_\tau = -\rho s_\tau + \bar{w}, \qquad \tau \in [0, T], \tag{2.42e}$$

$$g_j^{\mathrm{s}}(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}) + c_j^{\mathrm{s}} s_\tau \leq 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{s}}]},\ \tau \in [0, T], \tag{2.42f}$$

$$g_{j,\tau|t}^{\mathrm{o}}(M\boldsymbol{z}_{\tau|t}) + c^{\mathrm{o}} s_\tau \leq 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{o}}]},\ \tau \in [0, T], \tag{2.42g}$$

$$(\boldsymbol{z}_{T|t}, s_\tau) \in \mathscr{X}^{\mathrm{f}}(\boldsymbol{x}_{t+T}^{\mathrm{r}}), \tag{2.42h}$$

with stage cost $\mathscr{J}^{\mathrm{s}}(\boldsymbol{z}, \boldsymbol{v}, \boldsymbol{r}) = \|\boldsymbol{z} - \boldsymbol{x}^{\mathrm{r}}\|_Q^2 + \|\boldsymbol{v} - \boldsymbol{u}^{\mathrm{r}}\|_R^2, Q \succ 0, R \succ 0$ and terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}}) = \|\boldsymbol{z} - \boldsymbol{x}^{\mathrm{r}}\|_P^2, P \succ 0$. In this formulation, Q and R are tuning matrices and P should be suitably computed to show practical asymptotic convergence, see Section 2.4.3. Note the inclusion of model bias $\boldsymbol{w}^{\mathrm{b}}$ in nominal system dynamics (2.42d) such that the model best describes the minimum and maximum disturbance bounds.

Similar to the TMPC formulation in Section 2.3, the RMPC formulation is applied in a receding-horizon fashion. In this case, the control law for the closed-loop system is defined as follows for $t \geq 0$:

$$\boldsymbol{u}_{t+\tau} = \kappa(\boldsymbol{x}_{t+\tau}, \boldsymbol{z}_{\tau|t}^*, \boldsymbol{v}_{\tau|t}^*) := \boldsymbol{v}_{\tau|t}^* + \kappa^{\delta}(\boldsymbol{x}_{t+\tau}, \boldsymbol{z}_{\tau|t}^*),\ \tau \in [0, T^{\mathrm{s}}], \tag{2.43}$$

with control law $\kappa(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) : \mathscr{X} \times \mathscr{Z} \to \mathbb{R}^{n^{\mathrm{u}}}$ including feedback law $\kappa^{\delta}(\boldsymbol{x}, \boldsymbol{z}) : \mathscr{X}^2 \to \mathbb{R}^{n^{\mathrm{u}}}$ that will be defined later according to the design presented in this section.

2.4.3. RMPC PROPERTIES

If the nominal system (2.1) is subject to a predictable disturbance, we could adjust (2.1) to incorporate the corresponding disturbance dynamics and apply the TMPC described in the previous section. However, in general, disturbances are not or only partially predictable. The predictable part could be learned with a learning-based method such as a

GP, see for example [42], [43], [75]. However, in almost all cases, there will be a part of the dynamics that cannot be predicted. In that case, we still need to be able to guarantee that operating our MPC scheme on the robot is safe.

Then why not execute the TMPC from Section 2.3 and hope for the best? Let us consider the scenario where the reference trajectory is computed using the same constraints as the TMPC. At some time t, the state might exactly overlap with the reference state, i.e., $\boldsymbol{x}_t = \boldsymbol{x}_t^{\mathrm{r}}$. However, the worst-case disturbance happens during interval $[t, t+T^{\mathrm{s}}]$. This causes the state to exceed the constraint boundary. Suddenly, the problem has become infeasible, and there is no structured way to recover, i.e., there does not exist a suitable candidate solution that could be applied in an open loop. From the analysis above, we can conclude that if the reference trajectory is generated far enough away from the constraints, i.e., the tightening $c_j^{\mathrm{s}}\alpha$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ or $c^{\mathrm{o}}\alpha$ is sufficiently large, we can prevent the occurrence of this problem. This is the exact reason why the experiments on the real robot using the TMPC scheme presented in Chapter 3 were successful despite explicit robust design: the distance $d_{\max}$ to the obstacle avoidance constraints was tuned sufficiently well to avoid the obstacles and make sure the system constraints were satisfied at all times in the presence of disturbances.

This is not the complete story, though. Consider the scenario where the reference trajectory is generated far enough from the constraint boundaries. And let us again consider the positive scenario where $\boldsymbol{x}_t = \boldsymbol{x}_t^{\mathrm{r}}$. At t, the TMPC would make a nominal prediction according to dynamics (2.1) and execute the control law in a closed loop. However, during interval $[t, t+T^{\mathrm{s}}]$, the worst-case disturbance impacts the system. So, instead of continuing to track the reference, the state moves away from the reference. The TMPC did not anticipate this. At time $t+T^{\mathrm{s}}$, the TMPC optimizes a nominal predicted trajectory again from the last received sample of the disturbed state. By executing the optimal input during the interval $[t+T^{\mathrm{s}}, t+2T^{\mathrm{s}}]$, the system will move towards the reference trajectory again. Thus, even without explicitly considering disturbances, the TMPC might be able to track (2.4). Of course, the tracking is not perfect, but at least the system stays close to the reference trajectory. This feedback property of TMPC is called inherent robustness [76]. It is a property that most practical MPC implementations rely on, and that might be sufficient to track the reference trajectory.

Inherent robustness alone might not be sufficient to ensure the robot's safety. For example, another disturbance may impact the system during $[t+T^{\mathrm{s}}, t+2T^{\mathrm{s}}]$ and, despite the optimal control effort of the MPC, moves the system further away from the reference. In this case, the closed-loop system is called unstable. In other words, the MPC is not 'strong' enough to attenuate the impact of the disturbances. This notion of strongness is a property that is commonly called contractivity. In other words, the difference between the actual and the reference trajectory should contract with a minimum rate. How close the actual system comes to the reference depends on the contraction rate, usually denoted by ρ, and the worst-case impact of the disturbances, usually denoted by $\bar{w}$. These are the quantities you can also find in RMPC formulation (2.42).

A common approach to improve the disturbance attenuation capability of the MPC and ensure that the closed-loop system stays close to the reference trajectory is to design a feedback law executed at a higher frequency than the sampling frequency of the MPC. This way, the impact of disturbances can be diminished before the MPC can do so at the

discrete sampling moments. Similar to the design of the terminal control law (2.12) that renders terminal set (2.14) invariant, one could design the feedback law to render a so-called tube around the reference trajectory invariant to the disturbances. This forms the basis for a popular class of RMPC methods: tube MPC [37], [77]–[79].

Tube MPC is designed to tighten all the constraints by a rigid tube size such that the closed-loop system satisfies the actual constraints if the nominal predicted trajectory satisfies the tightened constraints. Specifically, in tube MPC, the complete predicted trajectory is optimized, including initial state constraint (2.42b). This increases computational complexity but gives a trivial expression for the terminal set [80].

Central to the design of tube MPC schemes is to find a suitable expression for the tube. A common approach is to design a feedback law $\kappa^\delta(\boldsymbol{x},\boldsymbol{z})$ that renders a sublevel set of incremental Lyapunov function $V^\delta(\boldsymbol{x},\boldsymbol{z})$ invariant to the disturbances acting on the system actuated by closed-loop control law (2.43). Such combination of feedback law in $\kappa^\delta(\boldsymbol{x},\boldsymbol{z})$ and $V^\delta(\boldsymbol{x},\boldsymbol{z})$ exists if the system satisfies the following assumption [72], [81]:

Assumption 3 (Incremental stabilizability)**.** *There exist a feedback law $\kappa^\delta(\boldsymbol{x},\boldsymbol{z}) : \mathscr{X}^2 \to \mathbb{R}^{n^{\mathrm{u}}}$, an incremental Lyapunov function $V^\delta(\boldsymbol{x},\boldsymbol{z}) : \mathscr{X}^2 \to \mathbb{R}_{\geq 0}$ that is continuously differentiable and satisfies $V^\delta(\boldsymbol{z},\boldsymbol{z}) = 0, \forall \boldsymbol{z} \in \mathscr{X}$, parameters $c^{\delta,\mathrm{l}}, c^{\delta,\mathrm{u}}, \rho > 0$, and Lipschitz constants $c_j^{\mathrm{s}} > 0, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and $c^{\mathrm{o}} > 0$, such that the following properties hold for all $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}) \in \mathscr{X} \times \mathscr{Z}$:*

$$c^{\delta,\mathrm{l}}\|\boldsymbol{x}-\boldsymbol{z}\|^2 \leq V^\delta(\boldsymbol{x},\boldsymbol{z}) \leq c^{\delta,\mathrm{u}}\|\boldsymbol{x}-\boldsymbol{z}\|^2, \tag{2.44a}$$

$$g_j^{\mathrm{s}}\big(\boldsymbol{x},\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})\big) - g_j^{\mathrm{s}}(\boldsymbol{z},\boldsymbol{v}) \leq c_j^{\mathrm{s}}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})},\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.44b}$$

$$g^{\mathrm{o}}(M\boldsymbol{x}) - g^{\mathrm{o}}(M\boldsymbol{z}) \leq c^{\mathrm{o}}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}, \tag{2.44c}$$

$$\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z}) \leq -2\rho V^\delta(\boldsymbol{x},\boldsymbol{z}), \tag{2.44d}$$

with $\dot{\boldsymbol{x}} = f\big(\boldsymbol{x},\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}}$ and $\dot{\boldsymbol{z}} = f(\boldsymbol{z},\boldsymbol{v}) + E\boldsymbol{w}^{\mathrm{b}}$. Furthermore, the following norm-like inequality holds $\forall \boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3 \in \mathbb{R}^{n^{\mathrm{x}}}$:

$$\sqrt{V^\delta(\boldsymbol{x}_1,\boldsymbol{x}_3)} \leq \sqrt{V^\delta(\boldsymbol{x}_1,\boldsymbol{x}_2)} + \sqrt{V^\delta(\boldsymbol{x}_2,\boldsymbol{x}_3)}. \tag{2.45}$$

In other words, we want to find a squared incremental Lyapunov function $V^\delta(\boldsymbol{x},\boldsymbol{z})$ that quantifies the difference between disturbed trajectory $\boldsymbol{x}$ and nominal, or reference, trajectory $\boldsymbol{z}$. $V^\delta(\boldsymbol{x},\boldsymbol{z})$ is lower- and upper-bounded by (2.44a), that contracts at least with rate 2ρ according to (2.44d), and satisfies norm-like inequality (2.45), such that the system and obstacle avoidance constraints are Lipschitz-bounded by a constant multiplied with the sqrt of $V^\delta(\boldsymbol{x},\boldsymbol{z})$. The meaning and purpose of the above equations will become more clear later in this section.

Remark 7. *The superscript δ denotes the incremental property of the Lyapunov function, i.e., the fact that the Lyapunov function describes the difference between two time-varying trajectories x and z that are close to each other.*

Remark 8. *When comparing the expressions in Assumption 3 with [73, Ass. 1], note that V^δ contains a different number of arguments. In general, one can construct V^δ that also*

depends on the nominal input $\boldsymbol{v}$. However, here, we do not leverage this property and leave the additional argument out for notational convenience.

Thus, if we can upper bound $\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}$, we know that we can tighten the system and obstacle avoidance constraints using (2.44b) and (2.44c) to ensure that the open-loop execution of control law (2.43) guarantees that the real system trajectory $\boldsymbol{x}$ satisfies the non-tightened constraints.

Note that applying the sqrt operator to (2.44d) gives the following upper bound on the derivative of $\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}$:

$$\frac{d}{dt}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})} \leq -\rho\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}. \tag{2.46}$$

This upper bound evolves for $t,\tau \geq 0$ as

$$\sqrt{V^\delta(\boldsymbol{x}_{t+\tau},\boldsymbol{z}_{t+\tau})} \leq e^{-\rho\tau}\sqrt{V^\delta(\boldsymbol{x}_t,\boldsymbol{z}_t)}. \tag{2.47}$$

In other words, if the trajectories $\boldsymbol{x}$ and $\boldsymbol{z}$ deviate from each other, they will contract over time according to (2.47). Correspondingly, the error

$$\boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{z} \tag{2.48}$$

exponentially converges to zero. As mentioned above, this is not the case if disturbances act on the system. Therefore, we would like to find an expression for (2.46) that involves the impact of disturbances $\boldsymbol{w}$. To this end, let us first define the following property that holds with the definition of disturbance set (2.40):

Property 2. *There exists a constant $\bar{w} > 0$ such that for any $(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}) \in \mathcal{X}\times\mathcal{Z}$ and any $\boldsymbol{w} \in \boldsymbol{w}^\mathrm{b} \oplus \mathcal{W}$, it holds that*

$$\frac{d}{dt}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})} \leq -\rho\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})} + \bar{w}, \tag{2.49}$$

with $\dot{\boldsymbol{x}} = f\big(\boldsymbol{x},\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})\big) + E\boldsymbol{w}$ and $\dot{\boldsymbol{z}} = f(\boldsymbol{z},\boldsymbol{v}) + E\boldsymbol{w}^\mathrm{b}$.

Consequently, if we denote the upper bound of $\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}$, also called the tube size, by s, i.e.,

$$\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})} \leq s, \tag{2.50}$$

we can write the time-derivative of s over time as

$$\dot{s} = -\rho s + \bar{w}. \tag{2.51}$$

Rewriting (2.51) as a function of time gives

$$s_t = (1 - e^{-\rho t})\frac{\bar{w}}{\rho}, \tag{2.52}$$

and taking the limit $t \to \infty$ results in a maximum tube size of

$$\bar{s} = \frac{\bar{w}}{\rho}. \tag{2.53}$$

Thus, s does not grow unbounded, and the resulting tube size (2.53) is suitable for tightening the constraints in a tube MPC formulation.

In this work, we do not leverage tube MPC, but a slightly different variant: constraint tightening MPC [59], [82]–[84]. This is the formulation used in (2.42). The idea behind constraint tightening MPC is that at the current time t, in contrast to tube MPC, the initial predicted state $\boldsymbol{z}_{0|t}$ equals the measured state $\boldsymbol{x}_t$ (2.42b) and the initial tube size is set to zero (2.42c). From there, given a proper design of closed-loop control law (2.43), the predicted uncertainty grows according to (2.51), which is reflected in (2.42e), with contraction rate ρ and worst-case disturbance impact $\bar{w}$. The growing tube size is used to tighten the system constraints in (2.42f), using system constraints tightening constants c_j^{s}, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$, and obstacle avoidance constraints (2.42g), using obstacle avoidance constraints tightening constant c^{o}. Since the tube size does not grow unbounded as given in (2.53), $\bar{s}$ can be used to construct terminal set constraint (2.42h) such that the system stays within the terminal set for all future times given a suitable terminal control law.

Remark 9. *As will be shown later, $\bar{s}$ can be used to tighten the constraints of the reference trajectory, similar to the usage of the terminal set scaling α for tightening the constraints of the reference trajectory in the TMPC section.*

Similar to Section 2.3.2, for safe and efficient mobile robot navigation using MPC subject to disturbances, we need to ensure that the RMPC formulation (2.42) is recursively feasible and the closed-loop system converges to the reference trajectory. This is formalized in the following theorem:

Theorem 2. *There exist terminal control law κ^{f}, terminal cost $\mathscr{J}^{\mathrm{f}}$, and terminal set $\mathscr{X}^{\mathrm{f}}$ such that, if (2.42) is feasible at $t = 0$, closed-loop system (2.43) is recursively feasible, satisfies system constraints (2.2) and avoids obstacles for all $t \geq 0$, and, on average, the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ gets small over time.*

Proof. The proof is this theorem is provided throughout the rest of this section and structured as follows: first, the existence of the terminal ingredients is assumed. Together with a specific candidate solution, they enable proving recursive feasibility and that the tracking error gets small over time, on average. Finally, design choices are described that fulfil all assumptions made to complete the recursive feasibility and trajectory tracking proofs, thereby completing the proof of this theorem. □

To provide intuition for these proofs beforehand, similar to Section 2.3.2, Figure 2.2 shows the optimal solution at time t and the corresponding candidate solution for time $t + T^{\mathrm{s}}$. Note that, in contrast to the TMPC case, we cannot just take the tail part of the previously optimal solution as a candidate and append it for prediction interval ④ since the system is subject to disturbances $\boldsymbol{w}$ during interval ①. Instead, we measure the disturbed state at $t + T^{\mathrm{s}}$ and leverage the suitably designed feedback term in the control law to steer the candidate back to the previously optimal trajectory. Since the tube size, or equivalently, the constraints tightening in (2.42f) and (2.42g), increases over the horizon, the constraints in interval ② of the candidate are tightened less than the constraints in the same interval of the previously optimal solution. The visual interpretation is that the blue tube is always contained in the orange tube. As a result, it can be shown that

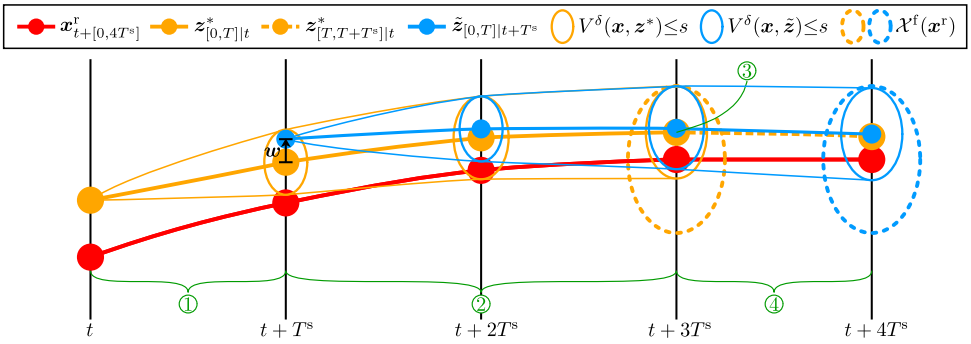


Figure 2.2: Visualization of a 1D reference trajectory $\boldsymbol{x}^{\mathrm{r}}_{t+[0,4T^{\mathrm{s}}]}$, previously optimal solution $\boldsymbol{z}^{*}_{[0,T]|t}$, appended part of previously optimal solution $\boldsymbol{z}^{*}_{[T,T+T^{\mathrm{s}}]|t}$, candidate solution $\tilde{\boldsymbol{z}}_{[0,T]|t+T^{\mathrm{s}}}$, tube $V^{\delta}(\boldsymbol{x},\boldsymbol{z}^{*}) \leq s$ around $\boldsymbol{z}^{*}_{[0,T]|t}$, tube $V^{\delta}(\boldsymbol{x},\tilde{\boldsymbol{z}}) \leq s$ around $\tilde{\boldsymbol{z}}_{[0,T]|t+T^{\mathrm{s}}}$, and terminal sets $\mathcal{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ around the reference trajectory corresponding to the previously optimal and the candidate solution. Again, we set $T = 4T^{\mathrm{s}}$. Compared to Figure 2.1, $\boldsymbol{w}$ impacts the system during interval ①. As a result, $\tilde{\boldsymbol{z}}_{0|t+T^{\mathrm{s}}} \neq \boldsymbol{z}^{*}_{T^{\mathrm{s}}|t}$. Note that the tubes and terminal sets are visualized as ellipses to clarify their shapes, whereas they should actually be visualized as vertical lines in this case.

the impact of the disturbance during interval ① sufficiently contracts by the feedback term in the candidate control law such that the candidate also satisfies the tightened constraints in interval ②. The same holds for a suitably designed terminal set. Furthermore, if we append the candidate with terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r})$ in interval ④, where the nominal term equals $\boldsymbol{u}^{\mathrm{r}}$ and the feedback term steers the candidate towards $\boldsymbol{x}^{\mathrm{r}}$, it can be shown that a suitably designed terminal set is invariant to this control law. Therefore, the candidate also satisfies (2.42h). If the system and obstacle constraints are also satisfied in the terminal set, the satisfaction of (2.42f) and (2.42g) in interval ④ trivially follows from the invariance property of the terminal set. Thus, a feasible solution at t implies that a feasible solution at $t + T^{\mathrm{s}}$ can be constructed. Therefore, the scheme is recursively feasible.

The trajectory tracking proof entails showing that cost (2.42a) is lower for the candidate than for the previously optimal solution such that convergence to the reference trajectory can be concluded. In the case of TMPC, we could show that cost ④ is lower than terminal cost ③ such that the candidate cost decreases at least by cost ① since cost ② is the same. However, cost ② is not the same. In fact, cost ② and cost ④ both increase depending on the worst-case disturbance impact during interval ①. Therefore, strict convergence to the reference trajectory cannot be concluded. Instead, we can prove that the average tracking error gets small.

ASSUMPTION ON EXISTENCE TERMINAL INGREDIENTS

Similar to Section 2.3.2, we need an assumption on the terminal ingredients such that the following terminal cost property is satisfied:

Assumption 4 (Terminal ingredients)**.** *There exist a terminal control law* $\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r}) : \mathcal{X} \times \tilde{\mathcal{Z}} \to \mathbb{R}^{n^{\mathrm{u}}}$ *and terminal cost* $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}}) : \mathcal{X} \times \tilde{\mathcal{X}} \to \mathbb{R}$ *such that the following property*

holds for all $\boldsymbol{z} \in \mathscr{X}$*:*

$$\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}) \leq -\int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{\tau|t}, \boldsymbol{v}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}) d\tau, \tag{2.54}$$

with sampling time $T^{\mathrm{s}} > 0$*, optimal solution* $(\boldsymbol{z}^*_{\cdot|t}, \boldsymbol{v}^*_{\cdot|t})$ *at* t *for all* $t \geq 0$ *and its extended version* $\boldsymbol{z}^*_{T+\tau|t}$ *using terminal control law* $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$ *for* $\tau \in [0, T^{\mathrm{s}}]$*, as defined in Section 2.4.3.*

Next to this assumption on terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$ and terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})$ we want to design terminal set $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ such that tightened system and obstacle avoidance constraints (2.42f) and (2.42g) are satisfied in $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$, respectively, and such that terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$ ensures robust positive invariance of $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$. The specific formulation of a tight version of $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ is presented in Section 2.4.3 as part of the proof and therefore not included in Assumption 4.

CANDIDATE SOLUTION

Similar to Section 2.3.2, the goal of defining the candidate solution is to show that there exists a feasible, not necessarily optimal, solution to (2.42) at time $t + T^{\mathrm{s}}$ given that (2.42) is successfully solved at time t. Again, this candidate solution is used to prove recursive feasibility and trajectory tracking and is not necessarily implemented in practice.

In the robust case, we pick a candidate solution similar to the nominal case. For convenience, we define for $\tau \in [T, T + T^{\mathrm{s}}]$:

$$\boldsymbol{v}^*_{\tau|t} = \kappa^{\mathrm{f}}(\boldsymbol{z}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}), \tag{2.55}$$

with $\dot{\boldsymbol{z}}^*_{\tau|t}$ given by (2.1) by applying (2.55).

Given these definitions, we can write the candidate for prediction interval $\tau \in [0, T]$ at time $t + T^{\mathrm{s}}$ as

$$\tilde{\boldsymbol{z}}_{0|t+T^{\mathrm{s}}} = \boldsymbol{x}_{t+T^{\mathrm{s}}}, \tag{2.56a}$$

$$\tilde{s}_0 = 0, \tag{2.56b}$$

$$\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} = \kappa(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}), \tag{2.56c}$$

with $\dot{\tilde{\boldsymbol{z}}}_{\tau|t+T^{\mathrm{s}}}$ and $\dot{\tilde{s}}_\tau$ according to (2.42d) and (2.42e), respectively.

Note that this candidate deviates from the one in Section 2.3.2. The main difference is that we cannot take the previously optimal state-input trajectory since the state at $t + T^{\mathrm{s}}$ is disturbed compared to prediction $\tau = T^{\mathrm{s}}$ at t. Therefore, we need to measure the state at $t + T^{\mathrm{s}}$ and use this as the initial state of the candidate solution. Consequently, following the previously optimal input trajectory $\boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \tau \in [0, T]$ from a different state results in a different trajectory. Thus, this solution cannot be directly used. Instead, we add the feedback law $\kappa^{\delta}(\boldsymbol{x}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}), \tau \in [0, T]$ described in Assumption 3.

RECURSIVE FEASIBILITY PROOF

The recursive feasibility proof follows similar arguments as the proof in Section 2.3.2 with some additional technicalities to account for the disturbances. As mentioned, the first difference is in the satisfaction of (2.42b). Whereas in Section 2.3.2, we could just

take the previously optimal solution, in this case, the state at $t+T^{\mathrm{s}}$ does not equal the nominal predicted one at predicted time T^{s} at time t because of the disturbance effect $\boldsymbol{w}_\tau \neq \mathbf{0}, \tau \in [t, t+T^{\mathrm{s}}]$. Therefore, we satisfy this constraint by measuring the disturbed state at $t+T^{\mathrm{s}}$ as given by (2.56a) of the candidate.

The initial state constraint, initial tube constraint, system dynamics constraint, and tube dynamics constraint are all trivially satisfied by (2.56). That leaves us to show that the (tightened) system constraints (2.42f), the (tightened) obstacle avoidance constraints (2.42g), and the terminal set constraint (2.42h) are satisfied in the interval $\tau \in [0, T]$.

Similar to the procedure in Section 2.3.2, we split the interval into two subintervals, $\tau \in [0, T-T^{\mathrm{s}})$ and $\tau \in [T-T^{\mathrm{s}}, T]$, to show satisfaction of (2.42f) and (2.42g).

System Constraints Satisfaction for $\tau \in [0, T-T^{\mathrm{s}})$

Let us start with the proofs for the interval $\tau \in [0, T-T^{\mathrm{s}})$. For this interval, (2.42f) are satisfied by candidate (2.56) if the following holds for $\tau \in [0, T-T^{\mathrm{s}})$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$g_j^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}) + c_j^{\mathrm{s}} s_\tau \leq 0. \tag{2.57}$$

We know that for $\tau \in [0, T-T^{\mathrm{s}})$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ the previously optimal solution satisfies

$$g_j^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}) + c_j^{\mathrm{s}} s_{T^{\mathrm{s}}+\tau} \leq 0. \tag{2.58}$$

Therefore, we would like to create an upper bound on $g_j^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}})$ in terms of $g_j^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t})$. This can be done using the Lipschitz assumption (2.44b) under control law $\kappa(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t})$. This gives the following upper bound for $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$g_j^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}) \leq g_j^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}) + c_j^{\mathrm{s}} \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t})}. \tag{2.59}$$

Filling in (2.59) in the left-hand side of (2.57) gives for $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$g_j^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}) + c_j^{\mathrm{s}} s_\tau \leq g_j^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}) + c_j^{\mathrm{s}} \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t})} + c_j^{\mathrm{s}} s_\tau. \tag{2.60}$$

Thus, a sufficient condition to satisfy (2.57) for $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ is the following:

$$g_j^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}) + c_j^{\mathrm{s}} \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t})} + c_j^{\mathrm{s}} s_\tau \leq 0. \tag{2.61}$$

We know that (2.58) is satisfied by solving (2.42) at t. Therefore, to show that (2.61) holds, we need to show that the following inequality holds for $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$c_j^{\mathrm{s}} \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t})} + c_j^{\mathrm{s}} s_\tau \leq c_j^{\mathrm{s}} s_{T^{\mathrm{s}}+\tau}, \tag{2.62}$$

Dividing the left-hand and right-hand sides by $c_j^{\mathrm{s}} > 0$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ gives

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t})} + s_\tau \leq s_{T^{\mathrm{s}}+\tau}. \tag{2.63}$$

We know that $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$ is upper-bounded by tube size s according to (2.50) and that the evolution of s over time is given by (2.52). Thus, after T^{s}, the tube has the following size:

$$s_{T^{\mathrm{s}}} = (1 - e^{\rho T^{\mathrm{s}}}) \frac{\bar{w}}{\rho}, \tag{2.64}$$

and we can write the following upper bound on $\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{0|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}|t})}$ at $t+T^{\mathrm{s}}$:

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{0|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}|t})} \leq s_{T^{\mathrm{s}}}. \tag{2.65}$$

By applying control law $\kappa(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t})$, to the system, we can show that the candidate response $\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}$ will contract to the previously optimal nominal prediction $\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}$ with factor ρ over time according to (2.47). Thus, by iteratively applying the contractivity, we can write the following upper bound for $\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})}$:

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})} \leq e^{-\rho\tau} s_{T^{\mathrm{s}}}. \tag{2.66}$$

Filling in (2.66) in (2.63) yields the following sufficient condition:

$$e^{-\rho\tau} s_{T^{\mathrm{s}}} + s_{\tau} \leq s_{T^{\mathrm{s}}+\tau}, \tag{2.67}$$

or, alternatively,

$$s_{\tau} \leq s_{T^{\mathrm{s}}+\tau} - e^{-\rho\tau} s_{T^{\mathrm{s}}}. \tag{2.68}$$

Appendix A.2 shows that (2.68) holds for $\tau \geq 0$, which includes $\tau \in [0, T]$. Therefore, the system constraints are satisfied for $\tau \in [0, T-T^{\mathrm{s}}]$ and thus for $\tau \in [0, T-T^{\mathrm{s}})$. Note that we cannot prove satisfaction for $\tau \in [0, T]$ since this reasoning relies on the fact that (2.58) holds, which is only enforced for $\tau \in [0, T]$ at t, and thus for $\tau \in [0, T-T^{\mathrm{s}}]$ at $t+T^{\mathrm{s}}$.

Obstacle Avoidance Constraints Satisfaction for $\tau \in [0, T-T^{\mathrm{s}})$

For interval $\tau \in [0, T-T^{\mathrm{s}})$, we know that (2.42g) are satisfied by candidate (2.56) if the following holds for $\tau \in [0, T-T^{\mathrm{s}})$, $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$:

$$g^{\mathrm{o}}_{j}(M\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}) + c^{\mathrm{o}} s_{\tau} \leq 0. \tag{2.69}$$

We know that for $\tau \in [0, T-T^{\mathrm{s}})$, $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$ the previously optimal solution satisfies

$$g^{\mathrm{o}}_{j}(M\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}) + c^{\mathrm{o}} s_{T^{\mathrm{s}}+\tau} \leq 0. \tag{2.70}$$

Therefore, similar to the system constraints proof for $\tau \in [0, T-T^{\mathrm{s}})$, we would like to create an upper bound on $g^{\mathrm{o}}_{j}(M\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}})$ in terms of $g^{\mathrm{o}}_{j}(M\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})$. Lipschitz assumption (2.44c) gives the following upper bound for $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$:

$$g^{\mathrm{o}}_{j}(M\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}) \leq g^{\mathrm{o}}_{j}(M\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}) + c^{\mathrm{o}} \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})}. \tag{2.71}$$

By following the same steps as for the system constraints, we can show that the proof boils down to showing that (2.68) holds, which is true by the combination of the proof in Appendix A.2 and Assumption 2 from Section 2.3.2. Thus, the obstacle avoidance constraints are also satisfied for $\tau \in [0, T-T^{\mathrm{s}})$.

Terminal Set Constraint Satisfaction for $\tau \in [T-T^{\mathrm{s}}, T]$

Next, we prove that if (2.42h) is satisfied at time t, candidate (2.56) satisfies (2.42h) in prediction interval $\tau \in [T-T^{\mathrm{s}}, T]$ at time $t+T^{\mathrm{s}}$. Before proving this, we need to know a suitable expression for the terminal set. In the TMPC case, $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ was chosen as a sublevel

set of $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})$. If we use this set in the robust case, it means that candidate (2.56) should satisfy this constraint for $\tau \in [T - T^{\mathrm{s}}, T]$ at $t + T^{\mathrm{s}}$. As discussed, the minimum terminal set scaling α could be chosen equal to 0 in the case of TMPC. However, even if the previously optimal solution $\boldsymbol{z}^*_{[0,T]|t}$ would exactly overlap with reference $\boldsymbol{x}^{\mathrm{r}}_{t+[0,T]}$, the disturbance impact during interval $[t, t + T^{\mathrm{s}}]$ renders terminal set constraint (2.42h) infeasible for candidate (2.56) since we only have a finite horizon and exponential contraction of the disturbance impact as given by (2.66). In other words, candidate $\tilde{\boldsymbol{z}}_{[T-T^{\mathrm{s}},T]|t+T^{\mathrm{s}}}$ deviates from previously optimal solution $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t}$. Therefore, we need another expression for the minimum terminal set that accounts for the disturbance impact during $[t, t + T^{\mathrm{s}}]$.

Besides the mentioned disturbance impact, we also need to account for the fact that it is likely to happen that the previously optimal solution does not overlap with the reference for $\tau \in [T, T + T^{\mathrm{s}}]$, i.e., $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t} \neq \boldsymbol{x}^{\mathrm{r}}_{t+[T,T+T^{\mathrm{s}}]}$, on a disturbed system, even if such a system has exponential disturbance rejection properties by closed-loop terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$.

In summary, to show that candidate (2.56) satisfies terminal set constraint (2.42h) for $\tau \in [T - T^{\mathrm{s}}, T]$ at time $t + T^{\mathrm{s}}$, we need a terminal set that accounts for both the error between previously optimal solution $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t}$ and reference $\boldsymbol{x}^{\mathrm{r}}_{t+[T,T+T^{\mathrm{s}}]}$ and the error between candidate $\tilde{\boldsymbol{z}}_{[T-T^{\mathrm{s}},T]|t+T^{\mathrm{s}}}$ and previously optimal solution $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t}$.

The error between $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t}$ and $\boldsymbol{x}^{\mathrm{r}}_{t+[T,T+T^{\mathrm{s}}]}$ contracts with rate ρ according to (2.47). For $\tau \in [0, T^{\mathrm{s}}]$ it is given by

$$\sqrt{V^{\delta}(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq e^{-\rho\tau} \sqrt{V^{\delta}(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{T|t})}. \tag{2.72}$$

The error between $\tilde{\boldsymbol{z}}_{[T-T^{\mathrm{s}},T]|t+T^{\mathrm{s}}}$ and $\boldsymbol{z}^*_{[T,T+T^{\mathrm{s}}]|t}$ is caused by the described disturbance impact during $[t, t + T^{\mathrm{s}}]$ and also contracts with rate ρ according to the following expression for $\tau \in [0, T^{\mathrm{s}}]$:

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T+\tau|t})} \overset{(2.66)}{\leq} e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}}. \tag{2.73}$$

We can construct an upper bound on the total tracking error of the candidate using (2.45) in Assumption 3. For $\tau \in [0, T^{\mathrm{s}}]$ this yields

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq \sqrt{V^{\delta}(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} + \sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T+\tau|t})}. \tag{2.74}$$

Therefore, we can write the upper bound on the total terminal tracking error for $\tau \in [0, T^{\mathrm{s}}]$ as

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq e^{-\rho\tau} \sqrt{V^{\delta}(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T})} + e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}}. \tag{2.75}$$

To prove that the candidate satisfies the terminal set constraint given this inequality, we leverage the expression of the terminal set proposed in [81]:

$$\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}) := \left\{ \boldsymbol{z} \in \mathscr{X}, s_T \in \mathbb{R} \;\middle|\; \sqrt{V^{\delta}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})} + s_T \leq \alpha \right\}, \tag{2.76}$$

with s_T representing the tube size evaluated at T according to (2.52) and α denoting the terminal set scaling, similar to the definition of α in (2.14). Note that the terminal set is expressed in non-squared form, so we use notation α instead of α^2.

Extending this terminal set to prediction interval $\tau \in [0, T^{\mathrm{s}}]$ gives

$$\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}_{t+T+\tau}) := \left\{ \boldsymbol{z}_{T+\tau|t} \in \mathscr{X}, s_{T+\tau} \in \mathbb{R} \;\middle|\; \sqrt{V^{\delta}(\boldsymbol{z}_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} + s_{T+\tau} \leq \alpha \right\}. \tag{2.77}$$

We know that pair $(\boldsymbol{z}^{*}_{T|t}, s_T)$ satisfies terminal set constraint (2.76). Therefore, it holds that

$$\sqrt{V^{\delta}(\boldsymbol{z}^{*}_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T})} \leq \alpha - s_T, \tag{2.78}$$

which leads to the following expression for (2.75) for $\tau \in [0, T^{\mathrm{s}}]$:

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq e^{-\rho\tau}(\alpha - s_T) + e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}}. \tag{2.79}$$

We can use this expression to derive a minimum value for terminal set scaling α such that $\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}$ satisfies the terminal set constraint for $\tau \in [0, T^{\mathrm{s}}]$, i.e.,

$$\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}+\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq \alpha - s_{T-T^{\mathrm{s}}+\tau}. \tag{2.80}$$

This inequality is satisfied if the addition of the two errors described above, i.e., the right-hand side of (2.79), is upper-bounded by the maximum error allowed in the terminal set, i.e., the right-hand side of (2.80):

$$e^{-\rho\tau}(\alpha - s_T) + e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}} \leq \alpha - s_{T-T^{\mathrm{s}}+\tau}, \tag{2.81}$$

which trivially holds for $\tau = 0$ by Appendix A.2. Working out this inequality for $\tau \in (0, T^{\mathrm{s}}]$ gives the following lower bound on α:

$$e^{-\rho\tau}(\alpha - s_T) + e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}} \overset{\text{(a)}}{\leq} \alpha - s_{T-T^{\mathrm{s}}+\tau} \tag{2.82a}$$

$$(1 - e^{-\rho\tau})\alpha + e^{-\rho\tau} s_T - s_{T-T^{\mathrm{s}}+\tau} \overset{\text{(b)}}{\geq} e^{-\rho(T-T^{\mathrm{s}}+\tau)} s_{T^{\mathrm{s}}} \tag{2.82b}$$

$$(1 - e^{-\rho\tau})\alpha + e^{-\rho\tau}(1 - e^{-\rho T})\frac{\bar{w}}{\rho} - (1 - e^{-\rho(T-T^{\mathrm{s}}+\tau)})\frac{\bar{w}}{\rho} \overset{\text{(c)}}{\geq} e^{-\rho(T-T^{\mathrm{s}}+\tau)}(1 - e^{-\rho T^{\mathrm{s}}})\frac{\bar{w}}{\rho} \tag{2.82c}$$

$$(1 - e^{-\rho\tau})\alpha + (e^{-\rho\tau} - e^{-\rho(T+\tau)} - 1 + e^{-\rho(T-T^{\mathrm{s}}+\tau)})\frac{\bar{w}}{\rho} \overset{\text{(d)}}{\geq} (e^{-\rho(T-T^{\mathrm{s}}+\tau)} - e^{-\rho(T+\tau)})\frac{\bar{w}}{\rho} \tag{2.82d}$$

$$(1 - e^{-\rho\tau})\alpha + (e^{-\rho\tau} - 1)\frac{\bar{w}}{\rho} \overset{\text{(e)}}{\geq} 0 \tag{2.82e}$$

$$(1 - e^{-\rho\tau})\alpha - (1 - e^{-\rho\tau})\frac{\bar{w}}{\rho} \overset{\text{(f)}}{\geq} 0 \tag{2.82f}$$

$$(1 - e^{-\rho\tau})\alpha \overset{\text{(g)}}{\geq} (1 - e^{-\rho\tau})\frac{\bar{w}}{\rho} \tag{2.82g}$$

$$\alpha \overset{\text{(h)}}{\geq} \frac{\bar{w}}{\rho}, \tag{2.82h}$$

where (a) is obtained by setting the upper bound of (2.79) smaller equal to (2.80), (b) by re-ordering terms and combining α multiplication factors, (c) by writing out the tube

dynamics according to (2.52), (d) by combining $\frac{\bar{w}}{\rho}$ multiplication factors, (e) by canceling common terms on both sides and (f)-(h) by trivially working out the inequality.

In other words, α should be at least as large as the maximum tube size given in (2.53). This makes sense since this means that terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{x}, \boldsymbol{r})$ can keep the system within that tube around the reference trajectory, guaranteeing that applying this solution will keep the system safe at all times. Note that α can also be chosen larger than $\frac{\bar{w}}{\rho}$, similar to the discussion in Section 2.3.2. This will result in a larger terminal set, which will render (2.42) easier to solve, but it will also increase conservatism in planning the reference trajectory.

Remark 10. *An alternative to the proof above is to show terminal set constraint satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$ at $t + T^{\mathrm{s}}$ by first proving that candidate $\tilde{\boldsymbol{z}}_{T-T^{\mathrm{s}}|t+T^{\mathrm{s}}}$ and corresponding tube $s_{T-T^{\mathrm{s}}}$ satisfy terminal set constraint* (2.76) *and then leveraging the fact that the terminal set is invariant for $\alpha \geq \frac{\bar{w}}{\rho}$, as shown in Appendix A.3.*

System Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$

As mentioned before, given that tube inequality (2.68) is proven to hold for $\tau \in [0, T]$, the only thing we need to ensure satisfaction of candidate system constraints (2.57) for $\tau \in [T - T^{\mathrm{s}}, T]$ is to show that system constraints of previously optimal solution (2.58) are satisfied for $\tau \in [T - T^{\mathrm{s}}, T]$ at $t + T^{\mathrm{s}}$, or, equivalently, for $\tau \in [T, T + T^{\mathrm{s}}]$ at t, given that terminal set (2.76) is invariant.

Appendix A.3 proves that terminal set (2.76) is invariant for $(\boldsymbol{z}^*_{[T,T+\tau]|t}, s_{T+\tau}), \tau \geq 0$ under the same condition $\alpha \geq \frac{\bar{w}}{\rho}$ as given in (2.82). Given the proof of (2.68) from before, candidate system constraints (2.57) are satisfied for $\tau \in [T - T^{\mathrm{s}}, T]$ under the following assumption:

Assumption 5 (System constraints tightening for reference trajectory)**.** *The reference trajectory satisfies the tightened systems constraints, i.e.,* (2.7) *holds, with $g_j^{\mathrm{r,s}}(\boldsymbol{x}, \boldsymbol{u})$ given by*

$$g_j^{\mathrm{r,s}}(\boldsymbol{x}, \boldsymbol{u}) = g_j^{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{u}) + c_j^{\mathrm{s}} \alpha, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.83}$$

where c_j^{s} is defined in (2.44b) *for $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and $\alpha \geq \frac{\bar{w}}{\rho}$.*

This assumption is similar to the design of (2.33) in Section 2.3.2, but with the required lower bound $\alpha \geq \frac{\bar{w}}{\rho}$ to account for the effect of disturbances.

Obstacle Avoidance Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$

Proving that candidate obstacle avoidance constraints (2.69) are satisfied for $\tau \in [T - T^{\mathrm{s}}, T]$ follows the same arguments as the proof for the system constraints for $\tau \in [T - T^{\mathrm{s}}, T]$ under the combination of Assumption 2 and a similar assumption as above:

Assumption 6 (Obstacle avoidance constraints tightening for reference trajectory)**.** *The reference trajectory satisfies the tightened obstacle avoidance constraints, i.e.,* (2.9) *holds, with $g_j^{\mathrm{r,o}}(\boldsymbol{p})$ given by*

$$g_j^{\mathrm{r,o}}(\boldsymbol{p}) = g_j^{\mathrm{o}}(\boldsymbol{p}) + c^{\mathrm{o}} \alpha, \ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}, \tag{2.84}$$

where c^{o} is defined in (2.44c) *and $\alpha \geq \frac{\bar{w}}{\rho}$.*

This assumption is similar to the design of (2.35) in Section 2.3.2 but with the required lower bound $\alpha \geq \frac{\bar{w}}{\rho}$ to account for the effect of disturbances. □

TRAJECTORY TRACKING PROOF

The trajectory tracking proof in this section follows similar arguments as the one presented in Section 2.3.2. For convenience, let us define

$$\mathscr{J}^{\mathrm{s}}_{\tau|t} \coloneqq \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{\tau|t}, \boldsymbol{v}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}), \tag{2.85}$$

$$\mathscr{J}^{\mathrm{f}}_{T|t} \coloneqq \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}), \tag{2.86}$$

$$\mathscr{J}^{*}_{t} \coloneqq \mathscr{J}^{*}_{t}(\boldsymbol{x}_t, \boldsymbol{r}_t), \tag{2.87}$$

$$\mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} \coloneqq \mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}), \tag{2.88}$$

$$\mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} \coloneqq \mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}), \tag{2.89}$$

$$\mathscr{J}^{*}_{t+T^{\mathrm{s}}} \coloneqq \mathscr{J}^{*}_{t+T^{\mathrm{s}}}(\boldsymbol{x}_{t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}}), \tag{2.90}$$

and start by following the same proof as in (2.22) in Section 2.3.2:

$$\mathscr{J}^{*}_{t+T^{\mathrm{s}}} \leq \int_0^T \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}}, \tag{2.91a}$$

$$= \int_0^{T-T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau + \int_{T-T^{\mathrm{s}}}^T \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}}. \tag{2.91b}$$

So far, the proof for TMPC and RMPC is the same. However, note that the stage costs are not equal at the same points in time, i.e., $\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) \neq \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}), \tau \in [0, T]$, since the candidate $\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}$ deviates from the previously optimal solution $\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}$. Therefore, (c) in (2.22) does not hold. Specifically, for $\tau \in [0, T]$, candidate stage cost $\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau})$ is defined as

$$\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) = \|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R, \tag{2.92}$$

and previously optimal stage cost $\mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau})$ as

$$\mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) = \|\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R. \tag{2.93}$$

Writing out the difference between the two gives

$$\begin{aligned}
&\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) - \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) \\
&= \|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R - \|\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R \\
&= \|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q - \|\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R - \|\boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R.
\end{aligned} \tag{2.94}$$

Note that this expression is continuously differentiable in compact set $\mathcal{Z} \times \tilde{\mathcal{Z}}$. Therefore, it is Lipschitz continuous in this set, and the following inequality holds for $\tau \in [0, T]$:

$$\begin{aligned}
&\|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q - \|\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_Q + \|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R - \|\boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t} - \boldsymbol{u}^{\mathrm{r}}_{t+T^{\mathrm{s}}+\tau}\|^2_R \\
&\leq L^{\mathscr{J}^{\mathrm{s}},\mathrm{x}} \|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}\| + L^{\mathscr{J}^{\mathrm{s}},\mathrm{u}} \|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}\|.
\end{aligned} \tag{2.95}$$

This result shows that we can find an upper bound on the difference in stage costs.

Ideally, we would like to find the smallest possible value for this upper bound. We can rewrite (2.44a) in Assumption 3 as

$$\|\boldsymbol{x}-\boldsymbol{z}\| \leq \frac{1}{\sqrt{c^{\delta,\mathrm{l}}}}\sqrt{V^{\delta}(\boldsymbol{x},\boldsymbol{z})}. \tag{2.96}$$

Inserting this upper bound in (2.95) gives for $\tau \in [0, T]$:

$$L^{\mathscr{J}^{\mathrm{s}},\mathrm{x}}\|\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}\| \leq \frac{L^{\mathscr{J}^{\mathrm{s}},\mathrm{x}}}{\sqrt{c^{\delta,\mathrm{l}}}}\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})}. \tag{2.97}$$

In other words, the difference between candidate and previously optimal states is upper-bounded by some other Lipschitz constant $\frac{L^{\mathscr{J}^{\mathrm{s}},\mathrm{x}}}{\sqrt{c^{\delta,\mathrm{l}}}}$ multiplied with the incremental Lyapunov function evaluated at the corresponding point.

To refine the expression for $L^{\mathscr{J}^{\mathrm{s}},\mathrm{u}}\|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}\|$, note that (2.44b) implies that control law $\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})$ is Lipschitz continuous given that the inputs are constrained by a box, as defined in (2.2). Thus, the following holds for $\tau \in [0, T]$:

$$\begin{aligned}\|\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}\| &\overset{\text{(a)}}{=} \|\kappa(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}) - \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}\| \\ &\overset{\text{(b)}}{=} \|\boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t} + \kappa^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}) - \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}\| \\ &\overset{\text{(c)}}{\leq} \bar{\kappa}^{\delta}\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})},\end{aligned} \tag{2.98}$$

where (a) is obtained by filling in (2.56c), (b) by filling in control law definition (2.43), and (c) by the resulting Lipschitz bound on the feedback law in $\kappa^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})$ with constant $\bar{\kappa}^{\delta}$.

Combining (2.94) and (2.95) and filling in upper bounds (2.97) and (2.98) gives the following stage cost upper bound for $\tau \in [0, T]$:

$$\begin{aligned}&\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) - \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) \\ &\leq \left(\frac{L^{\mathscr{J}^{\mathrm{s}},\mathrm{x}}}{\sqrt{c^{\delta,\mathrm{l}}}} + L^{\mathscr{J}^{\mathrm{s}},\mathrm{u}}\bar{\kappa}^{\delta}\right)\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})} \\ &=: L^{\mathscr{J}^{\mathrm{s}},\mathrm{xu}}\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})}.\end{aligned} \tag{2.99}$$

Using the upper bound for $\sqrt{V^{\delta}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t})}$ in (2.66), we can define the following upper bound on the stage costs for $\tau \in [0, T]$:

$$\begin{aligned}&\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) - \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) \\ &\leq L^{\mathscr{J}^{\mathrm{s}},\mathrm{xu}}e^{-\rho\tau}(1-e^{\rho T^{\mathrm{s}}})\frac{\bar{w}}{\rho} =: \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}),\end{aligned} \tag{2.100}$$

or, alternatively,

$$\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) \leq \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^{*}_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) + \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}). \tag{2.101}$$

In other words, this means that the candidate stage cost is maximum $\alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}})$ away from the previously optimal stage cost at prediction time $\tau \in [0, T]$.

Next to the stage costs, the terminal costs of the candidate and extended previously optimal solution (2.56) are not equal, i.e., $\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) \neq \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T})$. Since the terminal cost is also continuously differentiable in compact set $\mathscr{X} \times \tilde{\mathscr{X}}$, it is Lipschitz continuous. Therefore, we can write a similar inequality as for the difference in stage costs (2.95):

$$\begin{aligned}
&\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) \\
&= \|\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}\|^2_P - \|\boldsymbol{z}^{*}_{T^{\mathrm{s}}+T|t} - \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}\|^2_P \\
&\leq L^{\mathscr{J}^{\mathrm{f}},\mathrm{x}} \|\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}} - \boldsymbol{z}^{*}_{T^{\mathrm{s}}+T|t}\|.
\end{aligned} \tag{2.102}$$

Following the same steps above, we can write the upper bound on the difference in terminal costs as

$$\begin{aligned}
&\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^{*}_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) \\
&\leq L^{\mathscr{J}^{\mathrm{f}}} e^{-\rho T}(1 - e^{\rho T^{\mathrm{s}}}) \frac{\bar{w}}{\rho} =: \alpha^{\mathrm{f}}(\rho, \bar{w}, T^{\mathrm{s}}).
\end{aligned} \tag{2.103}$$

We can now further work out the trajectory tracking proof in (2.91) by replacing the equality for (c) in (2.22) with the inequality in (2.101):

$$\mathscr{J}^{*}_{t+T^{\mathrm{s}}} \overset{\text{(a)}}{\leq} \int_0^{T-T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau + \int_{T-T^{\mathrm{s}}}^{T} \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} \tag{2.104a}$$

$$\overset{\text{(b)}}{\leq} \int_{T^{\mathrm{s}}}^{T} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} \tag{2.104b}$$

$$\overset{\text{(c)}}{=} \int_0^{T} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau - \int_0^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} \tag{2.104c}$$

$$\overset{\text{(d)}}{=} \mathscr{J}^{*}_{t} - \mathscr{J}^{\mathrm{f}}_{T|t} - \int_0^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} \tag{2.104d}$$

$$\overset{\text{(e)}}{=} \mathscr{J}^{*}_{t} - \int_0^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \mathscr{J}^{\mathrm{f}}_{T|t+T^{\mathrm{s}}} - \mathscr{J}^{\mathrm{f}}_{T|t} + \int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau \tag{2.104e}$$

$$\overset{\text{(f)}}{\leq} \mathscr{J}^{*}_{t} - \int_0^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau$$

$$+ \mathscr{J}^{\mathrm{f}}_{T^{\mathrm{s}}+T|t} + \alpha^{\mathrm{f}}(\rho, \bar{w}, T^{\mathrm{s}}) - \mathscr{J}^{\mathrm{f}}_{T|t} + \int_{T}^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau \tag{2.104f}$$

$$\overset{(g)}{\leq} \mathscr{J}^{*}_{t} - \int_{0}^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \alpha^{\mathrm{f}}(\rho, \bar{w}, T^{\mathrm{s}}), \tag{2.104g}$$

where (a) is obtained by (2.91), (b) by filling in (2.101) in $\int_{0}^{T-T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau$ and $\int_{T-T^{\mathrm{s}}}^{T} \mathscr{J}^{\mathrm{s}}_{\tau|t+T^{\mathrm{s}}} d\tau$, (c), (d), and (e) by following the same steps as (d), (e), and (f) in (2.22), (f) by (2.103), and (g) by (2.54) in Assumption 4. Leveraging (2.54) in Assumption 4 gives

$$\mathscr{J}^{*}_{t+T^{\mathrm{s}}} - \mathscr{J}^{*}_{t} \leq - \int_{0}^{T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}_{\tau|t} d\tau + \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \alpha^{\mathrm{f}}(\rho, \bar{w}, T^{\mathrm{s}}). \tag{2.105}$$

Thus, we get the same descent bound for $\mathscr{J}^{*}_{t}$ as in Section 2.3.2, up to

$$\bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}, T^{\mathrm{s}}) := \int_{T^{\mathrm{s}}}^{T} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \int_{T}^{T+T^{\mathrm{s}}} \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}, T^{\mathrm{s}}) d\tau + \alpha^{\mathrm{f}}(\rho, \bar{w}, T^{\mathrm{s}}). \tag{2.106}$$

Similar to Section 2.3.2, we prove convergence by first noting that the descent property of the optimal cost (2.105) implies

$$\mathscr{J}^{*}_{t+T^{\mathrm{s}}} - \mathscr{J}^{*}_{t} \leq -c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} d\tau + \bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}, T^{\mathrm{s}}), \tag{2.107}$$

with constant $c^{\mathscr{J},\mathrm{d}} > 0$. Re-arranging the terms gives

$$c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} d\tau \leq \mathscr{J}^{*}_{t} - \mathscr{J}^{*}_{t+T^{\mathrm{s}}} + \bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}, T^{\mathrm{s}}). \tag{2.108}$$

Instead of $\| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q}$ we are interested in the actual tracking error $\| \boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q}$, which is upper-bounded by

$$\| \boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} \overset{(a)}{\leq} c^{\mathrm{t}} \| \boldsymbol{x}_{\tau} - \boldsymbol{z}^{*}_{\tau-t|t} \| + \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} \tag{2.109a}$$

$$\overset{(b)}{\leq} c^{\mathrm{t},1} s_{\tau} + \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} \tag{2.109b}$$

where (a) is obtained by a Lipschitz bound with constant c^{t} and (b) by another Lipschitz bound with constant $c^{\mathrm{t},1}$ on the norm using a combination of (2.44a), (2.50), and (2.52). Therefore, we can write a similar upper bound as in (2.108) but now for $\| \boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q}$:

$$c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} \| \boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} d\tau \leq c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} + c^{\mathrm{t},1} s_{\tau} d\tau \tag{2.110a}$$

$$\overset{(a)}{=} c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} \| \boldsymbol{z}^{*}_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau} \|^{2}_{Q} d\tau + c^{\mathscr{J},\mathrm{d}} \int_{t}^{t+T^{\mathrm{s}}} c^{\mathrm{t},1} s_{\tau} d\tau \tag{2.110b}$$

$$\overset{(b)}{\leq} \mathscr{J}^{*}_{t} - \mathscr{J}^{*}_{t+T^{\mathrm{s}}} + \bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}), \tag{2.110c}$$

with $\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}) := \bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}, T^{\mathrm{s}}) + c^{\mathscr{J},\mathrm{d}} c^{\mathrm{t,1}}\big(T^{\mathrm{s}} - \frac{1}{\rho}(1 - e^{-\rho T^{\mathrm{s}}})\big)\frac{\bar{w}}{\rho}$, and where (a) is obtained by writing out the integral expression and (b) by (2.108). Iterating this inequality from 0 to $t = 2T^{\mathrm{s}}$ yields

$$c^{\mathscr{J},\mathrm{d}} \int_0^{2T^{\mathrm{s}}} \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^{\mathrm{r}}\|_Q^2 d\tau \leq \mathscr{J}_0^* - \mathscr{J}_{T^{\mathrm{s}}}^* + \bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}) + \mathscr{J}_{T^{\mathrm{s}}}^* - \mathscr{J}_{2T^{\mathrm{s}}}^* + \bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}) \tag{2.111a}$$

$$\leq \mathscr{J}_0^* - \mathscr{J}_{2T^{\mathrm{s}}}^* + 2\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}) \tag{2.111b}$$

$$= \mathscr{J}_0^* - \mathscr{J}_{2T^{\mathrm{s}}}^* + \frac{t}{T^{\mathrm{s}}}\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}), \tag{2.111c}$$

In contrast to (2.25) in Section 2.3.2, iterating this inequality for $t \to \infty$ is not necessarily bounded for the worst-case disturbance impact. Therefore, we can conclude that the average of this inequality gets small by dividing it by t and leveraging the fact that $\mathscr{J}_t^*$ is uniformly bounded:

$$\lim_{t\to\infty} \frac{\int_0^t \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^{\mathrm{r}}\|_Q^2 d\tau}{t} \leq \frac{\mathscr{J}_0^* - \lim_{t\to\infty} \mathscr{J}_t^*}{c^{\mathscr{J},\mathrm{d}} t} + \frac{1}{c^{\mathscr{J},\mathrm{d}} T^{\mathrm{s}}}\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}}). \tag{2.112}$$

Thus, by applying Barbalat's lemma [71] on $\frac{\int_0^t \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^{\mathrm{r}}\|_Q^2 d\tau}{t}$ we conclude that, on average, the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ gets small over time, proportional to $\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}, T^{\mathrm{s}})$. Note that this conclusion makes sense in the presence of unpredictable and bounded disturbances. □

DESIGN TO SATISFY THE ASSUMPTIONS

This section will detail the design of:

- the obstacle avoidance constraints, which need to satisfy Assumption 2;
- incremental Lyapunov function $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$ and corresponding Lipschitz constants $c_j^{\mathrm{s}}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and c^{o}, which need to satisfy Assumption 3;
- the terminal ingredients, which need to satisfy Assumption 4;
- the reference trajectory, which needs to satisfy Assumptions 5 and 6.

Obstacle Avoidance Constraints Design

To prove recursive feasibility, we need the obstacle avoidance constraint to satisfy Assumption 2. We can achieve this by following the same design as written in Section 2.3.2.

Incremental Stabilizability Design

The incremental stabilizability design follows a similar procedure to the terminal ingredients design in Section 2.3.2. The goal is to compute a suitable incremental Lyapunov function $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$ that is bounded by tube size s given in (2.42e) and corresponding control law $\kappa(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v})$ that satisfy properties (2.44a)-(2.45).

Remark 11. *The most important difference with the design in Section 2.3.2 is the fact that $V^\delta(\boldsymbol{x},\boldsymbol{z})$ should satisfy contraction property (2.44d) that is used to prove trajectory tracking and recursive feasibility in the presence of disturbances. Furthermore, control law $\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})$ is used in the proofs and implemented in closed-loop execution according to (2.43).*

We can write the dynamics of $\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}$ as follows:

$$\frac{d}{dt}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})} = \frac{d}{dt}\big(V^\delta(\boldsymbol{x},\boldsymbol{z})\big)^{\frac{1}{2}} = \frac{1}{2}V^\delta(\boldsymbol{x},\boldsymbol{z})^{-\frac{1}{2}}\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z}). \tag{2.113}$$

Since it is easier to find an expression for $\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z})$ rather than $\frac{d}{dt}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}$, we rewrite (2.113) as

$$\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z}) = 2\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}\frac{d}{dt}\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}. \tag{2.114}$$

Filling in desired tube dynamics (2.42e) gives the following upper bound on $\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z})$:

$$\frac{d}{dt}V^\delta(\boldsymbol{x},\boldsymbol{z}) \leq 2\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}\big(-\rho\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}+\bar{w}\big) = -2\rho V^\delta(\boldsymbol{x},\boldsymbol{z}) + 2\sqrt{V^\delta(\boldsymbol{x},\boldsymbol{z})}\,\bar{w}. \tag{2.115}$$

Thus, the goal is to find a suitable incremental Lyapunov function of which the upper bound can be written in the form (2.115).

Using a control contraction metric (CCM) [85], we can define the incremental Lyapunov function $V^\delta(\boldsymbol{x},\boldsymbol{z})$ as

$$V^\delta(\boldsymbol{x},\boldsymbol{z}) \coloneqq \min_{\boldsymbol{c}^{\mathrm{x}}(s)\in C(\boldsymbol{x},\boldsymbol{z})} \int_0^1 \frac{\partial \boldsymbol{c}^{\mathrm{x}}(s)}{\partial s}^\top P^\delta\big(\boldsymbol{c}^{\mathrm{x}}(s)\big)\frac{\partial \boldsymbol{c}^{\mathrm{x}}(s)}{\partial s}\,ds, \tag{2.116}$$

with metric $P^\delta\big(\boldsymbol{c}^{\mathrm{x}}(s)\big)$ and where $\boldsymbol{c}^{\mathrm{x}}(s)$ is a curve in set of curves

$$C(\boldsymbol{x},\boldsymbol{z}) \coloneqq \Big\{\boldsymbol{c}^{\mathrm{x}}(s):[0,1]\to\mathbb{R}^{n^{\mathrm{x}}} \;\Big|\; \boldsymbol{c}^{\mathrm{x}}(0)=\boldsymbol{z},\boldsymbol{c}^{\mathrm{x}}(1)=\boldsymbol{x}\Big\}. \tag{2.117}$$

$\boldsymbol{\gamma}^{\mathrm{x}}(s)\in C(\boldsymbol{x},\boldsymbol{z})$ denotes the geodesic: the curve that minimizes $V^\delta(\boldsymbol{x},\boldsymbol{z})$.

We know that the evolutions of the nominal and disturbed state are described by (2.1) and (2.39), respectively. Thus, we know the evolution of the endpoints $\boldsymbol{\gamma}^{\mathrm{x}}(0)$ and $\boldsymbol{\gamma}^{\mathrm{x}}(1)$ of geodesic $\boldsymbol{\gamma}^{\mathrm{x}}(s)$ over time. A logical candidate for the evolution of $\boldsymbol{\gamma}^{\mathrm{x}}(s), s\in[0,1]$, over time, i.e., one that matches $\dot{\boldsymbol{z}}$ and $\dot{\boldsymbol{x}}$ for $s=0$ and $s=1$, respectively, is thus given by

$$\dot{\boldsymbol{\gamma}}^{\mathrm{x}}(s) \coloneqq f\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big) + E(\boldsymbol{w}^{\mathrm{b}} + s\boldsymbol{w}^0), \tag{2.118}$$

with control law $\boldsymbol{\gamma}^{\mathrm{u}}(s)$ that is obtained by integrating the feedback law over the geodesic:

$$\boldsymbol{\gamma}^{\mathrm{u}}(s) \coloneqq \boldsymbol{v} + \int_0^s K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(\tilde{s})\big)\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(\tilde{s})}{\partial \tilde{s}}\,d\tilde{s}. \tag{2.119}$$

Similarly, a candidate for the evolution of the path derivative of the geodesic over time is given by

$$\frac{d}{dt}\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s} \coloneqq A^{\mathrm{cl}}\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big)\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s} + E\boldsymbol{w}^0, \tag{2.120}$$

with closed-loop dynamics

$$A^{\mathrm{cl}}\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big) := A\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big) + B\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big)K^{\delta}\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big), \tag{2.121}$$

and Jacobians

$$A\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big) := \frac{\partial f^{\mathrm{w}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})}{\partial \boldsymbol{x}}\Big|_{(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s))}, \quad B\big(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s)\big) := \frac{\partial f^{\mathrm{w}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})}{\partial \boldsymbol{u}}\Big|_{(\boldsymbol{\gamma}^{\mathrm{x}}(s),\boldsymbol{\gamma}^{\mathrm{u}}(s))}, \tag{2.122}$$

since, locally, the nonlinear dynamics are given by its Jacobians.

Using definitions (2.118) and (2.120) for $\dot{\boldsymbol{\gamma}}^{\mathrm{x}}(s)$ and $\frac{d}{dt}\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s}$, respectively, and temporarily using the following shorthand notations:

$$\boldsymbol{\gamma}^{\mathrm{x}} := \boldsymbol{\gamma}^{\mathrm{x}}(s), \tag{2.123a}$$
$$\boldsymbol{\gamma}^{\mathrm{u}} := \boldsymbol{\gamma}^{\mathrm{u}}(s), \tag{2.123b}$$
$$A^{\mathrm{cl}} := A^{\mathrm{cl}}(\boldsymbol{\gamma}^{\mathrm{x}},\boldsymbol{\gamma}^{\mathrm{u}}), \tag{2.123c}$$
$$A := A(\boldsymbol{\gamma}^{\mathrm{x}},\boldsymbol{\gamma}^{\mathrm{u}}), \tag{2.123d}$$
$$B := B(\boldsymbol{\gamma}^{\mathrm{x}},\boldsymbol{\gamma}^{\mathrm{u}}), \tag{2.123e}$$
$$\boldsymbol{\gamma}^{\mathrm{x}}_s := \frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s}, \tag{2.123f}$$
$$\dot{\boldsymbol{\gamma}}^{\mathrm{x}}_s := \frac{d}{dt}\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s}, \tag{2.123g}$$
$$P^{\delta} := P^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}), \tag{2.123h}$$

we can write the time evolution of the upper bound of $\frac{d}{dt}V^{\delta}(\boldsymbol{x},\boldsymbol{z})$ in the form (2.115) using candidates $\boldsymbol{\gamma}^{\mathrm{x}}_s$ and $\dot{\boldsymbol{\gamma}}^{\mathrm{x}}_s$:

$$\frac{d}{dt}V^{\delta}(\boldsymbol{x},\boldsymbol{z}) \le \int_0^1 \dot{\boldsymbol{\gamma}}^{\mathrm{x}\top}_s P^{\delta}\boldsymbol{\gamma}^{\mathrm{x}}_s + \boldsymbol{\gamma}^{\mathrm{x}\top}_s \dot{P}^{\delta}\boldsymbol{\gamma}^{\mathrm{x}}_s + \boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta}\dot{\boldsymbol{\gamma}}^{\mathrm{x}}_s ds \tag{2.124a}$$
$$= \int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s (A^{\mathrm{cl}\top}P^{\delta} + \dot{P}^{\delta} + P^{\delta}A^{\mathrm{cl}})\boldsymbol{\gamma}^{\mathrm{x}}_s + (E\boldsymbol{w}^0)^{\top}P^{\delta}\boldsymbol{\gamma}^{\mathrm{x}}_s + \boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta}E\boldsymbol{w}^0 ds \tag{2.124b}$$
$$= \int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s (A^{\mathrm{cl}\top}P^{\delta} + \dot{P}^{\delta} + P^{\delta}gA^{\mathrm{cl}})\boldsymbol{\gamma}^{\mathrm{x}}_s + 2\boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta}E\boldsymbol{w}^0 ds \tag{2.124c}$$
$$= \int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s (A^{\mathrm{cl}\top}P^{\delta} + \dot{P}^{\delta} + P^{\delta}A^{\mathrm{cl}})\boldsymbol{\gamma}^{\mathrm{x}}_s ds + 2\int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta}E\boldsymbol{w}^0 ds \tag{2.124d}$$
$$= \int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s (A^{\mathrm{cl}\top}P^{\delta} + \dot{P}^{\delta} + P^{\delta}A^{\mathrm{cl}})\boldsymbol{\gamma}^{\mathrm{x}}_s ds + 2\int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta\frac{1}{2}}P^{\delta\frac{1}{2}}E\boldsymbol{w}^0 ds \tag{2.124e}$$
$$\overset{(a)}{\le} -2\rho\int_0^1 \boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta}\boldsymbol{\gamma}^{\mathrm{x}}_s ds + 2\int_0^1 \|\boldsymbol{\gamma}^{\mathrm{x}\top}_s P^{\delta\frac{1}{2}}\|\|P^{\delta\frac{1}{2}}E\boldsymbol{w}^0\| ds \tag{2.124f}$$
$$\overset{(b)}{=} -2\rho V^{\delta}(\boldsymbol{x},\boldsymbol{z}) + 2\int_0^1 \|\boldsymbol{\gamma}^{\mathrm{x}\top}_s\|_{P^{\delta}}\|E\boldsymbol{w}^0\|_{P^{\delta}} ds \tag{2.124g}$$
$$\overset{(c)}{\le} -2\rho V^{\delta}(\boldsymbol{x},\boldsymbol{z}) + 2\sqrt{V^{\delta}(\boldsymbol{x},\boldsymbol{z})}\,\bar{w}, \tag{2.124h}$$

for $\boldsymbol{w}^0 \in \mathcal{W}$ where (a) is obtained by the combination of the following contraction LMI [85]:

$$\begin{aligned} &\dot{P}^\delta(\boldsymbol{\gamma}^\mathrm{x}) + P^\delta(\boldsymbol{\gamma}^\mathrm{x})\big(A(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u}) + B(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})K^\delta(\boldsymbol{\gamma}^\mathrm{x})\big) + \\ &\big(A(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u}) + B(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})K^\delta(\boldsymbol{\gamma}^\mathrm{x})\big)^\top P^\delta(\boldsymbol{\gamma}^\mathrm{x}) + 2\rho P^\delta(\boldsymbol{\gamma}^\mathrm{x}) \preceq 0, \end{aligned} \tag{2.125}$$

with $\dot{P}^\delta(\boldsymbol{\gamma}^\mathrm{x}) := \frac{\partial P^\delta(\boldsymbol{\gamma}^\mathrm{x})}{\partial \boldsymbol{\gamma}^\mathrm{x}}\dot{\boldsymbol{\gamma}}^\mathrm{x}$ and contraction rate $\rho > 0$, which is based on the contraction analysis for nonlinear systems as presented in [86], and the Cauchy-Schwarz inequality, (b) by the definition of $V^\delta(\boldsymbol{x},\boldsymbol{z})$ in (2.116), and (c) by the following definition of $\bar{w}$:

$$\bar{w} := \max_{\substack{\boldsymbol{w}^0 \in \mathcal{W} \\ \boldsymbol{\gamma}^\mathrm{x} \in C(\boldsymbol{x},\boldsymbol{z})}} \|E\boldsymbol{w}^0\|_{\bar{P}^\delta(\boldsymbol{\gamma}^\mathrm{x})}, \tag{2.126}$$

with

$$\bar{P}^\delta(\boldsymbol{\gamma}^\mathrm{x}) := \lambda_{\max}\big(P^\delta(\boldsymbol{\gamma}^\mathrm{x})\big) I^{n^\mathrm{x}}, \tag{2.127}$$

where $\lambda_{\max}\big(P^\delta(\boldsymbol{\gamma}^\mathrm{x})\big)$ denotes the maximum eigenvalue of matrix $P^\delta(\boldsymbol{\gamma}^\mathrm{x})$.

Remark 12. *Since we use shorthand notations* (2.123a) *and* (2.123b)*, we can evaluate LMI* (2.125) *for an infinitely large number of points* $s \in [0,1]$ *on the geodesic. Thus,* (2.125) *can also be considered a set of LMIs. Since the constraints tightening is such that* $(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u}) \in \mathcal{Z}, s \in [0,1]$ *[62, Prop. 5], it suffices to grid* $\mathcal{Z}$ *in a similar way as described in Section 2.3.2.*

Remark 13. *Compared to [85],* (2.125) *with shorthand notation* (2.123)*, does not contain the time-dependency since we do not consider a time-varying system.*

Note that (2.126) confirms that Property 2 holds for compact disturbance set (2.40).

Note also that (2.125) contains the multiplication of $P^\delta(\boldsymbol{\gamma}^\mathrm{x})$ and $K^\delta(\boldsymbol{\gamma}^\mathrm{x})$. Therefore, it is a bilinear LMI in the context of an SDP in which both $P^\delta(\boldsymbol{\gamma}^\mathrm{x})$ and $K^\delta(\boldsymbol{\gamma}^\mathrm{x})$ are decision variables. Under change of coordinates $X^\delta(\boldsymbol{\gamma}^\mathrm{x}) = P^\delta(\boldsymbol{\gamma}^\mathrm{x})^{-1}$ and $Y^\delta(\boldsymbol{\gamma}^\mathrm{x}) = K^\delta(\boldsymbol{\gamma}^\mathrm{x})P^\delta(\boldsymbol{\gamma}^\mathrm{x})^{-1}$ (2.125) translates to the following convex contraction LMI [85]:

$$\begin{aligned} &-\dot{X}^\delta(\boldsymbol{\gamma}^\mathrm{x}) + A(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})X^\delta(\boldsymbol{\gamma}^\mathrm{x}) + B(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})Y^\delta(\boldsymbol{\gamma}^\mathrm{x}) + \\ &\big(A(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})X^\delta(\boldsymbol{\gamma}^\mathrm{x}) + B(\boldsymbol{\gamma}^\mathrm{x},\boldsymbol{\gamma}^\mathrm{u})Y^\delta(\boldsymbol{\gamma}^\mathrm{x})\big)^\top + 2\rho X^\delta(\boldsymbol{\gamma}^\mathrm{x}) \preceq 0. \end{aligned} \tag{2.128}$$

Thus, any combination of $P^\delta(\boldsymbol{\gamma}^\mathrm{x})$ and corresponding control law $\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})$ that is obtained by integrating over the geodesic, i.e.,

$$\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v}) = \boldsymbol{\gamma}^\mathrm{u}(1), \tag{2.129}$$

with $\boldsymbol{\gamma}^\mathrm{u}(s)$ given in (2.119), that satisfies LMI (2.128) ensures that the closed-loop trajectory $\boldsymbol{x}$ will exponentially contract towards $\boldsymbol{z}$ with rate ρ. For a more elaborate exposition of this result, please refer to [62] and references therein [37], [79], [85]–[87].

In general, $\boldsymbol{\gamma}^\mathrm{x}(s)$ is a nonlinear curve, so evaluating (2.116) as part of terminal set (2.76) in the solver is computationally expensive. To reduce solve time, we choose a state-independent incremental Lyapunov function, i.e., P^δ instead of $P^\delta(\boldsymbol{\gamma}^\mathrm{x})$, such that the geodesic becomes a straight line connecting $\boldsymbol{z}$ and $\boldsymbol{x}$. In this case, the geodesic is mathematically described by

$$\boldsymbol{\gamma}^\mathrm{x}(s) = \boldsymbol{z} + s(\boldsymbol{x} - \boldsymbol{z}) \tag{2.130}$$

and

$$\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s} = \boldsymbol{x} - \boldsymbol{z}. \tag{2.131}$$

As a result, contraction LMI (2.128) changes to

$$A(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) X^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}) + B(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) Y^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}) + \left(A(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) X^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}) + B(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) Y^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}})\right)^{\top} + 2\rho X^{\delta} \preceq 0, \tag{2.132}$$

in which the term $-\dot{X}^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}})$ disappears compared to (2.128) as the time derivative is given by $\dot{X}^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}) = \frac{\partial \dot{X}^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}})}{\partial \boldsymbol{\gamma}^{\mathrm{x}}} \dot{\boldsymbol{\gamma}}^{\mathrm{x}}$. Furthermore, incremental Lyapunov function (2.116) simplifies to the following quadratic expression:

$$V^{\delta}(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x} - \boldsymbol{z})^{T} P^{\delta}(\boldsymbol{x} - \boldsymbol{z}), \tag{2.133}$$

or, equivalently,

$$\sqrt{V^{\delta}(\boldsymbol{x}, \boldsymbol{z})} = \|\boldsymbol{x} - \boldsymbol{z}\|_{P^{\delta}}, \tag{2.134}$$

and the control law is given by

$$\kappa(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) = \boldsymbol{\gamma}^{\mathrm{u}}(1) = \boldsymbol{v} + \int_{0}^{1} K^{\delta}\left(\boldsymbol{\gamma}^{\mathrm{x}}(s)\right)(\boldsymbol{x} - \boldsymbol{z}) ds. \tag{2.135}$$

Note that this implies that (2.44a) is trivially satisfied with $c^{\delta,\mathrm{l}} = \lambda_{\min}(P^{\delta})$ and $c^{\delta,\mathrm{u}} = \lambda_{\max}(P^{\delta})$, where $\lambda_{\min}(P^{\delta})$ and $\lambda_{\max}(P^{\delta})$ denote the minimum and maximum eigenvalues of P^{δ}, respectively, and (2.45) is also satisfied. Correspondingly, we can constant $\bar{w}$ in (2.49) as

$$\bar{w} \coloneqq \max_{\boldsymbol{w}^{0} \in \mathrm{vert}(\mathcal{W})} \|E\boldsymbol{w}^{0}\|_{P^{\delta}}, \tag{2.136}$$

since $\mathcal{W}$ is a convex polytopic set. The choice for a state-independent P^{δ} increases conservatism since the resulting curve length is larger than nonlinear geodesic $\boldsymbol{\gamma}^{\mathrm{x}}(s)$ in parts of the state space where Jacobians (2.122) are highly nonlinear. Consequently, integrating the control law over this curve results in a sub-optimal control strategy. This control strategy will result in a contraction rate of at least ρ, but it will be slightly less effective than the control strategy based on a nonlinear geodesic.

To reduce conservatism, we leverage a state-dependent feedback gain $K^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}})$. $K^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}})$ is computed outside the solver using (2.129), so computation time is less important for enabling real-time implementation.

Remark 14. *Note that CCM methods provide a way to ensure uniform exponential stability, i.e., exponential stability guarantees that hold in the complete state space, even though metric $P^{\delta}(\boldsymbol{x})$ is state-dependent. This contrasts with the design of tracking LMI (2.27), which includes reference-dependent matrix $P(\boldsymbol{r})$, but is only valid in local regions ϵ around the reference trajectory. In practice, the value of ϵ for which the results hold is hard to obtain. Therefore, to obtain strict safety guarantees in the robust case, it is practically recommended to use CCMs according to the design above or remove the state dependency in P^{δ}. Using CCMs, uniform exponential stabilizability follows from computing integral expressions in the metric space. Using state-independent P^{δ}, uniform exponential stabilizability follows as a direct consequence of the mean value theorem (MVT) since choosing a constant metric P^{δ} results in a straight line minimizing geodesic $\boldsymbol{\gamma}^{\mathrm{x}}(s)$ between $\boldsymbol{z}$ and $\boldsymbol{x}$ as described above.*

Contraction LMI (2.128) is the main property that enables the robust recursive feasibility and trajectory tracking proofs to hold. The resulting metric P^δ is used to construct incremental Lyapunov function (2.116) that satisfies (2.44d). Consequently, tube (2.50) grows to a robust positive invariant (RPI) set:

$$\sqrt{V^\delta(\boldsymbol{x}_t, \boldsymbol{z}_t)} \leq \bar{s}, \ t \geq 0 \tag{2.137}$$

with $\bar{s}$ defined in (2.53). To reduce conservatism we optimize the shape of (2.137) by satisfying the following RPI LMI:

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + \lambda X^\delta & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top & -\lambda\delta^2 \end{bmatrix} \preceq 0, \tag{2.138}$$

with $\boldsymbol{\zeta} \in \mathscr{Z}$, $\boldsymbol{w}^0 \in \text{vert}(\mathscr{W})$ and multiplier $\lambda \geq 0$. This LMI enforces that sublevel set $V^\delta(\boldsymbol{x}, \boldsymbol{z}) \leq \delta^2$ is RPI for disturbed dynamics (2.39) for all $\boldsymbol{w}^0 \in \mathscr{W}$ and with $\delta := \bar{s}$. Note that ρ is a hyperparameter and $\bar{w}$ can be computed after optimizing for P^δ using (2.136). Therefore, without loss of generality, we can set $\delta = 1$, which helps to bridge between expressions in quadratic form $V^\delta(\boldsymbol{x}, \boldsymbol{z})$ and linear form $\sqrt{V^\delta(\boldsymbol{x}, \boldsymbol{z})}$. Furthermore, note that, since the disturbances enter linearly in system dynamics (2.39) and the set $\mathscr{W}$ is convex, it suffices to evaluate a set of LMIs of the form (2.138) at the vertices of $\mathscr{W}$. Appendix A.4 provides the proof for (2.138).

Remark 15. *An alternative to designing the RPI set using polytopic set $\mathscr{W}$ is to leverage universal $\mathscr{L}_\infty$-gain bounds on disturbances $\boldsymbol{w}^0$ as presented in [37].*

In contrast to the TMPC design, in which terminal set scaling α is computed based on the desired distance to obstacles $d_{\max}$, the terminal set and corresponding RPI set of $V^\delta(\boldsymbol{x}, \boldsymbol{z})$ have a minimum size caused by disturbances $\boldsymbol{w}^0 \in \mathscr{W}$. To prevent the sets from becoming large, thus resulting in conservative behavior, P^δ and $K^\delta(\boldsymbol{x})$ are computed using SDP (2.139), in which the tightening constants $c_j^{\text{s}}, j \in \mathbb{N}_{[1,n^{\text{s}}]}$ and c^{o}, normalized

with respect to the corresponding constraint interval, are minimized:

$$\min_{\substack{X^\delta, Y^\delta(\boldsymbol{x}) \\ c_j^{\mathrm{s}2}, c^{\mathrm{o}2}}} \; c^{\mathrm{c,o}} c^{\mathrm{o}2} + \sum_{j=1}^{n^{\mathrm{s}}} c_j^{\mathrm{c,s}} c_j^{\mathrm{s}2}, \tag{2.139a}$$

$$\text{s.t. } X^\delta \succeq 0, \tag{2.139b}$$

$$A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + 2\rho X^\delta \preceq 0, \tag{2.139c}$$

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + \lambda X^\delta & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top & -\lambda \end{bmatrix} \preceq 0, \tag{2.139d}$$

$$\begin{bmatrix} c_j^{\mathrm{s}2} & L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix} \\ \left(L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix}\right)^\top & X^\delta \end{bmatrix} \succeq 0, \; j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.139e}$$

$$\begin{bmatrix} c^{\mathrm{o}2} I^{n^{\mathrm{p}}} & MX^\delta \\ (MX^\delta)^\top & X^\delta \end{bmatrix} \succeq 0, \tag{2.139f}$$

$$\boldsymbol{\zeta} \in \mathcal{Z}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\mathcal{W}),$$

with contraction rate $\rho > 0$ and multiplier $\lambda \geq 0$ that can be computed using bi-section over this SDP and $\delta = 1$.

Note that LMI (2.139e), to ensure Lipschitz continuity of the system constraints, is the same as (2.38). We can derive similar a LMI (2.139f) to ensure Lipschitz continuity of the obstacle avoidance constraints. The proof is similar to the proof for (2.38) as written in [74] and included in Appendix A.5 for completeness.

Similar to SDP (2.29), SDP (2.139) is semi-infinite, so we need to evaluate the expressions at grid points and vertices, both in $\mathcal{Z}$ and $\mathcal{W}$ to account for the effect of disturbances.

Remark 16. *Note that LMI (2.139d) has to be evaluated at all grid points in both $\mathcal{Z}$ and $\mathcal{W}$, meaning that the number of LMIs equals the multiplication of the number of grid points in both sets. This might result in a drastic increase in (offline) computation time. To alleviate this effect, we can introduce decision variable $\bar{\mathcal{W}}$ and use the following combination of LMIs that is sufficient to ensure the satisfaction of (2.139d):*

$$\begin{bmatrix} 0^{n^{\mathrm{x}}} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top & 0 \end{bmatrix} \preceq \bar{\mathcal{W}}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\mathcal{W}), \tag{2.140}$$

$$\bar{\mathcal{W}} + \begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + \lambda X^\delta & \mathbf{0} \\ \mathbf{0} & -\lambda \end{bmatrix} \preceq 0, \boldsymbol{\zeta} \in \mathcal{Z}. \tag{2.141}$$

In conclusion, by successfully solving (2.139), we can find matrices P^δ and $K^\delta(\boldsymbol{x})$ such that incremental Lyapunov function $V^\delta(\boldsymbol{x}, \boldsymbol{z})$ is given by (2.133) and satisfies (2.44a) and (2.45). Furthermore, the design of contraction LMI (2.128), RPI LMI (2.138), and Lipschitz LMI (2.38) and (2.139f) ensures satisfaction of (2.44b), (2.44c), and (2.44d).

Therefore, we can conclude that this design satisfies Assumption 3.

Terminal Ingredients Design

The terminal ingredients design involves computing a suitable terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})$ such that Assumption 4 is satisfied. Given the design in the previous section, we know that $\sqrt{V^{\delta}(\boldsymbol{x},\boldsymbol{z})}$ in (2.134) decreases with at least factor ρ, or, equivalently, $V^{\delta}(\boldsymbol{x},\boldsymbol{z})$ decreases with at least factor 2ρ, given control law $\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})$ in (2.135). Therefore, if we design the terminal control law as

$$\kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r}) = \boldsymbol{u}^{\mathrm{r}} + \kappa^{\delta}(\boldsymbol{x},\boldsymbol{x}^{\mathrm{r}}) = \boldsymbol{u}^{\mathrm{r}} + \int_0^1 K^{\delta}\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big)\,ds\,(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}), \tag{2.142}$$

we just need to come up with a matrix P such that the following descent bound based on (2.54) is satisfied:

$$\frac{d}{dt}\big((\boldsymbol{x}-\boldsymbol{z})^{\top}P(\boldsymbol{x}-\boldsymbol{z})\big) \leq -\mathscr{J}^{\mathrm{s}}(\boldsymbol{z},\boldsymbol{v},\boldsymbol{r}), \tag{2.143}$$

with $\dot{\boldsymbol{x}} = f\big(\boldsymbol{x},\kappa(\boldsymbol{x},\boldsymbol{z},\boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}}$ and $\dot{\boldsymbol{z}} = f(\boldsymbol{z},\boldsymbol{v}) + E\boldsymbol{w}^{\mathrm{b}}$. The following LMI ensures that this condition holds, see also [38]:

$$\big(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)^{\top}P + P\big(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big) \preceq -Q - K^{\delta}(\boldsymbol{x})^{\top}RK^{\delta}(\boldsymbol{x}), \quad \boldsymbol{\zeta}\in\mathscr{Z}. \tag{2.144}$$

Thus, in order to find P with the lowest eigenvalues satisfying (2.54), we solve the following SDP:

$$\min_{P} \ \operatorname{trace} P, \tag{2.145a}$$

$$\text{s.t. } \big(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)^{\top}P + P\big(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big) \preceq -Q - K^{\delta}(\boldsymbol{x})^{\top}RK^{\delta}(\boldsymbol{x}), \tag{2.145b}$$

$$\boldsymbol{\zeta}\in\mathscr{Z}.$$

Remark 17. *A simpler but more conservative solution would be to define constant scaling factor μ such that $P = \mu P^{\delta}$. It can be shown that a lower bound for μ can be formulated as a function of $Q, R, K^{\delta}(\boldsymbol{x}), \rho$, and $c^{\delta,\mathrm{u}}$ such that the contraction of $V^{\delta}(\boldsymbol{x},\boldsymbol{z})$ by 2ρ implies that the minimum terminal cost decrease in (2.54) is satisfied.*

Thus, we can conclude that Assumption 4 is satisfied.

Reference Trajectory Design

Assumptions 5 and 6 are trivially satisfied if the system and obstacle avoidance constraints used for generating the reference trajectory are constructed according to (2.83) and (2.84) using tightening constants c^{s}_j, $j\in\mathbb{N}_{[1,n^{\mathrm{s}}]}$ and c^{o} from SDP (2.139), respectively, and α satisfying the lower bound computed in (2.82). Correspondingly, terminal set (2.76) is suitably designed around the reference trajectory to ensure closed-loop system and obstacle avoidance constraints satisfaction at all times.

2.4.4. CONCLUDING REMARKS

In conclusion, the design of the obstacle avoidance constraints, incremental Lyapunov function, terminal ingredients, and reference trajectory as presented above results in the satisfaction of Assumptions 2, 3, 4, 5, and 6. These assumptions enable the recursive feasibility and trajectory tracking proofs, thereby proving Theorem 2. Thus, RMPC scheme (2.42) is an effective tool to provide safety guarantees for tracking dynamically feasible reference trajectories with a mobile robot described by disturbed dynamics (2.39).

The scheme is based on the idea that the impact of disturbances can be bounded by a sublevel set of incremental Lyapunov function $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$ given a suitably designed control law. This bound, which is also called the tube size, can be used to tighten system and obstacle avoidance constraints such that the closed-loop system always satisfies the actual system and obstacle avoidance constraints. Recursive feasibility is proven by constructing a minimum invariant terminal set based on $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$. Furthermore, trajectory tracking is proven by showing that the terminal cost matrix P can be computed using the weighting matrix P^{δ} used in $V^{\delta}(\boldsymbol{x}, \boldsymbol{z})$.

Note that tightening the system and obstacle avoidance constraints of the reference trajectory by $c_j^{\mathrm{s}}\alpha$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and $c^{\mathrm{o}}\alpha$, respectively, is a sufficient condition to guarantee closed-loop constraints satisfaction. However, a feasible solution is not required to start at least this far away from the constraints at the beginning of the prediction horizon since the constraints are tightened by an increasing tube size that starts from zero. Therefore, implementing a similar suitably designed constraint tightening scheme in the planner generating the reference trajectory will allow for less conservative motions, i.e., motions closer to obstacles, thereby reaching goals quicker.

In practice, a physical robot is not only subject to dynamic disturbances but also to noisy sensor measurements. This means that we cannot directly measure the states of the system. To provide safety guarantees in the context of noisy measurements, the next section presents a similar step-by-step approach to the one in this section including an explicit uncertainty description of the measurements.

2.5. ROBUST OUTPUT-FEEDBACK MPC FOR TRAJECTORY TRACKING WITH DISTURBANCES AND MEASUREMENT NOISE

The goal of this section is to introduce a ROMPC scheme that will be used to track reference trajectories satisfying Property 1 when the system is described as given in Section 2.5.1. In this system, the state can only be measured indirectly via noisy output measurements. Like Section 2.3.1, Section 2.5.2 states the optimization problem and control law for the closed-loop system. Then, Section 2.5.3 builds an intuition for ROMPC and explains the required elements in the formulation and related properties. Finally, Section 2.5.4 gives concluding remarks on the ROMPC formulation.

2.5.1. DISTURBED MOBILE ROBOT WITH MEASUREMENT NOISE DESCRIPTION

The disturbed system dynamics are given by (2.39). The corresponding output equation is given by

$$\boldsymbol{y} = C\boldsymbol{x} + F\boldsymbol{\eta}, \tag{2.146}$$

with measured output $\boldsymbol{y} \in \mathbb{R}^{n^{\mathrm{y}}}$, observation matrix $C \in \mathbb{R}^{n^{\mathrm{y}} \times n^{\mathrm{x}}}$, measurement or sensor noise $\boldsymbol{\eta} \in \mathbb{R}^{n^{\eta}}$, and noise selection matrix $F \in \mathbb{R}^{n^{\mathrm{y}} \times n^{\eta}}$.

The measurement noise is assumed to be contained in a known bounding box $\boldsymbol{\eta} \in \mathscr{H}$. By recording data we can compute bounding box $\mathscr{H}$ with lower bound $\underline{\boldsymbol{\eta}}_i^{\mathrm{rec}}$ and upper bound $\bar{\boldsymbol{\eta}}_i^{\mathrm{rec}}$ for each dimension $i \in \mathbb{N}_{[1,n^{\eta}]}$ of the recorded measurement noise values:

$$\mathscr{H} := \left\{ \boldsymbol{\eta} \in \mathbb{R}^{n^{\eta}} \;\middle|\; \underline{\boldsymbol{\eta}}_i^{\mathrm{rec}} \leq \boldsymbol{\eta}_i \leq \bar{\boldsymbol{\eta}}_i^{\mathrm{rec}},\ i \in \mathbb{N}_{[1,n^{\eta}]} \right\}. \tag{2.147}$$

This measurement noise bounding box can be used in the ROMPC formulation described in the next section.

2.5.2. ROMPC FORMULATION

Consider the following ROMPC formulation for tracking reference trajectory (2.4):

$$\mathscr{J}_t^*(\boldsymbol{x}_t, \boldsymbol{r}_t) = \min_{\boldsymbol{z}_{\cdot|t}, \boldsymbol{v}_{\cdot|t}} \ \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}_{T|t}, \boldsymbol{x}_{t+T}^{\mathrm{r}}) + \int_0^T \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}, \boldsymbol{r}_{t+\tau})\, d\tau, \tag{2.148a}$$

$$\text{s.t. } \boldsymbol{z}_{0|t} = \hat{\boldsymbol{x}}_t, \tag{2.148b}$$

$$s_0 = 0, \tag{2.148c}$$

$$\dot{\boldsymbol{z}}_{\tau|t} = f(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}) + E\boldsymbol{w}^{\mathrm{b}}, \qquad \tau \in [0, T], \tag{2.148d}$$

$$\dot{s}_\tau = -\rho s_\tau + \bar{w}^{\mathrm{o}}, \qquad \tau \in [0, T], \tag{2.148e}$$

$$g_j^{\mathrm{s}}(\boldsymbol{z}_{\tau|t}, \boldsymbol{v}_{\tau|t}) + c_j^{\mathrm{s}} s_\tau + c_j^{\mathrm{s,o}} \epsilon \leq 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]},\ \tau \in [0, T], \tag{2.148f}$$

$$g_{j,\tau|t}^{\mathrm{o}}(M\boldsymbol{z}_{\tau|t}) + c^{\mathrm{o}}(s_\tau + \epsilon) \leq 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{o}}]},\ \tau \in [0, T], \tag{2.148g}$$

$$(\boldsymbol{z}_{T|t}, s_\tau, \epsilon) \in \mathscr{X}^{\mathrm{f}}(\boldsymbol{x}_{t+T}^{\mathrm{r}}), \tag{2.148h}$$

with estimated state $\hat{\boldsymbol{x}}_t$, stage cost $\mathscr{J}^{\mathrm{s}}(\boldsymbol{z}, \boldsymbol{v}, \boldsymbol{r}) = \|\boldsymbol{z} - \boldsymbol{x}^{\mathrm{r}}\|_Q^2 + \|\boldsymbol{v} - \boldsymbol{u}^{\mathrm{r}}\|_R^2, Q \succ 0, R \succ 0$ and terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}}) = \|\boldsymbol{z} - \boldsymbol{x}^{\mathrm{r}}\|_P^2, P \succ 0$. In this formulation, Q and R are tuning matrices, and P should be suitably computed to show practical asymptotic convergence; see Section 2.5.3.

Similar to the TMPC formulation in Section 2.3, the ROMPC formulation is applied receding-horizon. In this case, the control law for the closed-loop system is defined as follows for $t \geq 0$:

$$\boldsymbol{u}_{t+\tau} = \kappa(\hat{\boldsymbol{x}}_{t+\tau}, \boldsymbol{z}_{\tau|t}^*, \boldsymbol{v}_{\tau|t}^*),\ \tau \in [0, T^{\mathrm{s}}], \tag{2.149}$$

with control law $\kappa(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{v}) : \mathscr{X} \times \mathscr{Z} \to \mathbb{R}^{n^{\mathrm{u}}}$ as defined in (2.43) that will be detailed later according to the design presented in this section.

2.5.3. ROMPC PROPERTIES

The main difference between ROMPC and RMPC is that the actual states $\boldsymbol{x}$ are hidden, and we can only measure noisy outputs $\boldsymbol{y}$. A logical choice in the ROMPC formulation is to predict a nominal trajectory starting at some estimated $\hat{\boldsymbol{x}}$ based on $\boldsymbol{y}$ instead of $\boldsymbol{x}$. This trajectory will satisfy tightened constraints (2.148f), (2.148g) and (2.148h). However, tightening these constraints with tube dynamics (2.42e) is not sufficient to guarantee satisfaction of the non-tightened constraints for the closed-loop trajectory since $\boldsymbol{x}$ might be closer to the constraints than $\boldsymbol{y}$. Therefore, we want to increase the tightening to account for this additional observer error.

To determine the additional required tightening, the error $\boldsymbol{y} - C\boldsymbol{x}$ needs to be bounded. This bound is given by the compact set $\mathscr{H}$ described in (2.147). Furthermore, as further detailed in Section 2.5.3, we need to have an explicit expression for the derivative $\frac{d}{dt}(\boldsymbol{y} - C\boldsymbol{x})$ to determine the size of the tightening. Since this information is unavailable, we design an observer to estimate state $\hat{\boldsymbol{x}}$ given output $\boldsymbol{y}$ with the following dynamics:

$$\dot{\hat{\boldsymbol{x}}} := f(\hat{\boldsymbol{x}}, \boldsymbol{u}) + E\boldsymbol{w}^{\mathrm{b}} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}}), \tag{2.150}$$

with $\boldsymbol{u}$ given in (2.149) and observer gain matrix $L \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{y}}}$, which serves as a low-pass filter on measurements $\boldsymbol{y}$.

Similar to the properties described in Section 2.4.3 and given observer (2.150), we want to design a feedback law $\kappa^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ that renders tube size s of incremental Lyapunov function $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ invariant. In this case, $\kappa^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ are based on a geodesic that ends at $\hat{\boldsymbol{x}}$ rather than $\boldsymbol{x}$. Such a combination of $\kappa^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ exists if the system satisfies the following assumption:

Assumption 7 (Incremental stabilizability)**.** *There exist a feedback law $\kappa^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) : \mathscr{X}^2 \to \mathbb{R}^{n^{\mathrm{u}}}$, an incremental Lyapunov function $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) : \mathscr{X}^2 \to \mathbb{R}_{\geq 0}$ that is continuously differentiable and satisfies $V^{\delta}(\boldsymbol{z}, \boldsymbol{z}) = 0, \forall \boldsymbol{z} \in \mathscr{X}$, parameters $c^{\delta,\mathrm{l}}, c^{\delta,\mathrm{u}}, \rho > 0$, and Lipschitz constants $c_j^{\mathrm{s}} > 0, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and $c^{\mathrm{o}} > 0$, such that the following properties hold for all $(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v}) \in \mathscr{X} \times \mathscr{Z}$, $\boldsymbol{w} \in \boldsymbol{w}^{\mathrm{b}} \oplus \mathscr{W}$, and $\boldsymbol{\eta} \in \mathscr{H}$:*

$$c^{\delta,\mathrm{l}} \|\hat{\boldsymbol{x}} - \boldsymbol{z}\|^2 \leq V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq c^{\delta,\mathrm{u}} \|\hat{\boldsymbol{x}} - \boldsymbol{z}\|^2, \tag{2.151a}$$

$$g_j^{\mathrm{s}}\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - g_j^{\mathrm{s}}(\boldsymbol{z}, \boldsymbol{v}) \leq c_j^{\mathrm{s}} \sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})}, \; j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.151b}$$

$$g^{\mathrm{o}}(M\hat{\boldsymbol{x}}) - g^{\mathrm{o}}(M\boldsymbol{z}) \leq c^{\mathrm{o}} \sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})}, \tag{2.151c}$$

$$\frac{d}{dt} V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq -2\rho V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}), \tag{2.151d}$$

with $\dot{\hat{\boldsymbol{x}}} = f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}})$, $\dot{\boldsymbol{z}} = f(\boldsymbol{z}, \boldsymbol{v}) + E\boldsymbol{w}^{\mathrm{b}}$, $\boldsymbol{y}$ defined in (2.146), and $\dot{\boldsymbol{x}} = f\big(\boldsymbol{x}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}}$. Furthermore, the following norm-like inequality holds $\forall \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in \mathbb{R}^{n^{\mathrm{x}}}$:

$$\sqrt{V^{\delta}(\boldsymbol{x}_1, \boldsymbol{x}_3)} \leq \sqrt{V^{\delta}(\boldsymbol{x}_1, \boldsymbol{x}_2)} + \sqrt{V^{\delta}(\boldsymbol{x}_2, \boldsymbol{x}_3)}. \tag{2.152}$$

This assumption reflects the same properties as Assumption 3 in Section 2.4.3 with the only difference that it is defined with $\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x}$ using observer (2.150). Following similar arguments as in Section 2.4.3, we know that the time-derivative of $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ can be upper-bounded:

Property 3. *There exists a constant* $\bar{w}^{\mathrm{o}} > 0$ *such that for any* $(\boldsymbol{y}, \hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v}) \in \mathbb{R}^{n^{\mathrm{y}}} \times \mathbb{R}^{n^{\mathrm{x}}} \times \mathcal{Z}$, *any* $\boldsymbol{w} \in \boldsymbol{w}^{\mathrm{b}} \oplus \mathcal{W}$, *and any* $\boldsymbol{\eta} \in \mathcal{H}$, *it holds that*

$$\frac{d}{dt}\sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} \leq -\rho\sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} + \bar{w}^{\mathrm{o}}, \tag{2.153}$$

with $\dot{\hat{\boldsymbol{x}}} = f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}})$, $\dot{\boldsymbol{z}} = f(\boldsymbol{z}, \boldsymbol{v}) + E\boldsymbol{w}^{\mathrm{b}}$, $\boldsymbol{y}$ *defined in* (2.146), *and* $\dot{\boldsymbol{x}} = f\big(\boldsymbol{x}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}$.

Note that a different constant $\bar{w}^{\mathrm{o}} \neq \bar{w}$ is used in this case. Whereas $\bar{w}$ accounts for the impact of disturbances, $\bar{w}^{\mathrm{o}}$ accounts for the impact of both disturbances and measurement noise.

By Property 3, we know that $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ is upper-bounded by tube size s:

$$\sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} \leq s, \tag{2.154}$$

with the tube dynamics given by

$$\dot{s} = -\rho s + \bar{w}^{\mathrm{o}}, \tag{2.155}$$

which is the same as (2.148e). Rewriting as a function of time gives

$$s_t = (1 - e^{-\rho t})\frac{\bar{w}^{\mathrm{o}}}{\rho}, \tag{2.156}$$

and taking the limit $t \to \infty$ results in a maximum tube size of

$$\bar{s} = \frac{\bar{w}^{\mathrm{o}}}{\rho}. \tag{2.157}$$

Thus, s does not grow unbounded, and the resulting tube size (2.157) is suitable for tightening the constraints.

Note that contraction property (2.151d) and corresponding Property 3 and tube dynamics (2.148e) are used to upper-bound the controller error

$$\boldsymbol{\delta} := \hat{\boldsymbol{x}} - \boldsymbol{z}. \tag{2.158}$$

Thus, tightening using s results in closed-loop safety guarantees for $\hat{\boldsymbol{x}}$, not for $\boldsymbol{x}$.

To account for the observer error

$$\boldsymbol{\epsilon} := \boldsymbol{x} - \hat{\boldsymbol{x}}, \tag{2.159}$$

we can make the following assumption:

Assumption 8 (Bounded observer error)**.** *There observer gain matrix* L *is such that the observer error* $\boldsymbol{x} - \hat{\boldsymbol{x}}$ *is bounded and described by an* ϵ*-sublevel set of* $\sqrt{V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})}$*:*

$$\sqrt{V^{\delta}(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t)} \leq \epsilon, \ \forall\, t \geq 0. \tag{2.160}$$

This assumption implies that we can additionally tighten the system and obstacle avoidance constraints proportional to ϵ in order to guarantee closed-loop safety of both $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$. Note that the tightening proportional to ϵ only applies to the state constraints, not the input constraints, since control law (2.149) is independent of $\boldsymbol{x}$. Therefore, we introduce additional tightening constants $c_j^{\mathrm{s,o}},\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$:

$$c_j^{\mathrm{s,o}} = \begin{cases} 0, & j \in \mathbb{N}_{[1,2n^{\mathrm{u}}]} \\ c_j^{\mathrm{s}}, & j \in \mathbb{N}_{[2n^{\mathrm{u}}+1,n^{\mathrm{s}}]} \end{cases}. \tag{2.161}$$

Similar to Section 2.4.3, for safe and efficient mobile robot navigation using MPC subject to disturbances and noisy output measurements, we need to ensure that the ROMPC formulation (2.148) is recursively feasible and the closed-loop system converges to the reference trajectory. This is formalized in the following theorem:

Theorem 3. *There exist terminal control law κ^{f}, terminal cost $\mathscr{J}^{\mathrm{f}}$, and terminal set $\mathscr{X}^{\mathrm{f}}$ such that, if (2.148) is feasible at $t = 0$, closed-loop system (2.149) is recursively feasible, satisfies system constraints (2.2) and avoids obstacles for all $t \geq 0$, and, on average, the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ gets small over time.*

Proof. The proof is this theorem is provided throughout the rest of this section and structured as follows: first, the existence of the terminal ingredients is assumed. Together with a specific candidate solution, they enable proving recursive feasibility and that the tracking error gets small over time, on average. Finally, design choices are described that fulfil all assumptions made to complete the recursive feasibility and trajectory tracking proofs, thereby completing the proof of this theorem. □

Figure 2.3 provides intuition for the proofs related to ROMPC formulation (2.148). Compared to the RMPC case, instead of $\boldsymbol{x}$, we can only obtain estimated state $\hat{\boldsymbol{x}}$ via noisy output measurements $\boldsymbol{y}$. However, by suitably designing observer (2.150), we know that $\boldsymbol{x}$ is contained in a tube around $\hat{\boldsymbol{x}}$. Therefore, the system and obstacle avoidance constraints are tightened by an additional term propertional to this tube size. Furthermore, the tube size used to describe the difference between $\hat{\boldsymbol{x}}$ and $\boldsymbol{z}$ starts from zero at the beginning of the prediction horizon and is strictly larger than the tube size designed in Section 2.4.3 for both the previously optimal and the candidate solution. Given a suitably designed control law and incremental Lyapunov function, the candidate contracts with at least factor ρ towards the previously optimal solution. Hence, with the adjusted tube sizes above, we can apply a similar reasoning as in Section 2.4.3 to prove recursive feasibility.

Since the feedback law is based on $\hat{\boldsymbol{x}}$ rather than $\boldsymbol{x}$, the candidate might contract slower to the previously optimal solution compared to the case where $\boldsymbol{x}$ is used to compute the feedback law. This worst-case effect results in a larger tracking error between $\boldsymbol{x}$ and the reference trajectory. Since we know that $\boldsymbol{x} - \hat{\boldsymbol{x}}$ is bounded, we can still conclude that, on average, the tracking error gets small. However, the bound is less strict than the one obtained in Section 2.4.3.

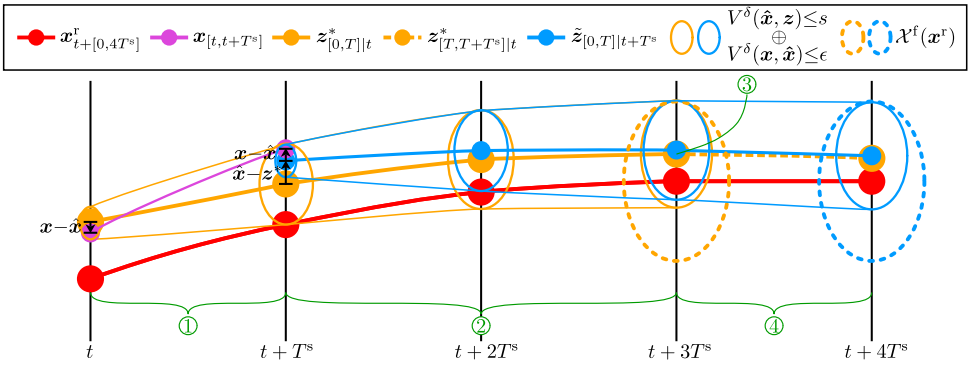


Figure 2.3: Visualization of a 1D reference trajectory $x^{\mathrm{r}}_{t+[0,4T^{\mathrm{s}}]}$, actual state trajectory $x_{[t,t+T^{\mathrm{s}}]}$, previously optimal solution $z^*_{[0,T]|t}$ starting from estimated state $\hat{x}_t$, appended part of previously optimal solution $z^*_{[T,T+T^{\mathrm{s}}]|t}$, candidate solution $\tilde{z}_{[0,T]|t+T^{\mathrm{s}}}$ starting from estimated state $\hat{x}_{t+T^{\mathrm{s}}}$, tubes $V^\delta(\hat{x},z^*)\leq s$ and $V^\delta(x,\hat{x})\leq\epsilon$ around $z^*_{[0,T]|t}$, tubes $V^\delta(\hat{x},\tilde{z})\leq s$ and $V^\delta(x,\hat{x})\leq\epsilon$ around $\tilde{z}_{[0,T]|t+T^{\mathrm{s}}}$, and terminal sets $\mathcal{X}^{\mathrm{f}}(x^{\mathrm{r}})$ around the reference trajectory corresponding to the previously optimal and the candidate solution. Again, we set $T=4T^{\mathrm{s}}$. Note that the Minkowski sum of tubes s and ϵ is plotted around the nominal predicted trajectory as the combination of these tubes is used to tighten the constraints. Compared to Figure 2.2, the initial tube size is centered around $\hat{x}$, has a non-zero size, and contains state x. Again, w impacts the system during interval ①. Thus, the differences $\tilde{z}_{T^{\mathrm{s}}|t}-\hat{x}_{t+T^{\mathrm{s}}}$ and $x_{t+T^{\mathrm{s}}}-\hat{x}_{t+T^{\mathrm{s}}}$ are both the result of the combination of w, η, and observer dynamics (2.150). Note that the tubes and terminal sets are visualized as ellipses to clarify their shapes, whereas they should be visualized as vertical lines in this case.

ASSUMPTION ON EXISTENCE TERMINAL INGREDIENTS

Similar to Section 2.4.3, we need to assume the existence of the following terminal ingredients:

Assumption 9 (Terminal ingredients)**.** *There exist a terminal control law* $\kappa^{\mathrm{f}}(\hat{x},r):\mathcal{X}\times\bar{\mathcal{Z}}\to\mathbb{R}^{n^{\mathrm{u}}}$ *and terminal cost* $\mathscr{J}^{\mathrm{f}}(z,x^{\mathrm{r}}):\mathcal{X}\times\bar{\mathcal{X}}\to\mathbb{R}$ *such that the following property holds for all* $z\in\mathcal{X}$*:*

$$\mathscr{J}^{\mathrm{f}}(z^*_{T^{\mathrm{s}}+T|t},x^{\mathrm{r}}_{t+T^{\mathrm{s}}+T})-\mathscr{J}^{\mathrm{f}}(z^*_{T|t},x^{\mathrm{r}}_{t+T})\leq-\int_T^{T+T^{\mathrm{s}}}\mathscr{J}^{\mathrm{s}}(z^*_{\tau|t},v^*_{\tau|t},r_{t+\tau})d\tau, \tag{2.162}$$

with sampling time $T^{\mathrm{s}}>0$*, optimal solution* $(z^*_{\cdot|t},v^*_{\cdot|t})$ *at* t *for all* $t\geq 0$ *and its extended version* $z^*_{T+\tau|t}$ *using terminal control law* $\kappa^{\mathrm{f}}(\hat{x},r)$ *for* $\tau\in[0,T^{\mathrm{s}}]$*, as defined in Section 2.5.3.*

Similar to Section 2.4.3, next to this assumption on terminal control law $\kappa^{\mathrm{f}}(\hat{x},r)$ and terminal cost $\mathscr{J}^{\mathrm{f}}(z,x^{\mathrm{r}})$, we want to design terminal set $\mathcal{X}^{\mathrm{f}}(x^{\mathrm{r}})$ such that tightened system and obstacle avoidance constraints (2.148f) and (2.148g) are satisfied in $\mathcal{X}^{\mathrm{f}}(x^{\mathrm{r}})$, respectively, and such that terminal control law $\kappa^{\mathrm{f}}(\hat{x},r)$ ensures robust positive invariance of $\mathcal{X}^{\mathrm{f}}(x^{\mathrm{r}})$. The specific formulation of a tight version of $\mathcal{X}^{\mathrm{f}}(x^{\mathrm{r}})$ is presented in Section 2.5.3 as part of the proof and therefore not included in Assumption 9.

CANDIDATE SOLUTION

Similar to Section 2.4.3, the goal of defining the candidate solution is to show that there exists a feasible, not necessarily optimal, solution to (2.148) at time $t+T^{\mathrm{s}}$ given that

(2.148) is successfully solved at time t. Again, this candidate solution is used to prove trajectory tracking and recursive feasibility, but it is not necessarily implemented in practice.

In the robust output-feedback case, we pick a candidate solution similar to the nominal case. For convenience, we define for $\tau \in [T, T+T^{\mathrm{s}}]$:

$$\boldsymbol{v}^*_{\tau|t} = \kappa^{\mathrm{f}}(\boldsymbol{z}^*_{\tau|t}, \boldsymbol{r}_{t+\tau}), \tag{2.163}$$

with $\dot{\boldsymbol{z}}^*_{\tau|t}$ given by (2.1) by applying (2.163).

Given these definitions, we can write the candidate for prediction interval $\tau \in [0, T]$ at time $t + T^{\mathrm{s}}$ as

$$\tilde{\boldsymbol{z}}_{0|t+T^{\mathrm{s}}} = \hat{\boldsymbol{x}}_{t+T^{\mathrm{s}}}, \tag{2.164a}$$

$$\tilde{s}_0 = 0, \tag{2.164b}$$

$$\tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}} = \kappa(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}), \tag{2.164c}$$

with $\dot{\tilde{\boldsymbol{z}}}_{\tau|t+T^{\mathrm{s}}}$ and $\dot{\tilde{s}}_\tau$ according to (2.148d) and (2.148e), respectively.

Note that this candidate deviates from the one in Section 2.4.3. The main difference is the initial estimated state. Consequently, feedback control law (2.164c) gives different values compared to (2.56c) in Section 2.4.3.

RECURSIVE FEASIBILITY PROOF

The recursive feasibility proof follows similar arguments as in Section 2.4.3 with additional technicalities to account for measurement noise $\boldsymbol{\eta}$. Instead of optimizing from the measured state $\boldsymbol{x}$, the optimization starts from the estimated state $\hat{\boldsymbol{x}}$, and we ensure that the controller error $\boldsymbol{\delta}$ is contained inside δ-sublevel set of $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and the observer error $\boldsymbol{\epsilon}$ inside ϵ-sublevel set of $V^\delta(\boldsymbol{x}, \hat{\boldsymbol{x}})$.

To prove recursive feasibility, first note that the initial state, initial tube, system dynamics, and tube dynamics constraints are all trivially satisfied by (2.164). Again, we have to show that the system constraints (2.148f), the obstacle avoidance constraints (2.148g), and the terminal set constraint (2.148h) are satisfied for $\tau \in [0, T]$.

Similar to the procedure in Section 2.4.3, we split the interval into two subintervals to show satisfaction of (2.148f) and (2.148g): $\tau \in [0, T - T^{\mathrm{s}})$ and $\tau \in [T - T^{\mathrm{s}}, T]$. By replacing $\bar{w}$ by $\bar{w}^{\mathrm{o}}$ and additionally tightening the state constraints with a factor proportional to ϵ, we obtain trivial satisfaction of the system and obstacle avoidance constraints for $\tau \in [0, T - T^{\mathrm{s}})$ following the same procedure as in Section 2.4.3. Note that in that case the additional tightening terms proportional to ϵ cancel out in (2.62) and the same proof in Appendix A.2 holds with a different expression for $s_{T^{\mathrm{s}}}$:

$$s_{T^{\mathrm{s}}} = (1 - e^{-\rho T^{\mathrm{s}}}) \frac{\bar{w}^{\mathrm{o}}}{\rho}. \tag{2.165}$$

Terminal Set Constraint Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$

Since the tightening is increased proportional to ϵ compared to Section 2.4, we need to account for this additional tube size in the terminal set. Therefore, the terminal set is now given by

$$\mathcal{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}) := \left\{ \boldsymbol{z} \in \mathcal{X}, s_T \in \mathbb{R}, \epsilon \in \mathbb{R} \;\middle|\; \sqrt{V^\delta(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})} + s_T + \epsilon \leq \alpha \right\}. \tag{2.166}$$

To show satisfaction of terminal set constraint (2.148h) for $\tau \in [T - T^{\mathrm{s}}, T]$, we follow similar steps to the ones outlined in Remark 10 in Section 2.4.3. We start by showing that terminal set constraint (2.166) is satisfied at $\tau = T - T^{\mathrm{s}}$ at $t + T^{\mathrm{s}}$. This gives an inequality similar to (2.82a) evaluated at $\tau = 0$:

$$\alpha - s_T - \epsilon + e^{-\rho(T-T^{\mathrm{s}})} s_{T^{\mathrm{s}}} \leq \alpha - s_{T-T^{\mathrm{s}}} - \epsilon, \tag{2.167}$$

or, equivalently,

$$s_{T-T^{\mathrm{s}}} \leq s_T - e^{-\rho(T-T^{\mathrm{s}})} s_{T^{\mathrm{s}}}. \tag{2.168}$$

This is the same expression as (2.68), which holds trivially by the proof in Appendix A.2 with $s_{T^{\mathrm{s}}}$ given by (2.165).

Appendix A.6 proves that terminal set (2.166) is invariant provided that $\alpha \geq \frac{\bar{w}^{\mathrm{o}}}{\rho} + \epsilon$. Therefore, we can conclude that the terminal set constraint is also satisfied with this lower bound on α for $\tau \in [T - T^{\mathrm{s}}, T]$ at $t + T^{\mathrm{s}}$.

System Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$

Following the same reasoning as detailed in Section 2.4.3 and leveraging the proof in Appendix A.2 and the fact that (2.76) is invariant, the candidate system constraints are satisfied for $\tau \in [T - T^{\mathrm{s}}, T]$ under the following assumption:

Assumption 10 (System constraints tightening for reference trajectory)**.** *The reference trajectory satisfies the tightened systems constraints, i.e.,* (2.7) *holds, with* $g_j^{\mathrm{r,s}}(\hat{\boldsymbol{x}}, \boldsymbol{u})$ *given by*

$$g_j^{\mathrm{r,s}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = g_j^{\mathrm{s}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) + c_j^{\mathrm{s}} \frac{\bar{w}^{\mathrm{o}}}{\rho} + c_j^{\mathrm{s,o}} \epsilon, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.169}$$

where c_j^{s} *and* $c_j^{\mathrm{s,o}}$ *are defined in* (2.151b) *and* (2.161)*, respectively.*

This is the same assumption as Assumption 5 in Section 2.4.3 with the only difference that we have access to $\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x}$ and, correspondingly, the lower bound on α is larger.

Obstacle Avoidance Constraints Satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$

Following a similar reasoning as for the system constraints satisfaction for $\tau \in [T - T^{\mathrm{s}}, T]$, the candidate obstacle avoidance constraints are satisfied for $\tau \in [T - T^{\mathrm{s}}, T]$ under the following assumption:

Assumption 11 (Obstacle avoidance constraints tightening for reference trajectory)**.** *The reference trajectory satisfies the tightened obstacle avoidance constraints, i.e.,* (2.9) *holds, with* $g_j^{\mathrm{r,o}}(\hat{\boldsymbol{p}})$ *given by*

$$g_j^{\mathrm{r,o}}(\hat{\boldsymbol{p}}) = g_j^{\mathrm{o}}(\hat{\boldsymbol{p}}) + c^{\mathrm{o}} \alpha, \ j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}, \tag{2.170}$$

where c^{o} *is defined in* (2.151c) *and* $\alpha \geq \frac{\bar{w}^{\mathrm{o}}}{\rho} + \epsilon$.

Again, this assumption is the same as Assumption 6 in Section 2.4.3 with the only difference that we have availability of $\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x}$ and, correspondingly, the lower bound on α is larger. □

TRAJECTORY TRACKING PROOF

The trajectory tracking proof follows the procedure detailed in Section 2.4.3. The only difference is that we show tracking for a trajectory optimized from $\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x}$. Correspondingly, we can follow the same proof as outlined in Section 2.4.3 with the exceptions that we now leverage (2.162) in Assumption 9 instead of (2.54) in Assumption 4. The upper bound on the difference in stage costs is now given by

$$\begin{aligned}&\mathscr{J}^{\mathrm{s}}(\tilde{\boldsymbol{z}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{v}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau}) - \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{v}^*_{T^{\mathrm{s}}+\tau|t}, \boldsymbol{r}_{t+T^{\mathrm{s}}+\tau})\\ &\leq L^{\mathscr{J}^{\mathrm{s}},\mathrm{xu}} e^{-\rho\tau}(1-e^{\rho T^{\mathrm{s}}})\frac{\bar{w}^{\mathrm{o}}}{\rho} =: \alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}),\end{aligned} \tag{2.171}$$

instead of (2.100), and that the upper bound on the difference in terminal costs is given by

$$\begin{aligned}&\mathscr{J}^{\mathrm{f}}(\tilde{\boldsymbol{z}}_{T|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T})\\ &\leq L^{\mathscr{J}^{\mathrm{f}}} e^{-\rho T}(1-e^{\rho T^{\mathrm{s}}})\frac{\bar{w}^{\mathrm{o}}}{\rho} =: \alpha^{\mathrm{f}}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}),\end{aligned} \tag{2.172}$$

instead of (2.103). Following similar steps as in Section 2.4.3 with different expressions for $\alpha^{\mathrm{s}}_{\tau}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}})$, $\alpha^{\mathrm{f}}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}})$, and $\bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}})$ changes (2.109) into the following upper bound:

$$\|\boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q \overset{\text{(a)}}{\leq} c^{\mathrm{t},1}\|\boldsymbol{x}_{\tau} - \hat{\boldsymbol{x}}_{\tau}\| + c^{\mathrm{t},2}\|\hat{\boldsymbol{x}}_{\tau} - \boldsymbol{z}^*_{\tau-t|t}\| + \|\boldsymbol{z}^*_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q \tag{2.173a}$$

$$\overset{\text{(b)}}{\leq} c^{\mathrm{t},3}\epsilon + c^{\mathrm{t},4}s_{\tau} + \|\boldsymbol{z}^*_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q, \tag{2.173b}$$

where (a) is obtained by Lipschitz bounds with constants $c^{\mathrm{t},1}$ and $c^{\mathrm{t},2}$ and (b) by other Lipschitz bounds with constants $c^{\mathrm{t},3}$ and $c^{\mathrm{t},4}$ on the norms using a combination of (2.44a), (2.160), (2.154), and (2.156). Therefore, we can write a similar upper bound as in (2.110):

$$c^{\mathscr{J},\mathrm{d}}\int_t^{t+T^{\mathrm{s}}} \|\boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q d\tau \leq c^{\mathscr{J},\mathrm{d}}\int_t^{t+T^{\mathrm{s}}} \|\boldsymbol{z}^*_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q + c^{\mathrm{t},4}s_{\tau} + c^{\mathrm{t},3}\epsilon d\tau \tag{2.174a}$$

$$\overset{\text{(a)}}{=} c^{\mathscr{J},\mathrm{d}}\int_t^{t+T^{\mathrm{s}}} \|\boldsymbol{z}^*_{\tau-t|t} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q d\tau + c^{\mathscr{J},\mathrm{d}}\int_t^{t+T^{\mathrm{s}}} c^{\mathrm{t},4}s_{\tau}d\tau + c^{\mathscr{J},\mathrm{d}}\int_t^{t+T^{\mathrm{s}}} c^{\mathrm{t},3}\epsilon d\tau \tag{2.174b}$$

$$\overset{\text{(b)}}{\leq} \mathscr{J}^*_t - \mathscr{J}^*_{t+T^{\mathrm{s}}} + \bar{\alpha}^{\mathrm{s,f},1}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}), \tag{2.174c}$$

with $\bar{\alpha}^{\mathrm{s,f},1}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}) := \bar{\alpha}^{\mathrm{s,f}}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}) + c^{\mathscr{J},\mathrm{d}}c^{\mathrm{t},4}\big(T^{\mathrm{s}} - \frac{1}{\rho}(1-e^{-\rho T^{\mathrm{s}}})\big)\frac{\bar{w}^{\mathrm{o}}}{\rho} + c^{\mathscr{J},\mathrm{d}}T^{\mathrm{s}}c^{\mathrm{t},3}\epsilon$, and where (a) is obtained by writing out the integral expressions and (b) by (2.108). Following the same steps as in Section 2.4.3, we obtain upper bound

$$\lim_{t\to\infty}\frac{\int_0^t \|\boldsymbol{x}_{\tau} - \boldsymbol{x}^{\mathrm{r}}_{\tau}\|^2_Q d\tau}{t} \leq \frac{\mathscr{J}^*_0 - \lim_{t\to\infty}\mathscr{J}^*_t}{c^{\mathscr{J},\mathrm{d}}t} + \frac{1}{c^{\mathscr{J},\mathrm{d}}T^{\mathrm{s}}}\bar{\alpha}^{\mathrm{s,f},1}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}}), \tag{2.175}$$

By applying Barbalat's lemma [71] on $\frac{\int_0^{2T^{\mathrm{s}}} \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^{\mathrm{r}}\|_Q^2 d\tau}{t}$ we conclude that, on average, the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ gets small over time, in this case proportional to $\bar{\alpha}^{\mathrm{s,f,1}}(\rho, \bar{w}^{\mathrm{o}}, T^{\mathrm{s}})$. □

DESIGN TO SATISFY THE ASSUMPTIONS

This section will detail the design of:

- the obstacle avoidance constraints, which need to satisfy Assumption 2;
- incremental Lyapunov function $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and corresponding Lipschitz constants $c_j^{\mathrm{s}}, c_j^{\mathrm{s,o}}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and c^{o}, which need to satisfy Assumptions 7 and 8;
- the terminal ingredients, which need to satisfy Assumption 9;
- the reference trajectory, which needs to satisfy Assumptions 10 and 11.

Obstacle Avoidance Constraints Design

To prove recursive feasibility, we need the obstacle avoidance constraint to satisfy Assumption 2. We can achieve this by following the same design as written in Section 2.3.2.

Incremental Stabilizability Design

Following similar arguments as in Section 2.4.3, the goal of the incremental stabilizability design is to find constant metric P^δ and state-dependent feedback $K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big)$ such that we can construct an upper bound on the incremental Lyapunov function. Since the tube dynamics in this case are defined for $\hat{\boldsymbol{x}}$ instead of $\boldsymbol{x}$, we would like to find the upper bound on $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ instead of $V^\delta(\boldsymbol{x}, \boldsymbol{z})$ in the following form:

$$\frac{d}{dt} V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq 2\sqrt{V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})}\big(-\rho\sqrt{V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})} + \bar{w}^{\mathrm{o}}\big) = -2\rho V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) + 2\sqrt{V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})}\,\bar{w}^{\mathrm{o}}, \quad (2.176)$$

with incremental Lyapunov function $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ defined as

$$V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) \coloneqq (\hat{\boldsymbol{x}} - \boldsymbol{z})^\top P^\delta(\hat{\boldsymbol{x}} - \boldsymbol{z}), \quad (2.177)$$

geodesic $\boldsymbol{\gamma}^{\mathrm{x}}(s)$ defined in (2.130), and control law $\kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})$ defined as

$$\kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v}) = \boldsymbol{\gamma}^{\mathrm{u}}(1) = \boldsymbol{v} + \int_0^1 K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big)(\hat{\boldsymbol{x}} - \boldsymbol{z})\,ds, \quad (2.178)$$

such that (2.148e) is a valid equation for the tube dynamics. To this end, let us first define the closed-loop observer dynamics:

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}} &= f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}}) \\ &= f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L\big(\boldsymbol{y} - C(\boldsymbol{x} + \boldsymbol{x} - \hat{\boldsymbol{x}})\big) \\ &= f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big), \end{aligned} \quad (2.179)$$

with $\boldsymbol{\eta}$ defined in (2.146).

Using the following shorthand notations and leveraging (2.179):

$$\begin{aligned}\dot{\boldsymbol{\delta}} &:= \dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{z}} && (2.180)\\ &= f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) - \big(f(\boldsymbol{z}, \boldsymbol{v}) - E\boldsymbol{w}^{\mathrm{b}}\big)\\ &= f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - f(\boldsymbol{z}, \boldsymbol{v}) + L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big), && (2.181)\\ A^{\mathrm{cl}} &:= A(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) + B(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}) K^{\delta}(\boldsymbol{\gamma}^{\mathrm{x}}), && (2.182)\end{aligned}$$

we can upper-bound the time-derivative of $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ as

$$\begin{aligned}\frac{d}{dt} V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) =& \dot{\boldsymbol{\delta}}^{\top} P^{\delta} \boldsymbol{\delta} + \boldsymbol{\delta}^{\top} P^{\delta} \dot{\boldsymbol{\delta}} && (2.183a)\\ =& 2\boldsymbol{\delta}^{\top} P^{\delta} \dot{\boldsymbol{\delta}} && (2.183b)\\ =& 2\boldsymbol{\delta}^{\top} P^{\delta} \Big(f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - f(\boldsymbol{z}, \boldsymbol{v}) + L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) \Big) && (2.183c)\\ =& 2\boldsymbol{\delta}^{\top} P^{\delta} \Big(f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - f(\boldsymbol{z}, \boldsymbol{v}) \Big) + 2\boldsymbol{\delta}^{\top} P^{\delta} L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) && (2.183d)\\ \overset{(a)}{\leq}& 2\boldsymbol{\delta}^{\top} P^{\delta} A^{\mathrm{cl}} \boldsymbol{\delta} + 2\boldsymbol{\delta}^{\top} P^{\delta} L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) && (2.183e)\\ \overset{(b)}{\leq}& -2\rho V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) + 2\boldsymbol{\delta}^{\top} P^{\delta} L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) && (2.183f)\\ =& -2\rho V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) + 2\boldsymbol{\delta}^{\top} P^{\delta \frac{1}{2}} P^{\delta \frac{1}{2}} L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big) && (2.183g)\\ \leq& -2\rho V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) + 2\|\boldsymbol{\delta}\|_{P^{\delta}} \|L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\|_{P^{\delta}} && (2.183h)\\ =& -2\rho V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z}) + 2\sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} \|L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\|_{P^{\delta}} && (2.183i)\end{aligned}$$

where (a) is obtained by knowing that there exists an $s \in [0,1]$ such that this expression holds by the MVT and (b) by contraction LMI (2.139c). Applying the sqrt operator to the last expression yields

$$\frac{d}{dt} \sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} \leq -\rho \sqrt{V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})} + \|L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\|_{P^{\delta}}. \tag{2.184}$$

Comparing this equation to (2.148e) indicates that we need to upper-bound $\|L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\|_{P^{\delta}}$ for all $(\hat{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{\eta}) \in \mathscr{X}^2 \times \mathscr{H}$ to obtain $\bar{w}^{\mathrm{o}}$ should upper-bound $\|L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\|_{P^{\delta}}$. We design L with bounded eigenvalues and we know that $\boldsymbol{\eta}$ is bounded since $\boldsymbol{\eta} \in \mathscr{H}$ with compact $\mathscr{H}$. This leaves us to show that $\boldsymbol{x} - \hat{\boldsymbol{x}}$ is bounded for all $(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \mathscr{X}^2$.

A sufficient condition to show boundedness of $\boldsymbol{x} - \hat{\boldsymbol{x}}$ is to enforce that sublevel set of $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ (2.160) is RPI. Similar to RPI LMI (2.139d) this gives the following LMI:

$$\begin{bmatrix} X^{\delta}\big(A(\boldsymbol{\zeta}) - LC\big)^{\top} + \big(A(\boldsymbol{\zeta}) - LC\big)X^{\delta} + \lambda^{\epsilon} X^{\delta} & E\boldsymbol{w}^0 - LF\boldsymbol{\eta} \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^{\top} & -\lambda^{\epsilon}\epsilon^2 \end{bmatrix} \leq 0, \tag{2.185}$$

with multiplier $\lambda^{\epsilon} \geq 0$ and for $\boldsymbol{w}^0 \in \mathrm{vert}(\mathscr{W})$ and $\boldsymbol{\eta} \in \mathrm{vert}(\mathscr{H})$. The proof of this so-called ϵ-RPI LMI is provided in Appendix A.7.

Remark 18. *Since LMI (2.185) enforces (2.160) to be RPI, the presented results below are only valid if (2.160) is satisfied at $t = 0$, i.e., $\sqrt{V^{\delta}(\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0)} \leq \epsilon$.*

Given metric P^δ satisfying this LMI, we can write the upper bound on $\|L\big(F\boldsymbol{\eta}+C(\boldsymbol{x}-\hat{\boldsymbol{x}})\big)\|_{P^\delta}$ as

$$\begin{aligned}
\|L\big(F\boldsymbol{\eta}+C(\boldsymbol{x}-\hat{\boldsymbol{x}})\big)\|_{P^\delta} &= \|P^{\delta\frac{1}{2}}L\big(F\boldsymbol{\eta}+C(\boldsymbol{x}-\hat{\boldsymbol{x}})\big)\| && \text{(2.186a)}\\
&= \|P^{\delta\frac{1}{2}}LF\boldsymbol{\eta}+P^{\delta\frac{1}{2}}LC(\boldsymbol{x}-\hat{\boldsymbol{x}})\| && \text{(2.186b)}\\
&\leq \|P^{\delta\frac{1}{2}}LF\boldsymbol{\eta}\|+\|P^{\delta\frac{1}{2}}LC(\boldsymbol{x}-\hat{\boldsymbol{x}})\| && \text{(2.186c)}\\
&= \|P^{\delta\frac{1}{2}}LF\boldsymbol{\eta}\|+\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}P^{\delta\frac{1}{2}}(\boldsymbol{x}-\hat{\boldsymbol{x}})\| && \text{(2.186d)}\\
&\leq \|P^{\delta\frac{1}{2}}LF\boldsymbol{\eta}\|+\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|\|P^{\delta\frac{1}{2}}(\boldsymbol{x}-\hat{\boldsymbol{x}})\| && \text{(2.186e)}\\
&= \|LF\boldsymbol{\eta}\|_{P^\delta}+\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_{P^\delta} && \text{(2.186f)}\\
&\leq \|LF\boldsymbol{\eta}\|_{P^\delta}+\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|\epsilon, && \text{(2.186g)}
\end{aligned}$$

with $\boldsymbol{\eta}\in\mathscr{H}$. Since $\mathscr{H}$ is a convex polytopic set, evaluating this expression at the vertices of $\mathscr{H}$ suffices. Thus, $\bar{w}^{\mathrm{o}}$ is given by

$$\bar{w}^{\mathrm{o}} := \max_{\boldsymbol{\eta}\in\mathrm{vert}(\mathscr{H})} \|LF\boldsymbol{\eta}\|_{P^\delta}+\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|\epsilon. \tag{2.187}$$

LMI (2.185) can also be used to optimize the shape of the tube using an RPI LMI for the controller error similar to (2.138). Let us call this LMI the δ-RPI LMI. It is given by

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^\delta+B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})+\big(A(\boldsymbol{\zeta})X^\delta+B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\big)^\top+\lambda^\delta X^\delta & LCX^\delta & LF\boldsymbol{\eta} \\ (LCX^\delta)^\top & -\lambda^{\delta,\epsilon}X^\delta & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^\delta\delta^2 \end{bmatrix} \preceq 0, \tag{2.188}$$

with $\boldsymbol{\zeta}\in\mathscr{Z}$, $\boldsymbol{\eta}\in\mathrm{vert}(\mathscr{H})$ and multipliers $\lambda^\delta,\lambda^{\delta,\epsilon}\geq 0$. Similar to the design in Section 2.4.3, we can set $\delta=1$ without loss of generality. Note that ϵ in LMI (2.185) does not necessarily equal 1. Instead, it should hold that $\lambda^\delta\delta^2\geq\lambda^{\delta,\epsilon}\epsilon^2$ for (2.188) to be satisfied. Since this LMI has a linear dependency on $\boldsymbol{\eta}$ and the set $\mathscr{H}$ is convex, it suffices to check (2.188) at the vertices of $\mathscr{H}$. Appendix A.8 provides the proof of (2.188).

Compared to the RMPC case, we want to solve the same SDP but with replacing RPI LMI (2.138) by the combination of ϵ-RPI LMI (2.185) and δ-RPI LMI (2.188) and weighting decision variables c^{s}_j, $j\in\mathbb{N}_{[2n^{\mathrm{u}}+1,n^{\mathrm{s}}]}$ and c^{o} differently in the cost function according

to the additional state constraints tightening proportional to ϵ:

$$\min_{\substack{X^\delta, Y^\delta(\boldsymbol{x}) \\ c_j^{\mathrm{s}2}, c^{\mathrm{o}2}}} \quad (1+\epsilon^2) c^{\mathrm{c,o}} c^{\mathrm{o}2} + \sum_{j=1}^{2n^{\mathrm{u}}} c_j^{\mathrm{c,s}} c_j^{\mathrm{s}2} + (1+\epsilon^2) \sum_{j=2n^{\mathrm{u}}+1}^{n^{\mathrm{s}}} c_j^{\mathrm{c,s}} c_j^{\mathrm{s}2}, \tag{2.189a}$$

$$\text{s.t. } X^\delta \succeq 0, \tag{2.189b}$$

$$A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + 2\rho X^\delta \preceq 0, \tag{2.189c}$$

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + \lambda^\delta X^\delta & LCX^\delta & LF\boldsymbol{\eta} \\ (LCX^\delta)^\top & -\lambda^{\delta,\epsilon} X^\delta & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2 - \lambda^\delta \delta^2 \end{bmatrix} \preceq 0, \tag{2.189d}$$

$$\begin{bmatrix} X^\delta \left(A(\boldsymbol{\zeta}) - LC\right)^\top + \left(A(\boldsymbol{\zeta}) - LC\right)X^\delta + \lambda^\epsilon X^\delta & E\boldsymbol{w}^0 - LF\boldsymbol{\eta} \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top & -\lambda^\epsilon \epsilon^2 \end{bmatrix} \preceq 0, \tag{2.189e}$$

$$\begin{bmatrix} c_j^{\mathrm{s}2} & L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix} \\ \left(L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix}\right)^\top & X^\delta \end{bmatrix} \succeq 0, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{2.189f}$$

$$\begin{bmatrix} c^{\mathrm{o}2} I^{n^{\mathrm{p}}} & MX^\delta \\ (MX^\delta)^\top & X^\delta \end{bmatrix} \succeq 0, \tag{2.189g}$$

$$\boldsymbol{\zeta} \in \mathcal{Z}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\mathcal{W}), \quad \boldsymbol{\eta} \in \mathrm{vert}(\mathcal{H}),$$

contraction rate $\rho > 0$ and multipliers $\lambda^\delta, \lambda^{\delta,\epsilon}, \lambda^\epsilon \geq 0$ that can be computed using bisection over this SDP, user-chosen observer gain matrix $L \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{y}}}$, and sublevel sets $\delta, \epsilon > 0$. Similar to the design in Section 2.4.3, we set $\delta = 1$ and we need to make sure $\lambda^\delta \delta^2 \geq \lambda^{\delta,\epsilon}\epsilon^2$ holds. This SDP is semi-infinite, so we need to evaluate the expressions at grid points and vertices, in this case in $\mathcal{Z}$, $\mathcal{W}$, and $\mathcal{H}$, to account for both the effect of disturbances and measurement noise.

Remark 19. *To reduce computation time, we split LMI (2.189d) following the same procedure as outlined in Remark 16:*

$$\begin{bmatrix} 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & LF\boldsymbol{\eta} \\ 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top & \mathbf{0} & 0 \end{bmatrix} \preceq \bar{H}, \quad \boldsymbol{\eta} \in \mathrm{vert}(\mathcal{H}), \tag{2.190}$$

$$\bar{H} + \begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \left(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\right)^\top + \lambda^\delta X^\delta & LCX^\delta & \mathbf{0} \\ (LCX^\delta)^\top & -\lambda^{\delta,\epsilon} X^\delta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2 - \lambda^\delta \delta^2 \end{bmatrix} \preceq 0,$$

$$\boldsymbol{\zeta} \in \mathcal{Z}. \tag{2.191}$$

Similarly, we split LMI (2.189e) as follows:

$$\begin{bmatrix} 0^{n^{\mathrm{x}}} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top & 0 \end{bmatrix} \preceq \bar{\mathscr{W}}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\mathscr{W}), \tag{2.192}$$

$$\begin{bmatrix} 0^{n^{\mathrm{x}}} & -LF\boldsymbol{\eta} \\ (-LF\boldsymbol{\eta})^\top & 0 \end{bmatrix} \preceq \bar{H}_1, \quad \boldsymbol{\eta} \in \mathrm{vert}(\mathscr{H}), \tag{2.193}$$

$$\bar{\mathscr{W}} + \bar{H}_1 + \begin{bmatrix} X^\delta \big(A(\boldsymbol{\zeta}) - LC\big)^\top + \big(A(\boldsymbol{\zeta}) - LC\big)X^\delta + \lambda^\epsilon X^\delta & \mathbf{0} \\ \mathbf{0} & -\lambda^\epsilon \epsilon^2 \end{bmatrix} \preceq 0, \quad \boldsymbol{\zeta} \in \mathscr{Z}. \tag{2.194}$$

Remark 20. *An alternative objective to (2.189a) is*

$$\min_{\substack{X^\delta, Y^\delta(\boldsymbol{x}) \\ c_j^{\mathrm{s}2}, c^{\mathrm{o}2}, \epsilon^2}} \; c^{\mathrm{c,o}} c^{\mathrm{o}2} + \sum_{j=1}^{n^{\mathrm{s}}} c_j^{\mathrm{c,s}} c_j^{\mathrm{s}2} + c^\epsilon \epsilon^2, \tag{2.195}$$

in which ϵ^2 is an additional decision variable and c^ϵ a high penalty term. Minimizing this objective results in a lower bound on ϵ that renders (2.189) feasible. Therefore, it might give less conservative constraint tightening compared to the case where (2.189) is optimized with a user-chosen value for ϵ.

Remark 21. *Note that (2.189) is bilinear in X^δ and L. Hence, a natural solution is to iterate between solving X^δ and L. For a given matrix $P^\delta = X^{\delta^{-1}}$, we use the following SDP to optimize L:*

$$\min_{\substack{\delta^2, \epsilon^2, l^2, \\ \bar{H}, \bar{H}_1}} \; \lambda^{\delta,\epsilon}\epsilon^2 + \lambda^\delta \delta^2 + c^l l^2, \tag{2.196a}$$

$$\text{s.t. (2.189d)}, \tag{2.196b}$$

$$\text{(2.189e)}, \tag{2.196c}$$

$$\begin{bmatrix} P^\delta & P^\delta LC \\ (P^\delta LC)^\top & l^2 P^\delta \end{bmatrix} \succeq 0, \tag{2.196d}$$

$$\boldsymbol{\zeta} \in \mathscr{Z}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\mathscr{W}), \quad \boldsymbol{\eta} \in \mathrm{vert}(\mathscr{H}),$$

with additional decision variable l such that

$$\| P^{\delta\frac{1}{2}} LCP^{\delta^{-\frac{1}{2}}} \| \leq l, \tag{2.197}$$

enforced by LMI (2.196d) as proven in Appendix A.9, and corresponding penalty term c^l. By minimizing l, we minimize the upper bound on $\| P^{\delta\frac{1}{2}} LCP^{\delta^{-\frac{1}{2}}} \|$, thereby also reducing conservatism in the computation of $\bar{w}^{\mathrm{o}}$ given in (2.187).

In conclusion, by successfully solving (2.189), we can find matrices P^δ and $K^\delta(\boldsymbol{x})$ such that incremental Lyapunov function $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ is given by (2.177) and satisfies (2.151a) and (2.152). Furthermore, properties (2.151b), (2.151c), and (2.151d) are also satisfied by the design of contraction LMI (2.189c), RPI LMIs (2.189d) and (2.189e),

and Lipschitz LMIs (2.189f), and (2.189g). Therefore, we can conclude that this design satisfies Assumption 7. Furthermore, Assumption 8 is directly satisfied by the design of ϵ-RPI LMI (2.185).

Terminal Ingredients Design

Similar to the terminal ingredients design presented in Section 2.4.3, we need to design a suitable terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})$ such that Assumption 9 is satisfied. In this case, we define the terminal control law as

$$\kappa^{\mathrm{f}}(\hat{\boldsymbol{x}}, \boldsymbol{r}) = \boldsymbol{u}^{\mathrm{r}} + \kappa^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{x}^{\mathrm{r}}) = \boldsymbol{u}^{\mathrm{r}} + \int_0^1 K^{\delta}\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big) ds \, (\hat{\boldsymbol{x}} - \boldsymbol{x}^{\mathrm{r}}). \tag{2.198}$$

Furthermore, matrix P needs to be defined such that descent bound (2.162) is satisfied:

$$\frac{d}{dt}\big((\hat{\boldsymbol{x}} - \boldsymbol{z})^{\top} P(\hat{\boldsymbol{x}} - \boldsymbol{z})\big) \leq -\mathscr{J}^{\mathrm{s}}(\boldsymbol{z}, \boldsymbol{v}, \boldsymbol{r}), \tag{2.199}$$

in which $\dot{\hat{\boldsymbol{x}}} := f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}})$ and $\dot{\boldsymbol{z}} = f(\boldsymbol{z}, \boldsymbol{v}) + E\boldsymbol{w}^{\mathrm{b}}$. Again, we find P with the lowest eigenvalues by solving SDP (2.145).

Thus, we can conclude that Assumption 9 is satisfied.

Reference Trajectory Design

Similar to the reference trajectory design in Section 2.4.3, Assumptions 10 and 11 are trivially satisfied if the system and obstacle avoidance constraints used for generating the reference trajectory are constructed according to (2.169) and (2.170) using tightening constants $c_j^{\mathrm{s}}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and c^{o} from SDP (2.189), respectively, and α satisfying the lower bound computed in (A.47). Correspondingly, terminal set (2.166) is suitably designed around the reference trajectory to ensure closed-loop system and obstacle avoidance constraints satisfaction at all times.

2.5.4. CONCLUDING REMARKS

In conclusion, the design of the obstacle avoidance constraints, incremental Lyapunov function, terminal ingredients, and reference trajectory as presented above results in the satisfaction of Assumptions 2, 7, 8, 9, 10, and 11. These assumptions enable the recursive feasibility and trajectory tracking proofs, thereby proving Theorem 3. Thus, ROMPC scheme (2.148) is an effective tool to provide safety guarantees for tracking dynamically feasible reference trajectories with a mobile robot described by disturbed dynamics (2.39) and noisy output measurements (2.146).

The scheme is based on the idea that both the impact of disturbances and measurement noise can be upper-bounded by the combination of an s-sublevel set of incremental Lyapunov function $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and an ϵ-sublevel set of incremental Lyapunov function $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$. This upper bound starts from ϵ to account for the observer error and grows using the dynamics of s, which results in a larger tube size than the one constructed in Section 2.4. If we use this increased tube size to tighten the system and obstacle avoidance constraints imposed on the trajectory optimization starting from estimated state $\hat{\boldsymbol{x}}$, we can guarantee that the closed-loop system $(\boldsymbol{x}, \boldsymbol{u})$ always satisfies the actual system

and obstacle avoidance constraints. In this work, both Lyapunov functions use the same metric. In general, one could consider using different metrics to quantify the impact of disturbances and measurement noise [64].

Recursive feasibility is proven by constructing a minimum positive invariant terminal set based on $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$. This expression is similar to the one found in Section 2.4.3 with the inclusion of the impact of measurement noise. Furthermore, similar to Section 2.4.3, trajectory tracking is proven by showing that terminal cost matrix P can be computed using the weighting matrix P^{δ} used in $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$ and $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$. The upper bound on the tracking error is less strict than the one derived in Section 2.4.3, which is caused by the impact of measurement noise $\boldsymbol{\eta}$.

Note that the designs presented in Section 2.4 and 2.5 assume known bounded sets $\mathscr{W}$ and $\mathscr{H}$. In practice, these bounds are often unknown. Therefore, Chapter 4 presents a pipeline for ROMPC synthesis that includes a module to quantify these uncertainty sets for general nonlinear systems.

3

EMBEDDED HIERARCHICAL MPC FOR AUTONOMOUS NAVIGATION

In Chapter 2, we established the theoretical foundations of NMPC for mobile robot navigation. Section 2.3 specifically examined an NMPC scheme designed for trajectory tracking with collision avoidance and recursive feasibility guarantees. However, implementing such a scheme on the embedded computer of an autonomous robot poses significant computational challenges, particularly due to the complexity of solving the optimization problem and generating obstacle avoidance constraints in real time.

To overcome these limitations, this chapter introduces a novel HMPC framework composed of a planning layer and a tracking layer. The planner operates at a slower rate with a long prediction horizon to generate feasible trajectories, while the tracker runs at a higher frequency to ensure accurate trajectory tracking. The proposed approach guarantees collision avoidance and recursive feasibility. Its effectiveness is demonstrated through simulations and real-world experiments using a quadrotor navigating a complex static environment. Efficient implementation on the quadrotor's embedded computer enables real-time operation. Compared to a state-of-the-art SMPC, the hierarchical scheme achieves a five-fold increase in planning horizon, resulting in significantly improved performance.

This chapter is based on and taken in parts literally from:

D. Benders, J. Köhler, T. Niesten, R. Babuška, J. Alonso-Mora and L. Ferranti, "Embedded Hierarchical MPC for Autonomous Navigation", in IEEE Transactions on Robotics, vol. 41, pp. 3556-3574, 2025, doi: 10.1109/TRO.2025.3567529. ©2025 IEEE.

Statement of contributions: DB performed the research and wrote the paper, TN provided code and platform development support, and JK, RB, JAM, and LF supervised the research and provided feedback on the paper.

The open-source implementation of this work is available at `https://github.com/dbenders1/hmpc`.
The video accompanying this work can be found at `https://youtu.be/0RnrKk6830I`.

3.1. INTRODUCTION

Autonomous mobile robots play an increasingly important role in our society [2]. The application domains are widespread, including self-driving cars [23] and environment exploration in search and rescue operations [8]. To successfully perform its task in such applications, a robot typically needs to navigate through a complex environment and reach a goal [88], [89]. This requires solving the motion planning and control problem: the robot needs to plan a smooth, collision-free, and dynamically feasible trajectory to avoid unnecessary braking and remain safe.

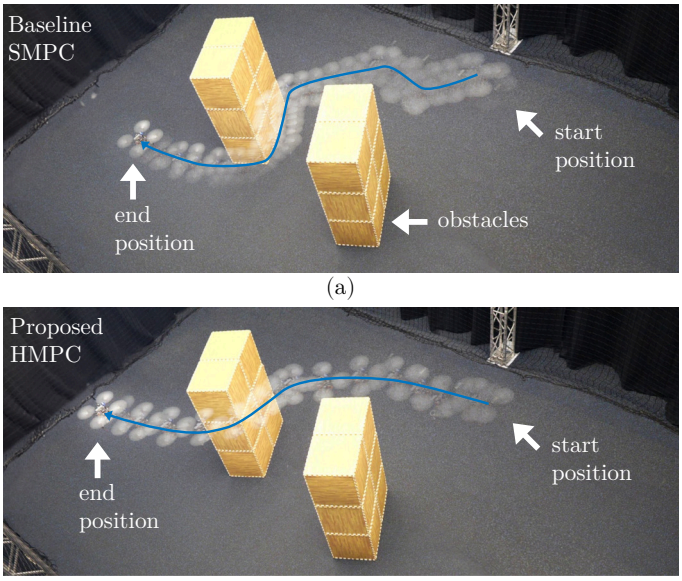


Figure 3.1: The closed-loop trajectory of (a) the baseline SMPC scheme and (b) the proposed HMPC scheme. Whereas SMPC solves the planning and tracking tasks in a single-layer formulation, HMPC decouples the tasks using a PMPC and a TMPC that use the same nonlinear model but run at different frequencies. SMPC uses the same nonlinear model as PMPC and TMPC and is sampled at the same rate as TMPC to ensure reliable operation. Both schemes fly from start to goal position and avoid the obstacles. Compared to SMPC, HMPC has a significantly longer planning horizon, given the limited computational resources on the onboard computer, and does not have to balance planning and tracking tasks. Therefore, HMPC reaches the goal faster, is less sensitive to model mismatch, and maintains altitude better than SMPC.

A natural way to solve this problem is to leverage MPC [22]. MPC is an effective method that, based on a nonlinear system model and an online observed environment, can optimize for a smooth, dynamically feasible, and collision-free trajectory. Using MPC, one can incorporate model, actuator, system, and obstacle avoidance constraints in a straightforward way, which, if constructed properly, provides safety guarantees of the closed-loop system [90]–[92].

A common approach is to use an *SMPC* scheme that executes trajectory planning and tracking as a single optimization problem [31], [69], [90]–[94]. However, the main drawback of this approach is the time to solve the problem, which increases with in-

creasing prediction horizon (for long-term predictions) and decreasing sampling times (for fast feedback), meaning that the user has to make a trade-off between long-term planning and accurate tracking. This trade-off leads to suboptimal solutions, especially on mobile robots that carry embedded hardware with limited computing power.

To overcome this problem, recent works consider decoupling the planning and tracking tasks in the form of a *HMPC* scheme, where a PMPC computes a long-term trajectory, which is tracked by a fast TMPC [32]–[34], [95]–[97]. In HMPC, the PMPC must generate a dynamically feasible trajectory that the TMPC is able to track. Although necessary for reliable operation, some of the mentioned works do not consider the co-design of PMPC and TMPC, i.e., designing the PMPC based on the tracking capabilities of the TMPC. Furthermore, to ensure runtime feasibility, these works either consider linearizing the nonlinear model or optimizing a plan from a pre-defined set of motion primitives, thereby limiting the maneuverability of the nonlinear system.

To address both problems, we propose a novel HMPC framework. In this framework, PMPC and TMPC use the same nonlinear robot model but are sampled at different rates. This allows planning a dynamically feasible trajectory far ahead and tracking it accurately; see Figure 3.1. The TMPC and PMPC are co-designed so that we can formally prove collision avoidance and recursive feasibility for the overall HMPC framework in static environments. Furthermore, we show the runtime feasibility of the framework on embedded hardware.

3.1.1. RELATED WORK

The problem of planning and tracking a trajectory for a mobile robot has been addressed in the literature using different methods. Popular planning methods include reactive [17], [98], sampling-based [18], optimization-based [19] or a combination of sampling and optimization-based [99] methods. The planner output can take various forms of plans, including paths [100], splines [101], and time-parameterized kinodynamic trajectories [19]. A general overview of motion planning methods can be found in [23]–[25].

Depending on the type of plan, various tracking methods might be preferred. An overview of trajectory tracking methods on road and aerial vehicles is given in [26] and [27], respectively. Popular methods to design tracking controllers are proportional-integral-derivative [20], linear-quadratic regulator [21], sum-of-squares programming [102] and MPC [96].

Most of the methods above focus on either the planning or the tracking task and do not consider the tracker capabilities in the planner. Consequently, they cannot guarantee the satisfaction of input and state constraints for general nonlinear robotic systems. Therefore, we consider works describing the planner and tracker co-design in the remainder of this section. We first discuss *general planner-tracker schemes* that do not use MPC in both the planner and tracker. These schemes can be separated into approaches that perform planning *offline* and *online*. Afterward, we focus on existing *HMPC* approaches.

PLANNING OFFLINE

Offline planning is useful in static scenarios such as car racing, [103] quadrotor racing [104], and industrial ground robot applications [105].

In the case of racing, there is usually no co-design of planner and tracker, and the assumption is that the planner and tracker are designed suitably and that the tracker is sampled at a sufficiently high frequency that it can accurately track the plan. Recent MPC schemes with tracking guarantees can be found in [106], [107]. However, ensuring recursive feasibility requires restrictive terminal equality constraints.

In [105], the authors show the performance of a hierarchical scheme in which a trajectory is planned and the velocity is smoothed offline, and the plan is tracked with MPC online on an industrial autonomous ground vehicle.

It should be noted that a significant number of works in the offline planning field show embedded hardware results since it is easier to obtain runtime feasibility for these algorithms compared to online planning methods. However, they can only be applied to solve a pre-defined task in a known environment.

PLANNING ONLINE

Online planning is relevant in scenarios where the perceived environment information of the robot changes during runtime. The main challenges of online planning methods are ensuring runtime feasibility and constructing a plan such that the closed-loop system avoids collisions despite deviating from the plan. Three relevant approaches to quantify this deviation are *Lyapunov-based*, *Hamilton-Jacobi (HJ) reachability-based*, and *contraction-based* methods.

Common *Lyapunov-based methods* design funnels around offline-designed motion primitives [108], [109] from which the composable ones are selected during runtime [110]. The disadvantage is that the robot is limited to the set of motion primitives, which can lead to conservative plans. This problem is addressed in recent works using parameterized funnels [36] or a combination of sampling and invariant set analysis [111]. Other relevant Lyapunov-based methods include explicit reference governors [112] and CBFs [35], [113], [114].

One of the possible applications of *HJ reachability-based methods* is to derive a tracking error bound between a low-fidelity planning and high-fidelity tracking model in order to reduce planning computation time [33], [115]. However, the tracking error bound is computationally expensive and can be overly conservative. To address these problems, [116] reduces computation time by linearizing around operating points and constructs corresponding zonotopic reachable sets, and [117] reduces conservatism by switching between a non-conservative 'slow' plan and a conservative 'fast' plan.

Contraction-based methods are based on contraction theory, which concludes about the convergence of two trajectories, the planned and closed-loop trajectories, based on the evolution of two infinitesimally close trajectories [86]. The approach in [79] leverages contraction theory and sum-of-squares programming to generate a CCM controller with a corresponding funnel around the trajectory but suffers from expensive computation time, both offline and online, and a conservative plan. The method proposed in [118] combines the approaches in [79] and [115] to reduce offline computations compared to [115] and online planning time compared to [79].

HMPC

Similar to general planner-tracker methods, HMPC schemes also suffer from long planning computation times and propose similar solutions: using a simpler model or optimizing from a pre-defined set of motion primitives.

Following a similar idea to [115], the authors in [33] use a low-fidelity nonlinear PMPC model and a high-fidelity linear time-varying TMPC from [119]. Other works leverage a linear PMPC model to reduce computation time. For example, [97] shows how to efficiently use a linear mixed-integer PMPC and nonlinear TMPC to cover an area with a quadrotor in simulation.

Optimizing based on a set of motion primitives can be done via mixed-integer programming. This approach was taken in [95], demonstrating results in simulation and on a real passenger car driving on a slippery surface.

Other approaches combine a linear model and a set of motion primitives. One of these works is [96], in which the authors implement a runtime feasible HMPC scheme on small race cars by linearizing the nonlinear dynamics around the previously optimal trajectory and selecting trajectories from a pre-defined set of motion primitives in the PMPC. Another work is [34], in which simulation results are shown of a PMPC optimizing from a set of motion primitives and aggressive modes, and a TMPC being a tube-based formulation.

To avoid limited maneuverability, the authors in [32] propose a mixed-integer PMPC formulation that explicitly considers the distance to obstacles based on the aggressiveness mode and a TMPC being a cyclic horizon MPC. Quadrotor simulation results demonstrate that the method can safely fly from start to goal in a known static environment. However, the PMPC computation time is already at its limit for real-time feasibility while only using a linear model, and it was not implemented on embedded hardware.

In conclusion, designing an HMPC that computes a plan using the nonlinear robot dynamics online and accurately tracks it in real time on a real robot with limited computation power is challenging. The methods above solve this problem by reducing the model complexity or optimizing a plan using a set of pre-defined motion primitives. In contrast, our method uses the same nonlinear model in both MPC layers and guarantees tracking of the planned trajectory, similar to the idea of decoupling PMPC and TMPC while maintaining recursive feasibility presented in [39]. Leveraging recently developed MPC tracking [38] that guarantees computationally efficient tracking performance and recursive feasibility, we propose an HMPC framework for a mobile robot navigating in a static environment. The framework ensures collision avoidance and recursive feasibility, thereby taking an important step toward closing the gap between real-world mobile robot deployment and theoretical MPC feasibility and tracking guarantees.

3.1.2. CONTRIBUTIONS

The main contribution of this work is a novel HMPC framework to enable embedded motion planning and tracking on a mobile robot. In particular, we propose:

(A) a combined design methodology for PMPC and TMPC leveraging the same nonlinear model, in which the PMPC constructs a dynamically feasible trajectory that the TMPC is guaranteed to track by tightening the constraints in accordance with the

offline terminal ingredients design of the TMPC. The proposed PMPC can solve the long-horizon planning problem completely independent of the TMPC over a longer, user-chosen time interval;

(B) theoretical guarantees that both PMPC and TMPC are recursively feasible and the HMPC framework avoids static obstacles in complex environments at all times;

(C) an efficient way to implement the convex free space decomposition scheme presented in [67] for an MPC-based planner.

While the framework can be used on any mobile robotic platform, we implemented it on a quadrotor's NVIDIA Jetson Xavier NX embedded computer. Simulations and experiments validate the theoretical guarantees and practicality of the proposed HMPC framework.

The code base, written in C++ as a Robot Operating System (ROS) package, implements different MPC formulations (i.e., SMPC, TMPC, and PMPC) and solves these problems leveraging the ForcesPro nonlinear programming (NLP) solver [120], [121]. The only change in code for the different simulations and experiments is the interface, which allows for easier debugging and enhances sim-to-real transfer. The package, together with simulation and analysis code, is contained in a Docker environment, in line with the recommendation in [122], and is available at `https://github.com/dbenders1/hmpc`.

3.1.3. OUTLINE

The rest of the chapter is organized as follows: Section 3.2 describes the considered robot and environment. Furthermore, it formulates the general trajectory planning and tracking problem and explains how the hierarchical framework follows from this scheme. Section 3.3 presents the TMPC design, including terminal ingredients (terminal cost and set) design with corresponding tracking properties. Section 3.4 provides the PMPC design to construct a reference trajectory and the corresponding obstacle avoidance constraints for the TMPC to ensure recursive feasibility and obstacle avoidance of the closed-loop system. The overall HMPC framework is described in Section 3.5, including concise pseudo-algorithms, the theoretical analysis of the framework and its components with proofs given in Appendix B.1 and B.2, and a summary of the theoretical properties that the framework provides. Section 3.6 provides the implementation details and experimental results. Finally, Section 3.7 concludes the chapter.

3.2. PROBLEM FORMULATION

3.2.1. ROBOT AND CONSTRAINTS DESCRIPTION

The mobile robot system is described by nonlinear dynamics (2.1) subject to system constraints (2.2) and obstacle avoidance constraints (2.3), as described in Section 2.2. Section 3.4.2 provides details on constructing the obstacle avoidance constraints.

3.2.2. TRAJECTORY PLANNING AND TRACKING

The problem we aim to solve is to generate a dynamically feasible and collision-free trajectory for a mobile robot navigating in a static environment while considering the lim-

ited computation resources of the robot's embedded computer. To solve this problem, we rely on MPC. The SMPC scheme to solve this problem is given by:

$$\begin{aligned}
\min_{\boldsymbol{x}_{\cdot|t},\boldsymbol{u}_{\cdot|t}} \quad & \mathscr{J}(\boldsymbol{x}_{\cdot|t},\boldsymbol{u}_{\cdot|t},\boldsymbol{p}^{\mathrm{g}}), && && \text{(3.1a)}\\
\text{s.t. } & \boldsymbol{x}_{0|t} = \boldsymbol{x}_t, && && \text{(3.1b)}\\
& \dot{\boldsymbol{x}}_{\tau|t} = f(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}), && \tau \in [0,T], && \text{(3.1c)}\\
& (\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}) \in \mathscr{Z}, && \tau \in [0,T], && \text{(3.1d)}\\
& \boldsymbol{p}_{\tau|t} \in \mathscr{F}_{\tau|t}, && \tau \in [0,T], && \text{(3.1e)}\\
& \boldsymbol{x}_{T|t} \in \mathscr{X}^{\mathrm{f}}, && && \text{(3.1f)}
\end{aligned}$$

which optimizes a trajectory by minimizing cost function $\mathscr{J}(\boldsymbol{x}_{\cdot|t},\boldsymbol{u}_{\cdot|t},\boldsymbol{p}^{\mathrm{g}})$ (3.1a), while satisfying system constraints (3.1d) and obstacle avoidance constraints (3.1e), in order to steer the robot with dynamics (3.1c) from an initial state (3.1b) to a terminal set (3.1f) in the direction of goal position $\boldsymbol{p}^{\mathrm{g}}$.

As discussed in the introduction, such an SMPC is challenging to implement on embedded hardware. Hence, our goal is to develop PMPC and TMPC formulations that, when combined, reach goal $\boldsymbol{p}^{\mathrm{g}}$ while satisfying constraints (3.1b)-(3.1f). PMPC and TMPC are combined so that (a) the TMPC is guaranteed to track the trajectory provided by the PMPC and (b) the overall scheme avoids collisions and is recursively feasible. In this setting, the PMPC minimizes a user-chosen planning cost, while the TMPC minimizes a tracking cost related to the reference generated by the PMPC. Both MPC formulations use the actual nonlinear dynamics (2.1) for the prediction.

To achieve this goal, the main TMPC design challenge is computing terminal ingredients, including terminal cost in (3.1a) and terminal set (3.1f), such that the closed-loop system tracks the reference computed by the PMPC.

The main PMPC design challenges are (a) constructing initial state constraint (3.1b), such that the communicated trajectory to the TMPC is continuous and dynamically feasible, (b) tightening the system and obstacle avoidance constraints, such that the robot does not collide with obstacles given the offline computed terminal set of the TMPC, and (c) choosing a proper terminal set ensuring recursive feasibility of the PMPC.

To summarize the co-design of PMPC and TMPC, Figure 3.2 provides an overview of the proposed HMPC framework. The framework is detailed in Section 3.5.

3.3. TMPC DESIGN

This section presents the TMPC formulation in Section 3.3.1. Section 3.3.2 describes the offline design of suitable terminal ingredients to prove recursive feasibility and convergence to the reference trajectory. Finally, Section 3.3.3 gives some properties that the reference trajectory and obstacle avoidance constraints generated by the PMPC should satisfy and formalizes the corresponding TMPC tracking properties in Proposition 1.

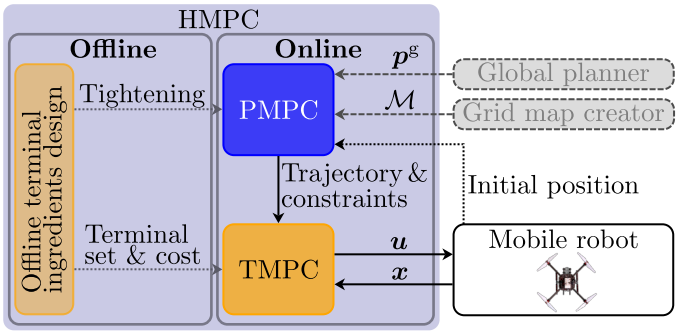


Figure 3.2: HMPC framework overview. In the offline phase, the terminal ingredients of the TMPC, including the terminal set, cost, and tightening constants, are computed. During online operation, the PMPC plans a trajectory based on goal position $\boldsymbol{p}^{\mathrm{g}}$ and grid map $\mathscr{M}$, and the TMPC tracks this trajectory using state feedback $\boldsymbol{x}$.

3.3.1. TMPC FORMULATION

The TMPC formulation is based on [38] with added obstacle avoidance capability:

$$\min_{\boldsymbol{x}_{\cdot|t},\boldsymbol{u}_{\cdot|t}} \; \mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}_{T^{\mathrm{t}}|t}-\boldsymbol{x}^{\mathrm{r}}_{T^{\mathrm{t}}|t}) + \int_{\tau=0}^{T^{\mathrm{t}}} \mathscr{J}^{\mathrm{s,t}}(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t},\boldsymbol{r}_{\tau|t})\, d\tau, \tag{3.2a}$$

$$\text{s.t. } \boldsymbol{x}_{0|t} = \boldsymbol{x}_t, \tag{3.2b}$$

$$\dot{\boldsymbol{x}}_{\tau|t} = f(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}), \quad \tau \in [0, T^{\mathrm{t}}], \tag{3.2c}$$

$$(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}) \in \mathscr{Z}, \quad \tau \in [0, T^{\mathrm{t}}], \tag{3.2d}$$

$$\boldsymbol{p}_{\tau|t} \in \mathscr{F}_{\tau|t}, \quad \tau \in [0, T^{\mathrm{t}}], \tag{3.2e}$$

$$(\boldsymbol{x}_{T^{\mathrm{t}}|t} - \boldsymbol{x}^{\mathrm{r}}_{T^{\mathrm{t}}|t}) \in \mathscr{X}^{\mathrm{f,t}}, \tag{3.2f}$$

with TMPC sampling time $T^{\mathrm{s,t}}$. This means that (3.2) is solved every $T^{\mathrm{s,t}}$ seconds.

The cost penalizes the error between the prediction and the reference trajectory $\boldsymbol{r} = [\boldsymbol{x}^{\mathrm{r}\top}\boldsymbol{u}^{\mathrm{r}\top}]^\top$ using stage and terminal costs:

$$\mathscr{J}^{\mathrm{s,t}}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{r}) := \|\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}\|^2_Q + \|\boldsymbol{u}-\boldsymbol{u}^{\mathrm{r}}\|^2_R, \tag{3.3}$$

$$\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}) := \|\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}\|^2_P, \tag{3.4}$$

with $Q, R, P \succ 0$. The optimal input to Problem (3.2) is denoted by $\boldsymbol{u}^*_{\tau|t}$. As a result, the control actions during each TMPC sampling interval are given by:

$$\boldsymbol{u}_{\tau+t} := \boldsymbol{u}^*_{\tau|t-T^{\mathrm{s,t}}}, \quad \tau \in [0, T^{\mathrm{s,t}}]. \tag{3.5}$$

3.3.2. OFFLINE TERMINAL INGREDIENTS DESIGN

The terminal ingredients are computed offline using the SDP (3.6) with linearized system dynamics:

$$A(\boldsymbol{z}) = \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{z}}, \quad B(\boldsymbol{z}) = \frac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{z}}. \tag{3.7}$$

$$\min_{X,Y,c_j^{s2}} \; -\log\det X + \sum_{j=1}^{n^s} c_j^c c_j^{s2}, \tag{3.6a}$$

$$\text{s.t.} \begin{bmatrix} A(\boldsymbol{z})X + B(\boldsymbol{z})Y + \big(A(\boldsymbol{z})X + B(\boldsymbol{z})Y\big)^\top & (Q^{\frac{1}{2}}X)^\top & (R^{\frac{1}{2}}Y)^\top \\ * & -I_{n^x} & 0 \\ * & 0 & -I_{n^u} \end{bmatrix} \preceq 0, \; \forall \boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} \in \mathcal{Z}, \tag{3.6b}$$

$$\begin{bmatrix} c_j^{s2} & L_j^s \begin{bmatrix} Y \\ X \end{bmatrix} \\ (L_j^s \begin{bmatrix} Y \\ X \end{bmatrix})^\top & X \end{bmatrix} \succeq 0, \; j \in \mathbb{N}_{[1,n^s]}, \; X \succeq 0. \tag{3.6c}$$

Based on the SDP solution, the terminal cost matrix P and feedback law gain K are computed using:

$$P = X^{-1}, \quad K = YP, \tag{3.8}$$

and the quadratic terminal cost is an incremental Lyapunov function [38] defined in (3.4).

The terminal set is defined as an ellipsoidal sublevel set of the incremental Lyapunov function given by:

$$\mathcal{X}^{f,t} := \{\boldsymbol{x} \in \mathbb{R}^{n^x} \mid \|\boldsymbol{x}\|_P^2 \leq \alpha^2\}, \tag{3.9}$$

with a user-chosen terminal set scaling $\alpha > 0$. $\mathcal{X}^{f,t}$ is positive invariant for control input $\boldsymbol{u} = \kappa^f(\boldsymbol{x}, \boldsymbol{r})$ with:

$$\kappa^f(\boldsymbol{x}, \boldsymbol{r}) := \boldsymbol{u}^r + K(\boldsymbol{x} - \boldsymbol{x}^r), \tag{3.10}$$

which is crucial to ensure recursive feasibility and safety.

The objective of SDP (3.6) is to find a P that gives a suitable terminal cost and set for trajectory tracking. The weights c_j^c are tuning parameters that are normalized with respect to the system constraint intervals to ensure equal tightening of each constraint $j \in \mathbb{N}_{[1,n^s]}$ using tightening constant c_j^s, which will be defined later in (3.14a), and terminal set scaling α.

LMI (3.6b) ensures that the incremental Lyapunov function (3.4) exponentially decreases when applying the feedback law (3.10) in the terminal set (3.9). LMI (3.6c) ensures that the system constraints are satisfied for all $\boldsymbol{z} \in \mathcal{Z}$, see [123] and [38] for more details.

Note that $\mathcal{Z}$ represents the continuous state-input space, and we optimize for X, Y, and c_j^s that uniformly hold for all $\boldsymbol{z} \in \mathcal{Z}$. Therefore, SDP (3.6) is semi-infinite, and we need to grid or convexify $\mathcal{Z}$ to solve it. If (3.6) is infeasible, one could leverage a more general formulation with state-dependent X and Y; see [38] for details.

Note also that, since (3.6) ensures exponential stability with linear feedback K, a feasible solution for non-holonomic systems, such as cars, only exists when enforcing a minimum velocity.

3.3.3. CLOSED-LOOP PROPERTIES

Property 4 and Property 5 formalize the required properties for the reference trajectory and obstacle avoidance constraints:

Property 4. *The reference trajectory satisfies the following construction (a) and update (b) conditions:*

(a) For $\tau \in [0, T^{\mathrm{t}}]$, it holds that:

$$\dot{\boldsymbol{x}}^{\mathrm{r}}_{\tau|t} = f(\boldsymbol{x}^{\mathrm{r}}_{\tau|t}, \boldsymbol{u}^{\mathrm{r}}_{\tau|t}), \tag{3.11a}$$

$$\boldsymbol{r}_{\tau|t} \in \tilde{\mathcal{Z}}, \tag{3.11b}$$

$$M\boldsymbol{x}^{\mathrm{r}}_{\tau|t} \in \tilde{\mathscr{F}}_{\tau|t}, \tag{3.11c}$$

with

$$\tilde{\mathcal{Z}} := \tilde{\mathcal{X}} \times \tilde{\mathcal{U}} = \left\{ (\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{u}}} \;\middle|\; \begin{array}{l} g^{\mathrm{s}}_j(\boldsymbol{x}, \boldsymbol{u}) + c^{\mathrm{s}}_j \alpha \le 0, \\ j \in \mathbb{N}_{[1, n^{\mathrm{s}}]} \end{array} \right\}, \tag{3.12}$$

and

$$\tilde{\mathscr{F}}_{\tau|t} := \left\{ \boldsymbol{p}_{\tau|t} \in \mathbb{R}^{n^{\mathrm{p}}} \;\middle|\; \begin{array}{l} g^{\mathrm{o}}_{j,\tau|t}(\boldsymbol{p}_{\tau|t}) + c^{\mathrm{o}} \alpha \le 0, \\ j \in \mathbb{N}_{[1, n^{\mathrm{o}}]} \end{array} \right\}, \tag{3.13}$$

with $\tilde{\mathcal{Z}} \subseteq \mathcal{Z}$ and $\tilde{\mathscr{F}}_{\tau|t} \subseteq \mathscr{F}_{\tau|t}$ following from tightening with $\alpha > 0$ from (3.9) and:

$$c^{\mathrm{s}}_j := \| P^{-\frac{1}{2}} [I \; K^\top] L^{\mathrm{s}}_j \|, \quad j \in \mathbb{N}_{[1, n^{\mathrm{s}}]}, \tag{3.14a}$$

$$c^{\mathrm{o}} := \| P^{-\frac{1}{2}} M^\top \|. \tag{3.14b}$$

(b) For $\tau \in [0, T^{\mathrm{s,t}}]$, it holds that:

$$\boldsymbol{x}^{\mathrm{r}}_{\tau + T^{\mathrm{s,t}}} = \boldsymbol{x}^{\mathrm{r}}_{\tau|t + T^{\mathrm{s,t}}} = \boldsymbol{x}^{\mathrm{r}}_{\tau + T^{\mathrm{s,t}}|t}. \tag{3.15}$$

Intuitively, Property 4 (a) states that the reference trajectory is dynamically feasible (3.11a), and it satisfies tightened system constraints (3.11b) and tightened obstacle avoidance constraints (3.11c). Property 4 (b) states that the reference trajectory is continuous, i.e. the reference trajectory in the time interval $[0, T^{\mathrm{s,t}}]$ of the current TMPC run is the same as the reference trajectory in the interval $[T^{\mathrm{s,t}}, 2T^{\mathrm{s,t}}]$ of the last TMPC run.

Figure 3.3 illustrates an example of a 2D Lyapunov function and corresponding ellipsoidal terminal set around the reference trajectory.

Remark 22. *The terminal set scaling α can be seen as a tuning parameter to make a trade-off between the performance of TMPC and PMPC. The smaller α, the closer PMPC can plan to the obstacle avoidance constraints, resulting in a less conservative trajectory. The bigger α, the bigger the terminal set for TMPC, making it easier for TMPC to find a solution. Although not formally proven in this paper, the size of the terminal region alpha is related to the inherent robustness of the TMPC, cf. [76]. This property ensures that closed-loop properties remain valid under sufficiently small disturbances, which is crucial for application on the real robot.*

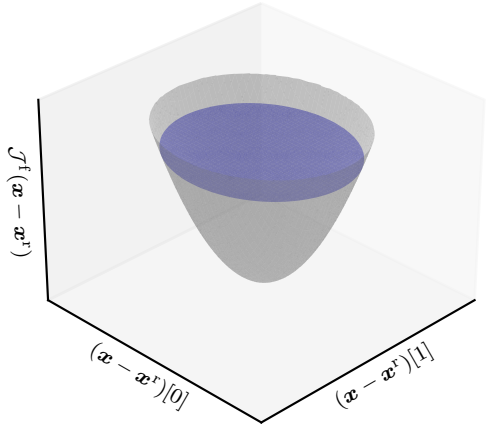


Figure 3.3: Quadratic Lyapunov function $\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}})$ for two state tracking error dimensions. The blue ellipsoid is the terminal set $\mathscr{X}^{\mathrm{f,t}}$ intersecting $\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}})$ at value α^2.

Property 5. *The obstacle avoidance constraints satisfy the following construction (a) and update (b) conditions:*

(a) They form a subset such that the robot does not collide with obstacles: $\mathscr{F}_{\tau|t} \cap \mathscr{O} \oplus \mathscr{R} = \emptyset,\ \tau \in [0, T^{\mathrm{p}}]$.

(b) They are updated in a consistent way, i.e., $\boldsymbol{p}^{\mathrm{r}}_{\tau|t} \in \bar{\mathscr{F}}_{\tau|t} \implies \boldsymbol{p}^{\mathrm{r}}_{\tau|t} \in \bar{\mathscr{F}}_{\tau - T^{\mathrm{s,p}}|t+T^{\mathrm{s,p}}},\ \tau \in [T^{\mathrm{s,p}}, T^{\mathrm{p}}]$, *with PMPC sampling time* $T^{\mathrm{s,p}}$ *and prediction horizon* T^{p}.

Intuitively, Property 5 (b) states the reference trajectory should be contained in the updated tightened obstacle-free region, given that it is contained in the current tightened obstacle-free region at the same point in time.

Section 3.4 shows that Properties 4 and 5 are ensured by the PMPC design.

Proposition 1 formalizes the properties of the TMPC terminal ingredients.

Proposition 1 (Terminal ingredients)**.** *Suppose the terminal ingredients are designed as above,* $\boldsymbol{r}$ *satisfies Property 4 and* $\mathscr{F}$ *satisfies Property 5. Then for any* $\boldsymbol{x}$ *satisfying* $(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}) \in \mathscr{X}^{\mathrm{f,t}}$:

$$\frac{d}{dt}\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}) \leq -\mathscr{J}^{\mathrm{s,t}}\big(\boldsymbol{x}, \kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r}), \boldsymbol{r}\big), \tag{3.16a}$$

$$\big(\boldsymbol{x}, \kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r})\big) \in \mathscr{Z}, \tag{3.16b}$$

$$M\boldsymbol{x} \in \mathscr{F}. \tag{3.16c}$$

Proof. (3.16a) holds given a feasible solution to SDP (3.6). The main steps of the proof,

similar to [38], are:

$$\frac{d}{dt}\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{\delta}) = 2\boldsymbol{\delta}^\top P\big(f(\boldsymbol{x},\boldsymbol{u}) - f(\boldsymbol{x}^{\mathrm{r}},\boldsymbol{u}^{\mathrm{r}})\big) \overset{\text{(a)}}{=} 2\boldsymbol{\delta}^\top P\Big(A\big(\boldsymbol{z}(s)\big) + B\big(\boldsymbol{z}(s)\big)K\Big)\boldsymbol{\delta}$$
$$\overset{\text{(b)}}{\leq} -\boldsymbol{\delta}^\top (Q + K^\top RK)\boldsymbol{\delta} \qquad = -\mathscr{J}^{\mathrm{s,t}}\big(\boldsymbol{x}, \kappa^{\mathrm{f}}(\boldsymbol{x},\boldsymbol{r}), \boldsymbol{r}\big), \tag{3.17}$$

where $\boldsymbol{\delta}=\boldsymbol{x}-\boldsymbol{x}^{\mathrm{r}}$. (a) uses the fact that $\boldsymbol{z}(s)=\begin{bmatrix}\boldsymbol{x}^{\mathrm{r}}\\ \boldsymbol{u}^{\mathrm{r}}\end{bmatrix} + s\begin{bmatrix}\boldsymbol{I}^{\mathrm{n}^x}\\ K\end{bmatrix}\boldsymbol{\delta}$ linearly interpolates between $(\boldsymbol{x},\boldsymbol{u})$ and $(\boldsymbol{x}^{\mathrm{r}},\boldsymbol{u}^{\mathrm{r}})$ for $s\in[0,1]$, and the MVT ensuring that this equality holds for some value of s. (b) follows from LMI (3.6b), see [38, Lemma 3]. Furthermore, given $L\in\mathbb{R}^{1\times n}$ satisfying $\|L\|=1$ and $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_P^2 \leq \alpha^2$ it holds that:

$$LM\boldsymbol{\delta} \leq \|LMP^{-\frac{1}{2}}P^{\frac{1}{2}}\boldsymbol{\delta}\| \leq \|LMP^{-\frac{1}{2}}\|\|P^{\frac{1}{2}}\boldsymbol{\delta}\| \leq \|MP^{-\frac{1}{2}}\|\|P^{\frac{1}{2}}\boldsymbol{\delta}\| \leq \|MP^{-\frac{1}{2}}\|\alpha, \tag{3.18}$$

where the second and third inequalities use the triangular inequality. Applying this to L_j^{o} with $\|L_j^{\mathrm{o}}\|=1$, it holds that $L_j^{\mathrm{o}}M\boldsymbol{\delta} \leq c^{\mathrm{o}}\alpha$, meaning that $L_j^{\mathrm{o}}M\boldsymbol{x} \leq L_j^{\mathrm{o}}M\boldsymbol{x}^{\mathrm{r}} + c^{\mathrm{o}}\alpha \leq l_j^{\mathrm{o}},\ j\in\mathbb{N}_{[1,n^{\mathrm{o}}]}$ holds by (3.11c), given (3.13) and (3.14b), thereby proving (3.16c). A similar proof holds for (3.16b), see [38] for more details. □

Intuitively, Proposition 1 states that, given Properties 4 and 5, the closed-loop trajectory converges to the reference trajectory and system and obstacle avoidance constraints are satisfied when applying terminal control law (3.10) in terminal set (3.9). Therefore, the terminal set is positively invariant under the terminal control law. This means that the shifted previously optimal solution appended with the terminal control law is a feasible, not necessarily optimal, candidate solution for the next TMPC run. Repeating this argument demonstrates that TMPC Problem (3.2) is recursively feasible. This is formally proven in Theorem 4 in Section 3.5.2.

3.4. PMPC DESIGN

The planner's goal is to optimize a dynamically feasible trajectory towards goal position $\boldsymbol{p}^{\mathrm{g}}$ such that the TMPC is guaranteed to track it. Section 3.4.1 presents the PMPC formulation to optimize the trajectory. The generation of the obstacle avoidance constraints is described in Section 3.4.2.

3.4.1. PMPC FORMULATION

The PMPC formulation is given by:

$$\min_{\boldsymbol{x}_{\cdot|t},\boldsymbol{u}_{\cdot|t}} \mathscr{J}^{\mathrm{f,p}}(\boldsymbol{x}_{T^{\mathrm{p}}|t},\boldsymbol{u}_{T^{\mathrm{p}}|t},\boldsymbol{p}^{\mathrm{g}}) + \int_{\tau=0}^{T^{\mathrm{p}}} \mathscr{J}^{\mathrm{s,p}}(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t},\boldsymbol{p}^{\mathrm{g}})\,d\tau, \tag{3.19a}$$
$$\text{s.t. } \boldsymbol{x}_{\tau|t} = \boldsymbol{x}^*_{\tau+T^{\mathrm{s,p}}|t-T^{\mathrm{s,p}}},\ \tau\in[0,T^{\mathrm{s,p}}], \tag{3.19b}$$
$$\dot{\boldsymbol{x}}_{\tau|t} = f(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}),\quad \tau\in[0,T^{\mathrm{p}}], \tag{3.19c}$$
$$(\boldsymbol{x}_{\tau|t},\boldsymbol{u}_{\tau|t}) \in \bar{\mathcal{Z}},\qquad \tau\in[0,T^{\mathrm{p}}], \tag{3.19d}$$
$$\boldsymbol{p}_{\tau|t} \in \bar{\mathscr{F}}_{\tau|t},\qquad \tau\in[0,T^{\mathrm{p}}], \tag{3.19e}$$
$$f(\boldsymbol{x}_{T^{\mathrm{p}}|t},\boldsymbol{u}_{T^{\mathrm{p}}|t}) = 0, \tag{3.19f}$$

with PMPC sampling $T^{\mathrm{s,p}}$, and $\tilde{\mathscr{Z}}$ and $\tilde{\mathscr{F}}_{\tau|t}$, $\tau \in [0, T^{\mathrm{p}}]$, given by (3.12) and (3.13), respectively. As mentioned, the cost is a user-chosen planning cost that can include any terms suitable for navigating in the environment (e.g., goal-oriented MPC (GO-MPC), see [88], or model predictive contouring control, see [91], [93]). Section 3.6.1 elaborates on the specific function used to generate the experimental results. The combination of initial state constraint (3.19b) and system dynamics (3.19c), is sufficient to enforce both (3.11a) in Property 4 (a) and Property (b). Furthermore, (3.19d) and (3.19e) ensure (3.11b) and (3.11c) in Property 4 (a), respectively. Finally, (3.19f) is a terminal set constraint to ensure the recursive feasibility of the PMPC. Note that (3.19b) can be efficiently implemented by using a shorter horizon $T^{\mathrm{p}} - T^{\mathrm{s,p}}$ with initial state constraint $\boldsymbol{x}_{0|t} = \boldsymbol{x}^*_{T^{\mathrm{s,p}}|t-T^{\mathrm{s,p}}}$.

3.4.2. OBSTACLE AVOIDANCE CONSTRAINTS

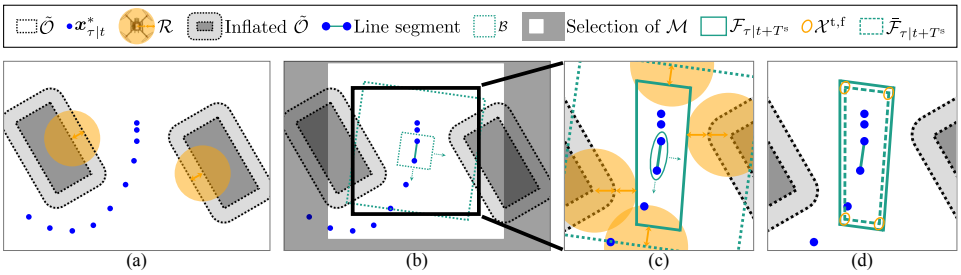


Figure 3.4: 2D visualization of map pre-processing and *I-DecompUtil*, given occupied grid cells $\tilde{\mathcal{O}}$ and the last optimized plan $\boldsymbol{x}^*_{\tau|t}$. (a) The obstacles are inflated by half of the robot radius (orange arrows). (b) To construct the obstacle avoidance constraints around a specific line segment, first, a subset of the grid map $\mathscr{M}$ is selected such that the bounding box $\mathscr{B}$ with any orientation fits in this subset. (c) The obstacle avoidance constraints $\mathscr{F}_{\tau|t+T^{\mathrm{s,p}}}$ are constructed according to the safe flight corridor method in [67] by growing an ellipsoid around the line segment, creating the tangential lines and clipping them to $\mathscr{B}$. Furthermore, $\mathscr{F}_{\tau|t+T^{\mathrm{s,p}}}$ are tightened by the other half of the robot radius, such that the robot does not collide with the obstacles if its center satisfies $\mathscr{F}_{\tau|t+T^{\mathrm{s,p}}}$. (d) The tightened obstacle avoidance constraints $\tilde{\mathscr{F}}_{\tau|t+T^{\mathrm{s,p}}}$ are constructed by tightening $\mathscr{F}_{\tau|t+T^{\mathrm{s,p}}}$ with $c^{\mathrm{o}}\alpha$ according to (3.13), visualized using terminal set $\mathscr{X}^{\mathrm{f,t}}$ in this figure. Note that (b)-(d) are repeated for all line segments.

This section describes the method to generate obstacle avoidance constraints of the form (2.3). To deal with an online observed environment with obstacles $\mathscr{O}$, we leverage grid map representation $\mathscr{M}$ in which the occupied cells $\tilde{\mathcal{O}}$ indicate the obstacle edges. Given these occupied cells, the method should generate constraints satisfying Property 5.

In particular, the method builds on *DecompUtil* as introduced in [67, Fig. 5-8]. Based on the grid map and a linear path segment, *DecompUtil* builds a convex obstacle-free region by appending the tangential line of a growing ellipsoid around the path segment at the first encountered occupied grid cell after removing all grid cells behind the previously generated tangential lines and clipping the result to bounding box $\mathscr{B}$.

In this work, the obstacle avoidance constraints are constructed following the same method, based on piecewise linear segments between optimally planned PMPC positions $\boldsymbol{p}^*_{iT^{\mathrm{s,p}}|t-T^{\mathrm{s,p}}}$ and $\boldsymbol{p}^*_{(i+1)T^{\mathrm{s,p}}|t-T^{\mathrm{s,p}}}, i \in \mathbb{N}_{[1,N-1]}, N = \frac{T^{\mathrm{p}}}{T^{\mathrm{s,p}}}$, starting with $\mathscr{B}$ around

the initial position. This approach is further referred to as *Iterative-DecompUtil* (*I-DecompUtil*) and visualized in Figure 3.4.

There are two important things to note in the figure: the *Selection of* $\mathcal{M}$ and the usage of the robot radius to both tighten the obstacle-free convex regions and inflate the obstacles.

Selection of $\mathcal{M}$ reduces the computation time by reducing the number of grid cells that need to be considered.

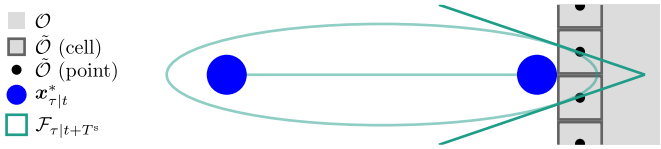


Figure 3.5: Visualization of a scenario in which the robot would crash into the obstacle if the constraints are not tightened. The obstacle is inflated, here represented by $\mathcal{O}$, with corresponding occupied grid cells $\tilde{\mathcal{O}}$ and the point representation of $\tilde{\mathcal{O}}$ in the code. While the ellipsoid does not intersect the grid points, it overlaps with the obstacle region, i.e., $\mathcal{F} \cap \mathcal{O} \neq \emptyset$.

Since (3.2) and (3.19) optimize for the geometric robot center, the distance between obstacle avoidance constraints and occupied grid cells should be at least the robot radius. Tightening the constraints by the robot radius would result in sharp corners between subsequent obstacle-free regions around sharp obstacle corners, resulting in more conservative plans. On the other hand, inflating the obstacles by the robot radius would lead to the scenario depicted in Figure 3.5, meaning that a crash may occur. To prevent both issues, half of the robot radius is used to tighten the constraints, and the other half is used to inflate the obstacles. This ensures collision avoidance, given that the robot radius is significantly larger than the grid map resolution.

The resulting obstacle-free convex regions from *I-DecompUtil* are piecewise-defined as:

$$\mathcal{F}_{\tau+iT^{\mathrm{s,p}}|t} = \mathcal{F}_{iT^{\mathrm{s,p}}|t},\ \tau \in (0, T^{\mathrm{s,p}}],\ i \in \mathbb{N}_{[0,N-1]}. \tag{3.20}$$

This means that $\boldsymbol{p}^*_{(iT^{\mathrm{s,p}},(i+1)T^{\mathrm{s,p}}]|t} \in \mathcal{F}_{iT^{\mathrm{s,p}}|t}, i \in \mathbb{N}_{[0,N-1]}$. Note that, although the regions are constructed using piecewise linear segments, the nonlinear robot trajectory is contained in these convex regions since the constraints are imposed on the TMPC and PMPC trajectories through (3.2e) and (3.19e), respectively.

To prove recursive feasibility of PMPC formulation (3.19) later in Section 3.5.2, we consider the following assumption regarding the grid map $\mathcal{M}$ and the PMPC sampling time $T^{\mathrm{s,p}}$:

Assumption 12. *The grid map* $\mathcal{M}$ *and PMPC sampling time* $T^{\mathrm{s,p}}$ *are such that each trajectory segment* $\boldsymbol{p}^*_{\tau+iT^{\mathrm{s,p}}|t-T^{\mathrm{s,p}}}, \tau \in [0, T^{\mathrm{s,p}}], i \in \mathbb{N}_{[0,N-1]}$ *is contained within the ellipsoid to used to generate the first half-space constraint,* $j = 1$ *in* (2.3)*, of the obstacle-free region* $\mathcal{F}_{iT^{\mathrm{s,p}}|t}, i \in \mathbb{N}_{[0,N-1]}$.

Remark 23. *This assumption becomes a limiting factor in scenarios with a high obstacle density, a long PMPC sampling interval, and a high velocity. If this assumption is not sat-*

isfied, the PMPC sampling time or the maximum velocity constraint needs to be reduced. This was not a limiting factor in the simulations and experiments in Section 3.6.

Combining (3.20), Assumption 12, and the fact that the convex regions are generated based on the previously optimal solution $\boldsymbol{p}^*_{[0,T^{\mathrm{p}}]|t-T^{\mathrm{s,p}}}$, $\boldsymbol{p}^*_{[T^{\mathrm{s,p}},T^{\mathrm{p}}]|t-T^{\mathrm{s,p}}}$ also satisfies the new constraints:

$$\boldsymbol{p}^*_{\tau|t-T^{\mathrm{s,p}}} \in \mathscr{F}_{\tau-T^{\mathrm{s,p}}|t},\ \tau \in [T^{\mathrm{s,p}}, T^{\mathrm{p}}], \tag{3.21a}$$

$$\boldsymbol{p}^*_{T^{\mathrm{p}}|t-T^{\mathrm{s,p}}} \in \mathscr{F}_{T^{\mathrm{p}}|t}. \tag{3.21b}$$

Thus, Property 5 (a) is ensured by the combination of Assumption 12 and the fact that the obstacle avoidance constraints are generated by growing ellipsoids starting in free space until it touches $\tilde{\mathcal{O}}$. Moreover, Property 5 (b) is satisfied by (3.21).

In conclusion, by formulation (3.19), Assumption 12, and the design of the obstacle avoidance constraints generation method, both Property 4 and Property 5 are satisfied.

The following section provides an overview of how TMPC and PMPC are co-designed and details the theoretical properties of the overall HMPC framework.

3.5. HMPC FRAMEWORK

This section provides the design and implementation overview of the overall HMPC framework in Section 3.5.1, followed by the theoretical analysis in Section 3.5.2, and a summary of the framework design and properties in Section 3.5.3.

3.5.1. DESIGN AND IMPLEMENTATION OVERVIEW

Figure 3.2 gave an overview of the HMPC framework. Given $\boldsymbol{p}^{\mathrm{g}}$ from a global planner, and $\mathscr{M}$ satisfying Assumption 12, the PMPC, formulated in (3.19) with sampling time $T^{\mathrm{s,p}}$ satisfying Assumption 12 and horizon length T^{p}, generates obstacle avoidance constraints $\mathscr{F}$ and optimizes a corresponding feasible reference trajectory $(\boldsymbol{x}^{\mathrm{r}}, \boldsymbol{u}^{\mathrm{r}})$ towards $\boldsymbol{p}^{\mathrm{g}}$. Based on the state feedback $\boldsymbol{x}$, the TMPC, formulated in (3.2) with sampling time $T^{\mathrm{s,t}}$ and horizon length T^{t}, computes the control input $\boldsymbol{u}$ required to track $(\boldsymbol{x}^{\mathrm{r}}, \boldsymbol{u}^{\mathrm{r}})$.

The main motivation for this framework is its capability to solve an expensive optimization-based planning and tracking problem on an embedded computer with limited computation power in real time. Since the optimization takes a non-negligible amount of time (as shown later in Section 3.6), it introduces a delay in both the reference trajectory update and the control input update. To ensure the dynamic feasibility of the optimized solution, the framework is implemented with timing properties highlighted in Figure 3.6.

Both PMPC and TMPC run at constant sampling times $T^{\mathrm{s,p}}$ and $T^{\mathrm{s,t}}$, respectively, with $T^{\mathrm{s,p}} = \beta T^{\mathrm{s,t}}, \beta \geq 2$. For convenience of visualization, the figure is constructed using $\beta = 3$. Note that the value of β used in the implementation is provided in Section 3.6. Furthermore, both PMPC and TMPC optimize a trajectory that becomes valid one sampling time in the future. This means that TMPC optimizes a trajectory $(\boldsymbol{u}^{\mathrm{t},*}_{\tau|t_i}, \boldsymbol{x}^{\mathrm{t},*}_{\tau|t_i})$ at $t_i - T^{\mathrm{s,t}}$ before it becomes valid at t_i, based on a reference plan $(\boldsymbol{u}^{\mathrm{p},*}_{\tau|t_i}, \boldsymbol{x}^{\mathrm{p},*}_{\tau|t_i})$ that the PMPC started optimizing at $t_{i-1} - T^{\mathrm{s,t}} = t_i - T^{\mathrm{s,p}} - T^{\mathrm{s,t}}$. At t_i, $\boldsymbol{u}^{\mathrm{t},*}_{0|t_i}$ is sent to the robot before the TMPC

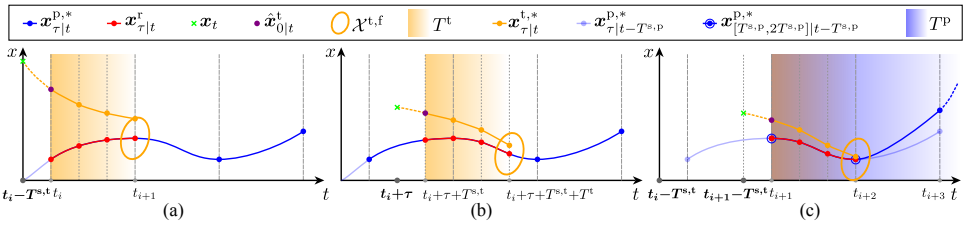


Figure 3.6: Example timing diagram with 1D state trajectories. (a) At time $t_i - T^{\mathrm{s,t}}$, the TMPC optimizes a trajectory (orange) over horizon T^{t} starting from forward-simulated state $\hat{\boldsymbol{x}}^{t}_{0|t_i}$ (purple), which is the model response when applying $\boldsymbol{u}^{\mathrm{t},*}_{[0,T^{\mathrm{s,t}}]|t_i-T^{\mathrm{s,t}}}$ starting from current state $\boldsymbol{x}_{t_i-T^{\mathrm{s,t}}}$ (green). The trajectory ends in terminal set $\mathcal{X}^{\mathrm{f,t}}$ around the reference trajectory $\boldsymbol{x}^{\mathrm{r}}_{\tau|t_i}$ (red) that becomes valid at t_i. $\boldsymbol{x}^{\mathrm{r}}_{\tau|t_i}$ is the sub-sampled version of reference plan $\boldsymbol{x}^{\mathrm{p},*}_{\tau|t_i}$ (blue). (b) In this example, the TMPC executes 2 more times ($\tau \in \{0, T^{\mathrm{s,t}}\}$) until the next reference trajectory becomes valid, thereby getting closer to the reference. (c) At time $t_{i+1} - T^{\mathrm{s,t}}$, the TMPC starts optimizing a trajectory based on a new reference plan $\boldsymbol{x}^{\mathrm{p},*}_{\tau|t_{i+1}}$ that becomes valid at t_{i+1}. This reference plan is optimized by the PMPC starting at time $t_i - T^{\mathrm{s,t}}$ and satisfies the initial state equality constraint (3.19b).

starts optimizing the next trajectory $(\boldsymbol{u}^{\mathrm{t},*}_{\tau|t_i+T^{\mathrm{s,t}}}, \boldsymbol{x}^{\mathrm{t},*}_{\tau|t_i+T^{\mathrm{s,t}}})$. Thus, the TMPC optimization can take a maximum of $T^{\mathrm{s,t}}$ and the PMPC a maximum of $T^{\mathrm{s,p}}$ in execution time.

In conclusion, the HMPC framework gives valid reference trajectories and constraints and control inputs by the satisfaction of Properties 4 and 5 and allows sending control commands to the robot at a fixed frequency.

Remark 24. *In theory, any $\beta \in \mathbb{I}_{\geq 1}$ works provided that $T^{\mathrm{p}} \geq (1 + \lceil \frac{T^{\mathrm{t}}}{T^{\mathrm{s,p}}} \rceil + 2) T^{\mathrm{s,p}}$ such that the PMPC has at least three stages free to optimize to move towards the goal and satisfy (3.19f). If $\lceil \frac{T^{\mathrm{t}}}{T^{\mathrm{s,p}}} \rceil > 1$ one has to adjust (3.19b) accordingly by constraining more PMPC stages. Given a TMPC design with fixed $T^{\mathrm{s,t}}$ and T^{t}, increasing β means that $T^{\mathrm{s,p}}$ increases, giving a coarser re-planning time and thus reduced computational load for the PMPC, and vice versa. Hence, the user can trade off PMPC execution time and performance by choosing β.*

Algorithm 1 and Algorithm 2 give an overview of the relevant design choices of the HMPC framework.

Algorithm 1 Offline design

Input: $f, \mathcal{Z}$

1 X, Y ← Solve (3.6) using gridding or convexification
2 $P, K, c^{\mathrm{s}}_j, c^{\mathrm{o}}$ ← Compute using (3.8) and (3.14)
3 α ← Choose
4 $\mathcal{X}^{\mathrm{f,t}}, \bar{\mathcal{Z}}$ ← Compute using (3.9) and (3.12)

Algorithm 2 Online implementation

Input: Results from Algorithm 1, $\mathscr{B}$, $\mathscr{R}$, $T^{\mathrm{s,t}}$, T^{t}, $T^{\mathrm{s,p}}$ (β), T^{p}, $\boldsymbol{p}^{\mathrm{g}}$ and $\mathscr{M}$
Define: $t_i = iT^{\mathrm{s,p}}, i \in \mathbb{N}_{\geq 0}$
Define: $t_{i,j} = t_i + jT^{\mathrm{s,t}}, j \in \mathbb{N}_{[0,\beta-1]}$
$i = 0, j = 0$
for $t = t_{i,j}$ **do run** TMPC
  Apply $\boldsymbol{u}^*_{\tau+t}, \tau \in [0, T^{\mathrm{s,t}}]$ (3.5)
  $\boldsymbol{x}_t \leftarrow$ Measure
  $\boldsymbol{x}_{0|t_{i,j+1}} \leftarrow$ Forward-predict using $\boldsymbol{x}_t$
  $\mathscr{F}_{\cdot|t} \leftarrow$ Select based on $\mathscr{F}_{\cdot|t_i}$
  $\boldsymbol{u}^*_{\cdot|t} \leftarrow$ Solve (3.2) with $\boldsymbol{x}_{0|t} = \boldsymbol{x}_{0|t_{i,j+1}}$
  $j \leftarrow j + 1$
  if $j = \beta - 1$ **and** $\boldsymbol{p}^{\mathrm{g}}$ not reached **then run** PMPC
    $\mathscr{F}_{\cdot|t_i}, \bar{\mathscr{F}}_{\cdot|t_i} \leftarrow$ Construct as in Section 3.4.2
    $(\boldsymbol{x}^{\mathrm{p},*}_{\cdot|t_i}, \boldsymbol{u}^{\mathrm{p},*}_{\cdot|t_i}) \leftarrow$ Solve (3.19)
    Send $(\boldsymbol{x}^{\mathrm{r}}_{\cdot|t_i} = \boldsymbol{x}^{\mathrm{p},*}_{\cdot|t_i}, \mathscr{F}_{\cdot|t_i})$ to TMPC
    $i \leftarrow i + 1$
    $j \leftarrow 0$

3.5.2. THEORETICAL ANALYSIS

First, Theorem 4 formalizes the TMPC obstacle avoidance and recursive feasibility guarantees. Secondly, Theorem 5 formalizes the recursive feasibility of the PMPC. By combining these results, Corollary 1 concludes the recursive feasibility and obstacle avoidance of the HMPC framework.

Theorem 4. *Suppose the terminal ingredients are designed according to Section 3.3.2, $\boldsymbol{r}$ satisfies Property 4, $\mathscr{F}$ satisfies Property 5, and (3.2) is feasible at $t = 0$. Then, the resulting closed-loop system ensures that (3.2) is recursively feasible, satisfies system constraints $(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathscr{Z}$ and avoids obstacle collisions $M\boldsymbol{x}_t \notin \mathscr{O}$ for all $t \geq 0$. Moreover, the tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}^{\mathrm{r}}_t\|$ asymptotically converges to zero.*

The proof is detailed in Appendix B.1. It uses a candidate solution that shifts the previously optimal solution by $T^{\mathrm{s,t}}$ and appends the terminal control law (3.10). Feasibility is ensured by the invariant terminal set, cf. Proposition 1, and convergence follows by showing that the optimal cost decreases.

Theorem 5. *Suppose that Assumption 12 holds and (3.19) is initialized with a feasible steady-state plan, i.e. $\dot{\boldsymbol{x}}^{\mathrm{p}} = 0$, $(\boldsymbol{x}, \boldsymbol{u}) \in \tilde{\mathscr{Z}}$, $\boldsymbol{p} \in \bar{\mathscr{F}}_{\cdot|0}$. Then, (3.19) is recursively feasible for $t \geq 0$, the optimal solution satisfies Property 4, and the constructed obstacle avoidance constraints satisfy Property 5.*

The recursive feasibility proof is detailed in Appendix B.2. Similar to the proof for Theorem 4, it is based on a candidate solution equal to the previously optimal solution, appended with an input ensuring steady state. Given that the previously optimal solution ends in steady state, the candidate also ends in steady state, meaning that sys-

tem and obstacle avoidance constraints are always satisfied. Satisfaction of Properties 4 and 5 follows from the reasoning explained in Section 3.4.2.

Corollary 1. *Suppose the terminal ingredients are designed according to Section 3.3.2, Algorithm 2 is initialized with a steady-state reference plan satisfying Property 4, and* (3.2) *and* (3.19) *are both feasible at* $t = 0$. *Then,* (3.2) *and* (3.19) *are recursively feasible for the closed-loop system in Algorithm 2 and the closed-loop system satisfies system constraints* $(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathcal{Z}$ *and avoids obstacle collisions* $\mathcal{R} \cap \mathcal{O} = \emptyset$ *for* $t \geq 0$. *Moreover, the tracking error* $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ *asymptotically converges to zero.*

By Theorem 5, (3.19) is recursively feasible and ensures that the reference trajectory satisfies Property 4 and obstacle avoidance constraints satisfy Property 5. Then, given the initial feasibility of (3.2), applying Theorem 4 gives system and obstacle avoidance constraints satisfaction and asymptotic convergence to the reference trajectory of the closed-loop system.

3.5.3. SUMMARY

We solve SDP (3.6) offline for known nonlinear system dynamics (2.1). A feasible solution to (3.6) gives matrices X and Y, and tightening constants c_j^{s}. X and Y are used to compute matrices P and K, which are subsequently used to calculate the tightening constants c^{o} and to construct the TMPC terminal cost, set, and control law.

Note that LMI (3.6c) ensures that c_j^{s} are Lipschitz constants, so the terminal set scaling α does not have to known before solving (3.6) and can instead be computed as $\alpha = \frac{d}{c^{\mathrm{o}}}$ with d the minimum distance between obstacles and reference trajectory.

Online, we compute the reference trajectory by solving PMPC Problem (3.19) using tightening constants c_j^{s} and c^{o} and obstacle avoidance constraints generated as described in Section 3.4.2. Theorem 5 states that the reference trajectory and corresponding constraints satisfy Properties 4 and 5, respectively, and that the PMPC is recursively feasible by including PMPC terminal constraint (3.19f).

Given the offline-computed terminal ingredients as proposed in Proposition 1 and online-computed reference trajectory satisfying Property 4 with corresponding obstacle avoidance constraints satisfying Property 5, Theorem 4 states that the online execution of TMPC Problem 3.2 guarantees asymptotic convergence to the reference trajectory, obstacle avoidance, and recursive feasibility.

Consequently, by the proposed co-design of TMPC and PMPC, Corollary 1 concludes that the HMPC framework is recursively feasible, satisfies system constraints, and avoids obstacle collisions at all times. Additionally, the timing of PMPC and TMPC as described in Section 3.5.1 allows sending control commands to the robot at a fixed frequency while maintaining the properties described above.

Important to note is that the optimized trajectory of the PMPC evolves based on the previously optimal PMPC solution, so there is no state feedback from the real system. In contrast, the state feedback is incorporated in the TMPC formulation as an initial state constraint (3.2b).

3.6. RESULTS

This section demonstrates an efficient way to generate obstacle avoidance constraints. Furthermore, it shows the performance and advantages of the proposed HMPC framework compared to SMPC, which is commonly used in local motion planning schemes [31], [69], [94]. To make a fair comparison, SMPC inherits the cost function and system dynamics from PMPC, and initial state, system, and obstacle avoidance constraints from TMPC. This means, it solves (3.19), where (3.19b), (3.19d) and (3.19e) are replaced by (3.2b), (3.2d) and (3.2e), respectively.

The results include simple simulations without model mismatch, Gazebo simulations with model mismatch, and lab experiments. All results will focus on the lab experiments and highlight the differences with simulations when relevant. The simulations and lab experiments use the same PMPC, TMPC, and SMPC settings, which are chosen to ensure real-time performance on the hardware of the real quadrotor. These settings include sampling times, horizon lengths, and weights.

We first detail the different MPC implementations and test setup in Section 3.6.1. After that, we provide the results on both obstacle avoidance constraints generation and comparison between HMPC and SMPC, in Section 3.6.2.

3.6.1. IMPLEMENTATION AND SETUP DETAILS

The implementation and setup consider several aspects, including the considered grid map to the quadrotor model, the MPC details, the software setup, and the hardware setup, as described next.

GRID MAP

The considered grid map $\mathscr{M}$ is 2D and generated offline with size 12x12 m and resolution 0.01 m. The rectangular obstacles are described by their center, width, and length. The obstacle contours, formed by thin lines of occupied grid cells, are efficiently created using OpenCV [124]. As described in Section 3.4.2, an extra inflation layer is designed around the obstacles to allow smoother planning of the PMPC around the obstacles. The same holds for the lines representing the environmental boundaries. $\mathscr{M}$ is the same for the simulation and lab results below for consistency.

QUADROTOR MODEL

The following quadrotor model is used in the MPC schemes:

$$\begin{aligned}
\begin{bmatrix} \dot{p}^x \\ \dot{p}^y \\ \dot{p}^z \end{bmatrix} &= \begin{bmatrix} v^x \\ v^y \\ v^z \end{bmatrix}, \\
\begin{bmatrix} \dot{v}^x \\ \dot{v}^y \\ \dot{v}^z \end{bmatrix} &= \begin{bmatrix} s^\phi s^\psi + c^\phi s^\theta c^\psi \\ -s^\phi c^\psi + c^\phi s^\theta s^\psi \\ c^\phi c^\theta \end{bmatrix} a - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \\
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{a} \end{bmatrix} &= \begin{bmatrix} -\frac{1}{\tau^\phi} & 0 & 0 & 0 \\ 0 & -\frac{1}{\tau^\theta} & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau^\psi} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau^a} \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ a \end{bmatrix} + \begin{bmatrix} \frac{k^\phi}{\tau^\phi} \\ \frac{k^\theta}{\tau^\theta} \\ \frac{k^\psi}{\tau^\psi} \\ \frac{k^a}{\tau^a} \end{bmatrix} \begin{bmatrix} \phi^\mathrm{c} \\ \theta^\mathrm{c} \\ \psi^\mathrm{c} \\ a^\mathrm{c} \end{bmatrix},
\end{aligned} \tag{3.22}$$

with states including 3D positions, velocities and attitudes, and acceleration given by $\boldsymbol{x} = [p^x\ p^y\ p^z\ v^x\ v^y\ v^z\ \phi\ \theta\ \psi\ a]^\top$ and inputs including 3D attitude commands and collective mass-normalized thrust, or acceleration command, given by $\boldsymbol{u} = [\phi^\mathrm{c}\ \theta^\mathrm{c}\ \psi^\mathrm{c}\ a^\mathrm{c}]^\top$, sine ($s$) and cosine ($c$) expressions, gravitational constant g, and time and gain constants $\tau^\phi = \tau^\theta = 0.18, \tau^\psi = 0.56, \tau^a = 0.050, k^\phi = k^\theta = k^\psi = k^a = 1$ for roll, pitch, yaw and acceleration respectively.

The system constraints are given by:

$$\begin{aligned}
-15\ \mathrm{m} \leq p^x,\ p^y &\leq 15\ \mathrm{m}, \\
0\ \mathrm{m} \leq p^z &\leq 4\ \mathrm{m}, \\
-2\ \mathrm{m/s} \leq v^x,\ v^y,\ v^z &\leq 2\ \mathrm{m/s}, \\
-30^\circ \leq \phi,\ \theta,\ \psi,\ \phi^\mathrm{c},\ \theta^\mathrm{c},\ \psi^\mathrm{c} &\leq 30^\circ, \\
5\ \mathrm{m/s^2} \leq a,\ a^\mathrm{c} &\leq 15\ \mathrm{m/s^2}.
\end{aligned} \tag{3.23}$$

The PMPC and SMPC leverage an extended model in which the inputs are given by $[\dot{\phi}^\mathrm{c}\ \dot{\theta}^\mathrm{c}\ \dot{\psi}^\mathrm{c}\ a^\mathrm{c}]$, with $-60^\circ/s \leq \dot{\phi}^\mathrm{c},\ \dot{\theta}^\mathrm{c},\ \dot{\psi}^\mathrm{c} \leq 60^\circ/s$, to allow for penalizing angle rates in the cost function and thus creating a smoother trajectory. The absolute angle commands $[\phi^\mathrm{c}, \theta^\mathrm{c}, \psi^\mathrm{c}]$ are included in the states in that case.

Both PMPC and TMPC implement the discretization of the quadrotor model (3.22) using the fourth-order Runge–Kutta method (RK4) with a 50 ms time interval. The TMPC has a sampling time of 50 ms, requiring 1 RK4 step, whereas the PMPC has a sampling time of 500 ms, requiring $\beta = 10$ RK4 steps. Thus, although the sampling times differ between PMPC and TMPC, the model discretization is the same.

TMPC OFFLINE COMPUTATIONS

The offline computations described in Section 2.3 are implemented using Yalmip. The attitude states occur nonlinearly, and the acceleration state occurs linearly in the linearized system dynamics (3.7). Therefore, the LMIs are generated using a grid of five points equally divided over the attitude state constraint intervals and the two endpoints of the acceleration constraint interval. The solution to SDP (3.6) is checked for 21 points

per attitude state and two points for the acceleration state constraint interval. All LMIs are satisfied at the checked points in the state space, meaning that the theoretical guarantees hold for those points. The total time to generate the LMIs, solve the SDP, and check the results is 38 s.

PMPC AND SMPC COST FUNCTION

The PMPC and SMPC aim to reach the goal with a cost similar to GO-MPC [88]. On a quadrotor, one can control the yaw angle independently from the position, meaning that the goal is described by a 3D position with yaw angle $(\boldsymbol{p}^{\mathrm{g}}, \psi^{\mathrm{g}}) \in \mathbb{R}^4$. Consequently, the cost function is given by:

$$
\begin{aligned}
&\mathscr{J}^{\mathrm{s,p}}(\boldsymbol{x}_{\tau|t}, \boldsymbol{u}_{\tau|t}, \boldsymbol{p}^{\mathrm{g}}, \psi^{\mathrm{g}}) \\
&:= w^{\mathrm{xy,g}} \mathscr{H}(\boldsymbol{p}^{xy}_{\tau|t}, \boldsymbol{p}^{xy,\mathrm{g}}) + w^{\mathrm{z,g}} (p^{z}_{\tau|t} - p^{z,\mathrm{g}})^2 \\
&+ w^{\psi,\mathrm{c}} (\psi_{\tau|t} - \psi^{\mathrm{g}})^2 + w^{a} (a^2_{\tau|t} - g)^2 \\
&+ \| \boldsymbol{u}_{\tau|t} \|^2_U + w^{\phi\theta,\mathrm{c}} ({\phi^{\mathrm{c}}_{\tau|t}}^2 + {\theta^{\mathrm{c}}_{\tau|t}}^2) + w^{\psi,\mathrm{c}} {\psi^{\mathrm{c}}_{\tau|t}}^2,
\end{aligned} \tag{3.24}
$$

with goal position weights $w^{\mathrm{xy,g}}$ and $w^{\mathrm{z,g}}$, goal yaw weight $w^{\psi,\mathrm{g}}$, acceleration weight w^{a}, input weights matrix $U = \mathrm{diag}(u_1, \ldots, u_{n^{\mathrm{u}}})$ and, and input attitude memory state weights $w^{\phi\theta,\mathrm{c}}$ and $w^{\psi,\mathrm{c}}$. Terminal cost $\mathscr{J}^{\mathrm{f,p}}$ has the same expression as stage cost $\mathscr{J}^{\mathrm{s,p}}$, but with different weight values. $\mathscr{H}(\boldsymbol{p}^{xy}_{\tau|t}, \boldsymbol{p}^{xy,\mathrm{g}})$ denotes the Huber loss [125] of the 2D Euclidean distance from position $\boldsymbol{p}^{xy}_{\tau|t}$ towards goal position $\boldsymbol{p}^{xy,\mathrm{g}}$. The linear relation further away from the origin makes the Huber loss useful to ensure moving towards the goal with approximately constant velocity without accelerating quickly and reaching the goal slowly.

PMPC AND SMPC TERMINAL STEADY STATE

The terminal state used to implement (3.19f) in both PMPC and SMPC is given by $\{v^x{=}0, v^y{=}0, v^z{=}0, \phi{=}0, \theta{=}0, a{=}g\}$.

TMPC TERMINAL SET

As Section 2.3 shows, the terminal set scaling α is a tuning parameter to trade off PMPC and TMPC performance. In this case, α is computed using $\alpha = \frac{d}{c^{\mathrm{o}}}$, in which $d = 0.1$ m is the minimum allowed distance from obstacles to reference trajectory and c^{o} is given by (3.14b).

SMPC SLACK

Both the Gazebo simulations and lab experiments in Section 3.6.2 include model mismatch. HMPC indirectly accounts for this by tightening the obstacle avoidance constraints in the PMPC. However, SMPC would become infeasible because the system would exceed the obstacle avoidance constraints. Therefore, the obstacle avoidance constraints in SMPC are implemented using a slack variable, e.g., see [126], allowing the system to exceed the constraints but with high penalization to steer the system back into the constraints set. Similar to the tightening of constraints in HMPC, SMPC uses a safety distance of 0.1 m to avoid collisions.

PMPC, TMPC, AND SMPC SETTINGS

Table 3.1 summarizes the PMPC, TMPC, and SMPC settings, including the sampling times, horizon lengths, and weights. The TMPC and SMPC sampling times are chosen sufficiently low to ensure accurate control but long enough to solve the MPC optimization problems in real time. Based on the TMPC sampling time $T^{\mathrm{s,t}}$, the PMPC sampling time is chosen using $\beta = 10$. Given $T^{\mathrm{s,t}}$, this factor is selected sufficiently low for the PMPC to optimize a reasonable plan and sufficiently high for the TMPC to satisfy the terminal set constraint.

Table 3.1: PMPC, TMPC, and SMPC settings in simulations and lab experiments.

Method	T^{s} **(s)**	T **(s)**	**Weighting matrices**
PMPC	0.5	2.5	$w^{\mathrm{xy,g^a}}$=40, $w^{\mathrm{z,g^a}}$=40, $w^{\psi,\mathrm{g^a}}$=40, $w^{\mathrm{xy,g^t}}$=200, $w^{\mathrm{z,g^t}}$=200, $w^{\psi,\mathrm{g^t}}$=200, w^{a} = 40, $w^{\phi\theta,\mathrm{c}}$ = 16, $w^{\psi,\mathrm{c}}$ = 16, U = diag(16, 16, 16, 16)
TMPC	0.05	0.5	Q= diag(2e3, 2e3, 2e3, 20, 20, 20, 100, 100, 100, 100), R= diag(2e3, 2e3, 2e3, 100)
SMPC	0.05	0.4	$w^{\mathrm{xy,g^a}}$=200, $w^{\mathrm{z,g^a}}$=200, $w^{\psi,\mathrm{g^a}}$=200, $w^{\mathrm{xy,g^t}}$=2e3, $w^{\mathrm{z,g^t}}$=2e3, $w^{\psi,\mathrm{g^t}}$=2e3, w^{a} = 200, $w^{\phi\theta,\mathrm{c}}$ = 160, $w^{\psi,\mathrm{c}}$ = 160, U = diag(160, 160, 160, 160)

The PMPC horizon is chosen long enough to be able to find a reasonable plan through the environments described below. The TMPC horizon is chosen to equal the PMPC sampling time. Furthermore, the SMPC horizon is reduced until the MPC scheme is runtime feasible on the hardware described below.

The GO-MPC objective weights used in PMPC and SMPC are chosen such that the terminal stage and other stages contribute equally to the cost function. Note that PMPC and SMPC have a similar weight ratio between the terminal stage and the rest of the horizon for a fair comparison. One of the differences is that the PMPC GO-MPC objective stage weights are increased to gain aggressiveness and improve goal-reaching performance, which is not possible for SMPC for stability reasons. Stability is ensured in both PMPC and SMPC by increasing the input weights. Note that this increase is more significant for SMPC than for PMPC.

The TMPC position state and attitude input weights are chosen large enough to ensure that the reference positions are accurately tracked to avoid collisions, and the system will not aggressively compensate for any reference tracking error, thereby increasing model mismatch. To further enhance stable flight, the acceleration input and state weights are increased. Since the reference acceleration is given in the body frame, its value is only valid for attitude values resulting in the corresponding body frame. Therefore, the attitude weights are increased as well.

Since we compute the terminal cost offline, we hard-code the TMPC weighting matrices to save computation time during runtime. In contrast, the PMPC and SMPC

weights are not hard-coded to allow the user to change them without generating a new solver.

Note that the costs are given in continuous time and discretized using Euler.

SOFTWARE SETUP

In all simulations and lab experiments, the ForcesPro NLP solver [120], [121] is leveraged, using x86 compilation for simulations and ARM compilation for lab experiments. Furthermore, in both Gazebo simulations and lab experiments, PX4 [127] stable release v1.12.3 is leveraged, using Posix compilation for simulations and NuttX compilation for lab experiments. The advantage of this setup is that both the solver and low-level controllers are the same in simulations and lab experiments to reduce sim-to-real transfer.

HARDWARE SETUP

The quadrotor used to generate the lab experiment results is the HoverGames drone [128], with the flight controller being replaced by a Pixhawk 6X mini and an added NVIDIA Jetson Xavier NX embedded computer, which has a 6-core 1400MHz NVIDIA Carmel ARMv8.2 processor. The position and orientation of the quadrotor are tracked using the Vicon Vantage V5 camera system and sent via Wi-Fi to the embedded computer, where all code runs. Figure 3.7 shows the lab experiment setup.

The simulations are run on a Dell XPS 15 laptop with a 12-core 2.60GHz Intel i7-10750H CPU.

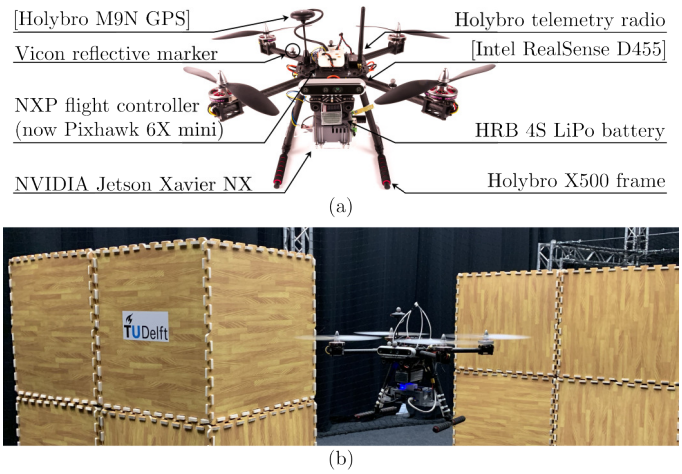


Figure 3.7: Lab experiment setup. (a) The quadrotor with important components is highlighted. Components between square brackets are not used in this work's experiments. (b) The quadrotor is flying near the obstacles in the lab.

3.6.2. RESULTS

This section presents simple simulation results focusing on the computational efficiency and size of the obstacle avoidance constraint sets in Section 3.6.2 and the effect of the

tuning parameter β in Section 3.6.2. Thereafter, it describes the performance of the HMPC framework in lab experiments and simulations in Section 3.6.2.

I-DecompUtil

The purpose of this section is twofold. First, it shows how to make the runtime code for obstacle avoidance constraint generation efficient. Second, it demonstrates the impact of the bounding box width on the computation time and the time required to reach the goal.

It is important to note that *DecompUtil* iterates through grid map $\mathscr{M}$ to find all occupied grid cells $\tilde{\mathscr{O}}$ that are contained in the bounding box $\mathscr{B}$ around the line segment. After finding the occupied grid cells, *DecompUtil* iterates through them to compute the constraints for a specific line segment. *I-DecompUtil* repeats this process for all line segments between the stages in the horizon.

The time to compute the obstacle avoidance constraints varies significantly depending on the implementation. Table 3.2 shows the mean and standard deviation of the timing results for the different combinations of two options to reduce computation time: *C++ compiler optimization*, i.e., setting compiler flag `-O3` to allow for loop unrolling, and *map pre-processing*, i.e., selecting only the relevant part corresponding to the line segment to reduce the number of occupied grid cells to iterate through. The results are obtained by moving from start to goal using an MPC with a horizon of 100 stages and a sampling time of 50 ms. A long horizon is chosen so that the impact of other, high-priority processor tasks on the constraints computation time is averaged out.

Table 3.2: Mean (standard deviation) of constraint computation times (ms) over a single run from start to goal position.

		Map pre-processing	
		No	Yes
Compiler optimization	No	2312.6 (165.0)	147.7 (69.2)
	Yes	6.1 (0.8)	5.5 (1.1)

The results clearly illustrate that compiler optimization is the main factor in reducing constraints computation time, allowing for real-time feasibility. The larger the map, the more significant the impact of map pre-processing becomes, up to the point that map pre-processing becomes necessary to ensure real-time feasibility.

Another factor to consider is the bounding box width tuning parameter. This parameter affects the size of the bounding box $\mathscr{B}$ and, correspondingly, the size of the pre-processed map. The primary purpose of $\mathscr{B}$ is to limit the number of grid cells to consider for constraints computation. As a result, this parameter trades off the speed at which the MPC scheme converges to the goal, or can find a way to the goal at all, and the constraints computation time.

Figure 3.8 shows the computation times for four different bounding box widths, given the scenario with two obstacles described in Section 3.6.2. The figure illustrates that the computation time grows with the bounding box width but saturates after a certain width, depending on the obstacle shapes and density in the environment. The variation in timing depends on the environment as well.

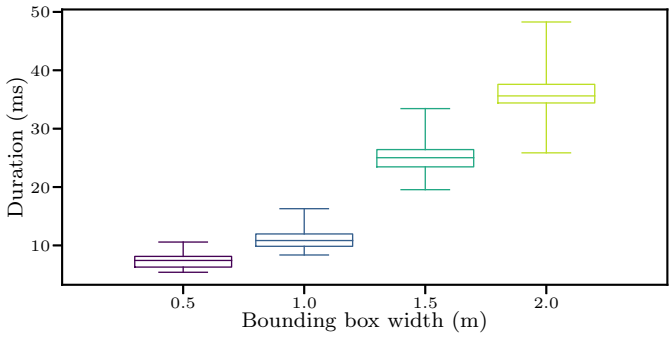


Figure 3.8: Obstacle avoidance constraints computation times for different bounding box widths when moving from start to goal with an MPC with 100 stages and 0.05 s sampling time.

Table 3.3 shows the time required to move from start to goal for SMPC and HMPC using different bounding box widths. The SMPC has such a short horizon that the size of the obstacle avoidance constraints does not influence the goal-reaching time. On the other hand, for HMPC, a short bounding box width of 0.5 m limits the goal-reaching time. This effect vanishes in our environment setup for bounding box widths $\{1, 1.5, 2\}$ m. Instead, the specific shape of the constraint regions and the ability of the PMPC to plan more aggressive maneuvers that cause a slight goal overshooting play a more significant role in the goal-reaching times for these bounding box widths.

Table 3.3: Time to reach goal with SMPC and HMPC for different bounding box widths.

		$\mathscr{B}$ **width (m)**			
		0.5	1	1.5	2
Time to reach goal (s)	SMPC	52.3	52.3	52.3	52.3
	HMPC	14.8	6.4	7.3	6.9

The combination of the results in Figure 3.8 and Table 3.3 demonstrates the trade-off above up to 1 m: the smaller the bounding box, the less computation time but the more limited the plan is, and vice versa. Since a bounding box width of 1 m gives a similar goal reaching time to the larger bounding box widths but with lower computation times, this value is used to generate the rest of the results in this section.

It is important to note that the above results are generated using the grid map as presented in Section 3.6.2. This grid map is pre-generated with thin obstacle edge lines to make SMPC feasible in real time. However, in experiments with onboard perception, the grid map will contain thicker obstacle edges due to noise in sensor measurements (e.g., from a depth camera). Furthermore, these grid maps are usually constructed in 3D instead of 2D, which will significantly increase the number of occupied grid cells and enlarge the impact of map-preprocessing.

EFFECT OF β

Next, we demonstrate the impact of β on the PMPC. Recall from Remark 24 that β trades off PMPC execution time and performance. This result is visible in Figure 3.9: the larger β, the shorter the PMPC execution time and the longer the goal-reaching time. The longer goal-reaching time is caused by the coarser re-planning time and the fact that the obstacle avoidance constraints are more conservative for a coarser plan, meaning that the PMPC can plan less far ahead in each optimization step.

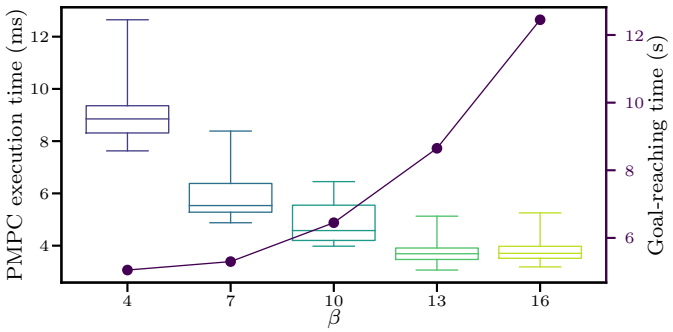


Figure 3.9: PMPC execution times (boxplots) and goal-reaching times (line) for different values of β when moving from start to goal with HMPC. The TMPC is implemented as given in Table 3.1. Given constant $T^{\mathrm{s,t}}$, the PMPC is implemented using $T^{\mathrm{s,p}} = \beta T^{\mathrm{s,t}}$ with $\beta \in \{4, 7, 10, 13, 16\}$. Except for $\beta = 16$, T^{p} was chosen to be the closest to the value in Table 3.1. For $\beta = 16$, the horizon has to be $T^{\mathrm{s,p}}$ longer according to Remark 24, which causes a slight increase in computation time compared to $\beta = 13$.

Based on Figure 3.9, we have set $\beta = 10$ in the next section. Note that $\beta = 10$ allows for convenient results interpretation since $T^{\mathrm{s,p}} = T^{\mathrm{t}}$ given the numbers in Table 3.1.

HMPC

This section describes the performance of HMPC compared to SMPC in lab experiments and highlights differences with simulation results when relevant. Two types of simulations are run: a simple simulation with perfect system knowledge, i.e., the simulated system is also (3.22) integrated using RK4 with sampling time $T^{\mathrm{s,t}}$, and one in Gazebo, i.e., including model mismatch.

First, we need to know whether all MPC schemes are feasible on the embedded computer in real-time. Figure 3.10 shows the execution times of the control loop and its main components for SMPC, PMPC, and TMPC. PMPC and TMPC comply with execution time limits, whereas SMPC is around its maximum of 50 ms and sometimes exceeds the maximum. This problem cannot easily be avoided on the embedded computer since it has a limited number of threads. The lower the number of threads, the greater the risk of the control loop being interrupted by other operating system tasks. Since most of the SMPC control loops finish in time, the frequency at which control loops are scheduled catches up with the desired frequency after exceeding the limit. Consequently, the results still provide valuable insights.

To show the impact of the embedded hardware on the execution times, Table 3.4 shows the mean and standard deviation of the control loop execution times for SMPC,

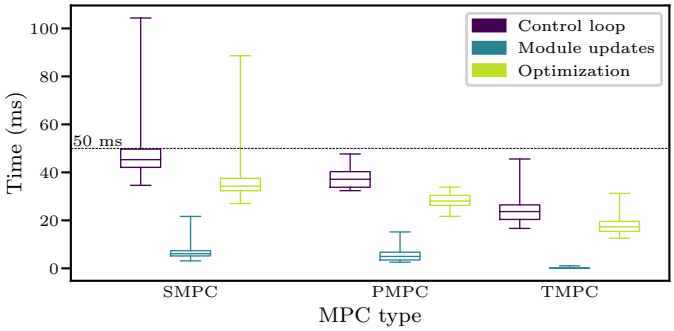


Figure 3.10: Boxplot with overall execution times for SMPC and HMPC (*Control loop*) and its most important parts: *Module updates* and *Optimization*. The maximum control loop execution time is 50 ms for real-time feasibility of SMPC and TMPC and 500 ms for PMPC. Note that *Module updates* includes constraints generation and loading new cost function terms, and is low compared to the optimization time. Since the TMPC receives the constraints from the PMPC, its *Module updates* computation time is even lower. *Optimization* is the time between calling the solver and obtaining the solution. SMPC contains several control loop cycles exceeding its real-time limit, whereas PMPC and TMPC always finish on time.

PMPC, and TMPC in both simulations and lab experiments. The numbers clearly illustrate a drastic increase in mean and standard deviation of the control loop execution time when moving from simulations to lab experiments with embedded hardware.

Table 3.4: Mean (standard deviation) of control loop execution times (ms) of SMPC, PMPC, and TMPC in simple simulations (sim), Gazebo simulations (gaz), and lab experiments using embedded hardware (lab).

		System		
		sim	gaz	lab
Method	SMPC	8.0 (1.2)	8.4 (1.3)	**46.8** (7.3)
	PMPC	4.9 (0.9)	5.2 (0.9)	**37.7** (4.2)
	TMPC	3.2 (0.6)	3.4 (0.7)	**24.1** (4.9)

The main limitation of SMPC is the computation time. Therefore, an SMPC with increasing discretization times was also tested in an attempt to reduce the number of stages while keeping the same planning horizon length. However, this scheme is not able to plan a trajectory around an obstacle as the line segments of two stages separated by a relatively large discretization time intersect with the obstacles when moving around them, i.e. Assumption 12 does not hold.

Note that since the PMPC has a sampling time of 500 ms and only needs 50 ms to solve the considered scenario, it has time left to account for processing a more realistic grid map and optimizing a plan in a more challenging environment.

Given the real-time feasibility of SMPC and HMPC, we can now analyze the schemes' closed-loop properties. Table 3.5 summarizes the goal-reaching times of SMPC and HMPC in all simulations and lab experiments. In this case, the goal is reached if the system enters the circle with a radius of 5 cm around the goal in (p^x, p^y)-plane. We can draw three conclusions based on the results. First, HMPC can move more efficiently from start to goal. Second, the presence of model mismatch significantly affects the goal-reaching

times of SMPC, while HMPC is less sensitive to the presence of model mismatch. Third, the goal-reaching time of SMPC in lab experiments is significantly shorter than in simulations. Compared to simple simulations, the quadrotor moves quicker around the obstacle edges since the combination of model mismatch and slack results in higher accelerations, also in the direction of moving alongside the obstacle. Compared to the Gazebo simulations, the real quadrotor responds more quickly to control commands. This effect becomes more significant when the cost function gives less incentive to increase attitude, and consequently, acceleration. Therefore, the real quadrotor reaches the circle with a radius of 5 cm around the goal faster.

Table 3.5: Time (s) to move from start to goal for SMPC and HMPC in simple simulations (sim), Gazebo simulations (gaz), and lab experiments (lab).

		System		
		sim	gaz	lab
Method	SMPC	52.3	107.5	35.1
	HMPC	6.4	8.4	8.1

Figure 3.11 shows the 2D closed-loop trajectories of both SMPC and HMPC in the lab experiments. Furthermore, Figure 3.12 displays the corresponding planning costs of SMPC and PMPC.

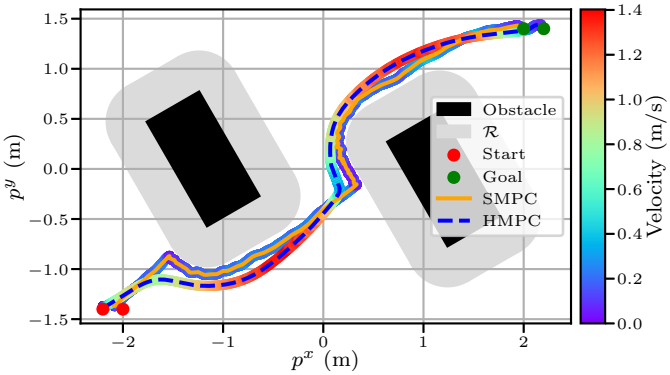


Figure 3.11: 2D closed-loop SMPC and HMPC trajectories with associated velocities in lab experiments. HMPC has a longer planning horizon and results in a smoother trajectory reaching the goal in 8.1 s versus 35.1 s for SMPC. Note that the start and goal positions for SMPC are closer. Otherwise, SMPC cannot find a way around the first encountered obstacle.

Related to the SMPC performance, we would like to highlight three aspects.

First of all, the *SMPC start and goal positions are closer* to each other compared to HMPC. With the original start and goal positions, SMPC gets stuck behind the first encountered obstacle. This is due to the short horizon that ends in steady state and the fact that SMPC is a local method. After moving the start and goal positions, SMPC still struggles to find a way around the obstacles because of its short horizon. As a result, the trajectory slowly rotates around the obstacle corners, and it takes a long time to reach the goal. This is reflected in Figure 3.12 by the 'plateaus' in the cost function.

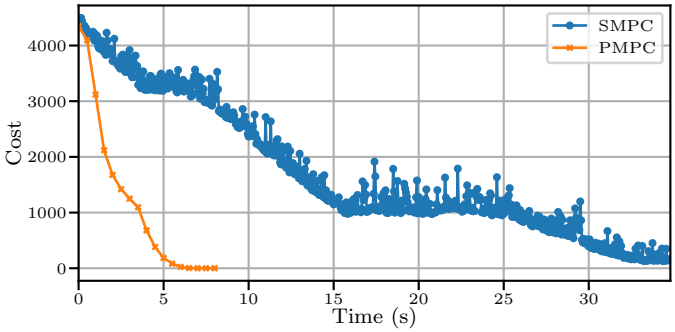


Figure 3.12: Optimal costs for SMPC and PMPC runs. Due to its longer horizon, PMPC reduces its cost quickly, whereas SMPC takes longer to move around obstacles and is sensitive to model mismatch.

Secondly, SMPC is *sensitive to model mismatch*. Without constraints softening, a slight disturbance effect causes the system to exceed the constraints, making the optimization problem infeasible. With constraints softening, the system gets aggressively pushed back due to the high slack penalization term in the cost function. However, this increases the distance to the goal, causing the robot to move back towards the constraints, repeating the same process. This is visible in Figure 3.11 by the oscillating trajectory near the obstacles and in Figure 3.12 by the noisy cost values.

To prevent collisions despite this oscillating behavior, the safety margin, as described in Section 3.6.1, is tuned to 0.1 m. Figure 3.13 shows the maximum slack value that occurs in the prediction horizon of the different SMPC runs over time. The maximum value is 0.15 m, meaning that the predicted position in at least one of the stages exceeds the safety margin of 0.1 m. This happens in the last stage, not in the first stage, meaning the robot does not collide with the obstacles. However, the SMPC scheme cannot provide strict safety guarantees in general.

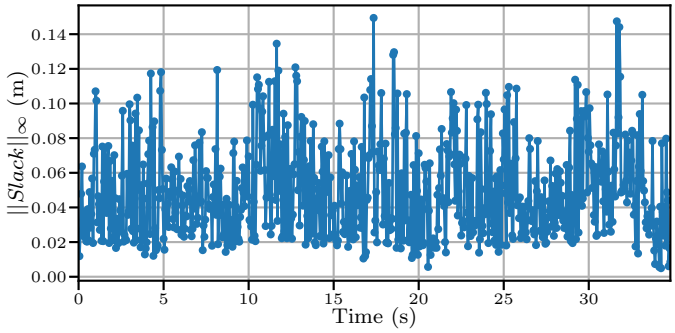


Figure 3.13: Maximum SMPC slack in the horizon over time. The maximum slack value is around 0.15 m.

Finally, SMPC is *not able to maintain altitude* during flight. This is caused by the fact that the SMPC needs to balance both the planning task, reaching the goal without collisions, and the tracking task, sending control commands to achieve stable flight, in

a single formulation with a short horizon and in the presence of model mismatch, especially in thrust dynamics. In simple simulation, the z position varies between 1.39 and 1.41 m, in Gazebo between 1.37 and 1.67 m, and in lab experiments between 0.20 and 1.56 m.

Next, we would like to highlight six aspects related to the performance of *HMPC*.

First of all, to verify the implementation of the HMPC scheme, we checked the simple simulation results and concluded that the TMPC tracking error was zero at all times. This proves the *validity of Theorems 4 and 5*.

Secondly, in contrast to SMPC, HMPC gives a similar closed-loop trajectory and similar goal-reaching times in simple simulations, Gazebo simulations, and lab experiments. Therefore, it is *less sensitive to model mismatch compared to SMPC*.

Thirdly, the PMPC horizon is long enough to *plan a reasonable trajectory around the obstacles*. This is clearly shown by the quickly decreasing PMPC cost compared to the SMPC cost in Figure 3.12. Due to its longer horizon, PMPC quickly finds a way towards the goal. On the other hand, SMPC needs time to move around the obstacles. The horizontal parts of the SMPC cost illustrate this effect. Moreover, the SMPC cost is noisier, which is caused by the model mismatch.

Fourthly, the TMPC has a sufficiently long horizon to *satisfy the terminal set constraint in every run*, as is clearly illustrated for a particular run between the obstacles in Figure 3.14.

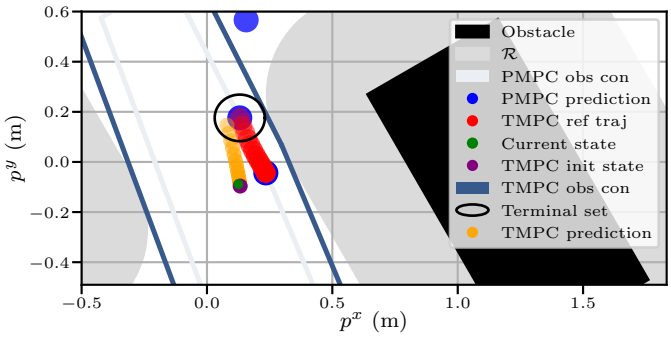


Figure 3.14: A specific snapshot of the *TMPC prediction* starting from a forward-simulated state (*TMPC init state*) and ending in terminal set (3.9), given the obstacles, their collision region indicated by the inflation with the radius of the robot region $\mathcal{R}$, *PMPC prediction* as reference plan and corresponding reference trajectory (*TMPC ref traj*), and obstacle avoidance constraints (*PMPC obs con*). Note that the PMPC obstacle avoidance constraints are tightened with respect to the TMPC obstacle avoidance constraints (*TMPC obs con*). Furthermore, the last measured state (*Current state*) is plotted. The last measured state and forward-simulated state do not overlap, showing the presence of model mismatch.

Fifthly, HMPC can *maintain altitude* during flight: in simple simulations, the z position varies between 1.37 and 1.40 m, in Gazebo between 1.39 and 1.42 m, and in lab experiments 1.32 and 1.43 m.

Finally, to show that HMPC *scales to more complex environments*, Figure 3.15 illustrates the closed-loop trajectory of HMPC in a corridor-like Gazebo environment. Even though HMPC is a local method, it finds its way from start to goal in 8.4 s without colliding with the obstacles. The maximum PMPC and TMPC control loop execution times

are 7.0 ms and 5.8 ms, respectively, meaning that HMPC is *runtime feasible*.

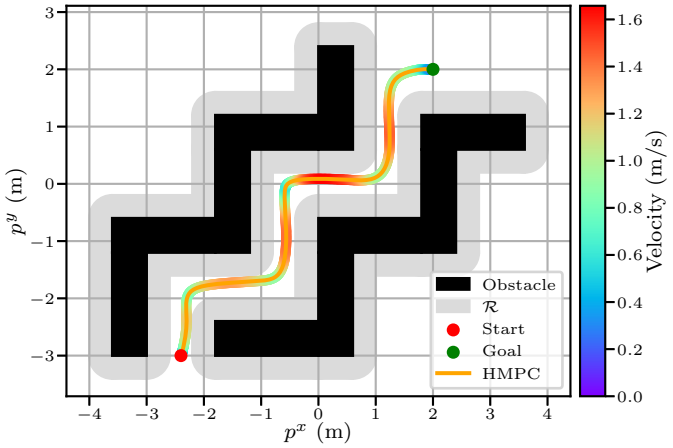


Figure 3.15: 2D closed-loop HMPC trajectory with associated velocity in a more complex Gazebo environment. HMPC can deal with more complex environments and finds its way to the goal in 8.4 s without collisions.

3.7. CONCLUSIONS AND FUTURE WORK

Optimizing a trajectory complying with nonlinear system dynamics and avoiding collisions is a computationally expensive task. Consequently, it is not feasible to run an SMPC scheme in real time that generates obstacle avoidance constraints and optimizes the corresponding trajectory with a reasonable horizon length on an embedded computer.

To address this problem, we propose a novel HMPC scheme, which includes the co-design of a PMPC and TMPC. In the offline phase, we compute the terminal ingredients of the TMPC. These are also utilized to adjust the constraints in the PMPC during runtime. By ensuring the continuity of the reference plan and consistency in updating the obstacle avoidance constraints, the PMPC can run independently from the current system state and construct a plan with a long time horizon. This saves constraints generation and optimization time on the side of the TMPC. Consequently, the TMPC runs at a higher frequency, resulting in accurate tracking of the reference plan.

We compared HMPC to SMPC in simulations without model mismatch, Gazebo simulations, and lab experiments. In general, the prediction horizon of SMPC needs to be reduced significantly to achieve real-time feasibility in the lab experiments. Therefore, SMPC has difficulty finding a way to the goal, needs constraints softening to remain feasible, and deviates significantly from the desired altitude. On the other hand, HMPC optimizes a smoother plan, does not require constraints softening, and maintains altitude more accurately.

To summarize, HMPC is more computationally efficient and less sensitive to model mismatch than SMPC. HMPC also provides tracking and recursive feasibility guarantees. It simplifies the task for robotic practitioners so they can leverage off-the-shelf nonlin-

ear solvers for motion planning and tracking without having to optimize code, linearize models, or take care of safety themselves. Furthermore, due to its computational efficiency, HMPC can easily be deployed on different robotic platforms.

While the experiments and simulations demonstrate collision avoidance and recursive feasibility, the derived theoretical guarantees are only valid in the absence of model mismatch. Therefore, Chapter 4 extends the TMPC to a *robust* formulation that can explicitly leverage pre-determined bounds on the model mismatch, similar to the ideas in [61]. As a next step, we would like to move away from motion capture cameras to onboard sensors, such as Light Detection And Ranging (LiDAR) or depth cameras, for obstacle detection and 3D constraint generation, e.g., in outdoor scenarios.

4

From Data to Safe Mobile Robot Navigation: An Efficient and Modular Robust MPC Design Pipeline

In Chapter 3, we explored how HMPC, including the nominal TMPC formulation from Section 2.3, enables real-time autonomous navigation of a mobile robot. However, that TMPC approach assumes perfect knowledge of the robot's dynamics and state, which is unrealistic in practice due to model uncertainties and measurement noise.

To address these limitations, this chapter introduces an efficient and modular pipeline for synthesizing an ROHMPC scheme. This scheme extends the HMPC framework by replacing the nominal TMPC with its robust output-feedback variant, as described in Section 2.5. The pipeline includes uncertainty quantification for general nonlinear systems, offline computation of ROMPC ingredients, calibration of tightening constants to reduce conservatism, and online implementation and validation. It leverages closed-loop experimental data to estimate disturbance bounds and generate ROMPC formulations, provided as deterministic and reproducible code. Simulations in Gazebo with a quadrotor demonstrate robust constraint satisfaction and recursive feasibility of the proposed approach.

This chapter is based on and taken in parts literally from:

D. Benders, J. Köhler, R. Babuška, J. Alonso-Mora and L. Ferranti, "From Data to Safe Mobile Robot Navigation: An Efficient and Modular Robust MPC Design Pipeline", *arXiv preprint arXiv:2508.07045*, Aug. 2025, under review.

Statement of contributions: DB performed the research and wrote the paper and JK, RB, JAM, and LF supervised the research and provided feedback on the paper.

The open-source implementation of this work is available at `https://github.com/dbenders1/rohmpc`.

4.1. Introduction

Mobile robots have significant potential to meet societal needs [2], for example, in autonomous search and rescue operations [8], intralogistics [129], and self-driving vehicles [23]. These applications require robots to navigate autonomously while avoiding collisions. To address this challenge, various planning techniques have been developed, including reactive [98], sampling-based [18], and optimization-based methods [19]. These planners typically use nonlinear dynamical system models, which, despite their accuracy, cannot capture all real-world behaviors, leading to model uncertainty. In addition, real-world experiments cannot provide accurate state information and only noisy estimates can be derived from sensor data. If not explicitly considered, such errors can jeopardize robot safety, potentially causing crashes [130]. Therefore, this work aims to ensure collision-free navigation despite uncertainties in the nonlinear robot model.

Several methods have been proposed to handle model uncertainty and ensure safe motion planning. These include CBFs [35], [113], reachability-based methods [36], [115], and MPC [22]. MPC approaches can be categorized into stochastic MPC, providing probabilistic safety guarantees, and RMPC, offering deterministic safety guarantees [131].

This work focuses on RMPC, optimizing trajectories to remain feasible and collision-free under worst-case disturbances. To mitigate conservatism, feedback controllers are used [132], for example in funnel-based approaches [108], [109] and tube MPC [77], [133], [134]. For nonlinear robot dynamics, recent approaches leverage contraction metrics [135] to derive bounds on the tracking error that are then leveraged in the MPC, see [37], [62], [79].

Designing RMPC schemes requires knowledge of the worst-case model uncertainty, which includes disturbances, parametric uncertainty, and measurement noise. State-of-the-art methods often assume simplified uncertainty sets, such as linear drag models [79], norm-bounded wind disturbances [37], and polytopic wind disturbances and mass uncertainty [62]. These assumptions can be satisfied in simulations, but are not realistic in experimental environments. Although, e.g., [79], has empirically demonstrated safety in hardware experiments using a quadrotor, this also relied on heuristically added safety margins of over half a meter. To facilitate the reliable deployment of RMPC in real-world applications, this work aims to infer bounds on the model uncertainty directly from noisy measured data.

Obtaining accurate and realistic bounds on uncertainties is crucial for the performance and reliability of RMPC schemes. Methods to quantify model uncertainty include learning-based approaches such as GPs and neural networks [136], which use statistical tools to provide probabilistic bounds. However, these tools rely on strong assumptions about the distribution of the noise and can typically not deal with noisy inputs (cf. [137]). Distribution-agnostic conformal prediction [138] offers another approach, though it requires samples of the uncertainty. Set-membership estimation [139] also provides distribution-agnostic bounds, but is susceptible to outliers. In general, the key challenge in quantifying uncertainty relates to the fact that the model error and states are not measured. Popular approaches for estimation include the extended Kalman filter and particle filter [140], but they suffer from linearized approximations or sample inefficiency, respectively. Moving horizon estimation (MHE) [22, Chap. 4] is a modern optimization-based estimation method, which can be applied to general nonlinear

robot dynamics to compute optimal estimates.

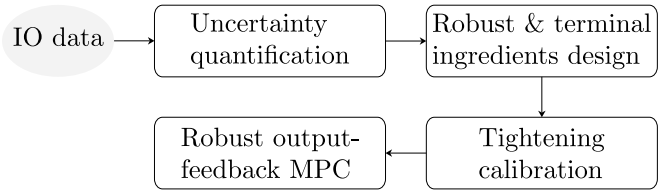


Figure 4.1: Main contribution: a ROMPC design pipeline, starting from input-output (IO) data up to the synthesis of a ROMPC scheme.

In summary, despite the strong theoretical foundation of RMPC, applications typically rely on simplifying assumptions about the model uncertainty that do not accurately reflect experiments. To overcome these limitations and move towards efficient and reliable applications of RMPC, we contribute an efficient and modular design pipeline, which is visualized in Figure 4.1. The main contributions are:

1. *Uncertainty quantification*: A scheme to determine bounds on the model uncertainty for general nonlinear systems using a simple iteration involving an MHE formulation.
2. *ROHMPC*: A robust output-feedback variant of the HMPC framework proposed in Chapter 3 that ensures safety and recursive feasibility in the presence of disturbances and measurement noise (Theorem 6).
3. *ROMPC synthesis pipeline*: An efficient, modular, and reproducible software pipeline, identifying model uncertainty, computing robust and terminal ingredients, yielding an RMPC implementation that can be directly deployed.

We apply the pipeline in simulation experiments with the Gazebo *RotorS* package [141] including significant structural model mismatch and noisy measurements, yielding an empirically safe RMPC within 2 hours.

4.2. PROBLEM FORMULATION

In this work, we consider a mobile robot described by disturbed dynamics (2.39), subject to system constraints (2.2) and obstacle avoidance constraints (2.3), and measurements (2.146).

General approach: We use a ROMPC formulation to ensure collision avoidance despite the presence of disturbances and measurement noise. This formulation tightens the constraints based on the impact of the worst-case disturbance on the system. Correspondingly, we would like to obtain a bound in the form $\boldsymbol{w} \in \mathscr{W}$ and $\boldsymbol{\eta} \in \mathscr{H}$, with polytopic disturbance and noise sets $\mathscr{W}$, $\mathscr{H}$. However, the sets $\mathscr{W}$ and $\mathscr{H}$ are not trivial to obtain since both $\boldsymbol{w}$ and $\boldsymbol{\eta}$ cannot be directly measured on the real system. Therefore, we propose a method to offline estimate $\mathscr{W}$ and $\mathscr{H}$ using an iterative MHE approach, as described in Section 4.3. These bounds are then used in the ROMPC design described in Section 4.4. This section also provides the theoretical analysis of the scheme, ensuring

recursive feasibility, robust constraint satisfaction, and a bounded tracking error. Finally, Section 4.5 demonstrates the results of the proposed uncertainty quantification method and the closed-loop properties of the ROHMPC scheme using a quadrotor simulation in Gazebo.

4.3. Uncertainty Quantification for General Nonlinear Systems

Given disturbed dynamics of state $\boldsymbol{x}$ (2.39), noisy measurements $\boldsymbol{y}$ (2.146), and a time series of IO data $(\boldsymbol{u}, \boldsymbol{y})$ of length T^{M}, the goal of this section is to quantify the model uncertainty, i.e., estimate disturbance $\boldsymbol{w}$ and measurement noise $\boldsymbol{\eta}$. Specifically, we find estimated values $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{\eta}}$ using the following formulation:

$$\min_{\hat{\boldsymbol{x}}_{\cdot|t}, \hat{\boldsymbol{w}}_{\cdot|t}, \hat{\boldsymbol{\eta}}_{\cdot|t}} \int_{\tau=0}^{T^{\mathrm{H}}} \|\hat{\boldsymbol{w}}_{\tau|t}\|_Q^2 + \|\hat{\boldsymbol{\eta}}_{\tau|t}\|_R^2 \, d\tau, \tag{4.1a}$$

$$\text{s.t. } \dot{\hat{\boldsymbol{x}}}_{\tau|t} = f(\hat{\boldsymbol{x}}_{\tau|t}, \boldsymbol{u}_{t+\tau}) + E\hat{\boldsymbol{w}}_{\tau|t}, \tag{4.1b}$$

$$\boldsymbol{y}_{t+\tau} = C\hat{\boldsymbol{x}}_{\tau|t} + F\hat{\boldsymbol{\eta}}_{\tau|t}, \tag{4.1c}$$

$$\tau \in [0, T^{\mathrm{H}}],$$

which is a receding horizon estimation scheme with prediction horizon T^{H} and weighting matrices Q and R, comparable to MHE [22, Chap. 4]. To obtain estimates $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{\eta}}$ that are representative of the real $\boldsymbol{w}$ and $\boldsymbol{\eta}$ encountered during closed-loop experiments, it is beneficial to cover a large part of the state-input space $\mathcal{Z}$. This requires a larger data length T^{M}. However, collecting more data increases the computational complexity. To prevent intractability of (4.1), we use a receding horizon implementation with horizon length $T^{\mathrm{H}} < T^{\mathrm{M}}$. Weighting matrices Q and R should reflect the inverse covariance matrices of the disturbance and measurement noise. Since their exact values are unknown, we repeatedly solve (4.1) and update Q and R until convergence of their eigenvalues. This iteration is comparable to the expectation-maximization algorithm classically used for identification [142]. In particular, the optimized cost in (4.1) is proportional to the neg-log likelihood (assuming noise is Gaussian distributed), and as we see in the experiments (Section 4.5), this iteration monotonically improves estimates. After this loop, we compute the bounding box $\hat{\mathcal{W}}$ containing all estimated disturbance samples $\hat{\boldsymbol{w}}_{\tau|t}, \tau = \frac{T^{\mathrm{H}}}{2}, t \in [0, T^{\mathrm{M}}]$, and set the model bias $\hat{\boldsymbol{w}}^{\mathrm{b}}$ equal to its center. We take the estimated disturbance samples in the middle of the horizon since their accuracy benefits from the combination of a dynamical model and past and future data. The bias can be used to increase the accuracy of the nominal model. Bounding box $\hat{\mathcal{H}}$ is computed similarly, however, centered around $\mathbf{0}$. Algorithm 3 summarizes the procedure.

4.4. ROMPC Design

The goal of this section is to describe the RMPC design that ensures safe and autonomous navigation towards $\boldsymbol{p}^{\mathrm{g}}$ while robustly avoiding collisions in the presence of $\boldsymbol{w}$

Algorithm 3 Uncertainty quantification

Input: $\boldsymbol{u}$, $\boldsymbol{y}$, f, E, C, F, T^{H}, $Q > 0$, $R > 0$
while eig(Q) and eig(R) not converged **do**
  for $t \in \{T^{\mathrm{H}}, \dots, T^{\mathrm{M}}\}$ **do**
    Solve (4.1) with Q and R
    Store $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{v}}$
  Update $Q = \mathrm{cov}(\hat{\boldsymbol{w}})$
  Update $R = \mathrm{cov}(\hat{\boldsymbol{v}})$
Compute bounding box $\hat{\mathscr{W}}$ and bias $\hat{\boldsymbol{w}}^{\mathrm{b}}$: $\hat{\boldsymbol{w}} \in \hat{\boldsymbol{w}}^{\mathrm{b}} \oplus \hat{\mathscr{W}}$
Compute bounding box $\hat{\mathscr{H}}$: $\hat{\boldsymbol{v}} \in \hat{\mathscr{H}}$

and $\boldsymbol{\eta}$. To this end, we summarize the ROMPC theory described in Section 2.5 and connect it to the uncertainty quantification method from previous section. Furthermore, we propose the ROHMPC scheme, which is based on the HMPC framework in Chapter 3, using a co-designed PMPC and TMPC scheme. The PMPC formulation only requires an adjusted model and tightening margins. Therefore, the focus in this section is on the ROMPC design. Section 4.4.1 describes the offline robust output-feedback TMPC, or ROMPC, design. This includes the combined design of an observer and an incremental Lyapunov function with the corresponding feedback law. The upper bound of this Lyapunov function, called the tube size, is used to formulate a simple constraint-tightening TMPC scheme in Section 4.4.2. Section 4.4.3 describes the offline design of suitable terminal ingredients used to prove robust constraint satisfaction and recursive feasibility of the ROHMPC scheme in Section 4.4.4.

4.4.1. OFFLINE ROMPC DESIGN

The main challenge in designing ROMPC schemes is to guarantee that the closed-loop system satisfies the original constraints in the presence of disturbances and measurement noise. To solve this problem, we compute an estimated state $\hat{\boldsymbol{x}}$ based on noisy measurements $\boldsymbol{y}$ and make sure that both controller error $\boldsymbol{\delta} \coloneqq \hat{\boldsymbol{x}} - \boldsymbol{z}$, with respect to nominal trajectory $\boldsymbol{z}$, and observer error $\boldsymbol{\epsilon} \coloneqq \boldsymbol{x} - \hat{\boldsymbol{x}}$, with respect to actual state $\boldsymbol{x}$, are bounded.

To ensure boundedness of $\boldsymbol{\delta}$, we design the control law

$$\boldsymbol{u}_{t+\tau} \coloneqq \boldsymbol{v}^*_{\tau|t} + \kappa^{\delta}(\hat{\boldsymbol{x}}_{t+\tau}, \boldsymbol{z}^*_{\tau|t}),\ \tau \in [0, T^{\mathrm{s}}], \tag{4.2}$$

which consists of a nominal input $\boldsymbol{v}^*_{\tau|t}$, recomputed every T^{s} seconds by solving the TMPC formulation described in Section 4.4.2, and a continuous feedback law $\kappa^{\delta}(\hat{\boldsymbol{x}}_{t+\tau}, \boldsymbol{z}^*_{\tau|t})$ that is applied between these discrete sampling times. Furthermore, to compute $\hat{\boldsymbol{x}}$ and ensure the boundedness of $\boldsymbol{\epsilon}$, we design an observer with the following dynamics:

$$\dot{\hat{\boldsymbol{x}}}_t \coloneqq f(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) + E\hat{\boldsymbol{w}}^{\mathrm{b}} + L(\boldsymbol{y}_t - \hat{\boldsymbol{x}}_t), \tag{4.3}$$

with observer gain $L \in \mathbb{R}^{n^{\mathrm{x}} \times n^{\mathrm{y}}}$.

Leveraging RMPC designs using contraction metrics [37], [62], [79], including their extension to the output-feedback case, see [64] and Section 2.5, we jointly optimize for

$$\min_{\substack{X^\delta, Y^\delta(\boldsymbol{x}),\\ c_j^{\mathrm{s}2}, c^{\mathrm{o}2}, \epsilon^2}} \quad c^{\mathrm{c,o}} c^{\mathrm{o}2} + \sum_{j=1}^{n^{\mathrm{s}}} c_j^{\mathrm{c,s}} c_j^{\mathrm{s}2} + c^{\epsilon}\epsilon^2, \tag{4.4a}$$

$$\text{s.t. } X^\delta \succeq 0, \tag{4.4b}$$

$$A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \big(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\big)^\top + 2\rho X^\delta \preceq 0, \tag{4.4c}$$

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x}) + \big(A(\boldsymbol{\zeta})X^\delta + B(\boldsymbol{\zeta})Y^\delta(\boldsymbol{x})\big)^\top + \lambda^\delta X^\delta & LCX^\delta & LF\boldsymbol{\eta} \\ (LCX^\delta)^\top & -\lambda^{\delta,\epsilon} X^\delta & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2 - \lambda^\delta \delta^2 \end{bmatrix} \preceq 0, \tag{4.4d}$$

$$\begin{bmatrix} X^\delta\big(A(\boldsymbol{\zeta}) - LC\big)^\top + \big(A(\boldsymbol{\zeta}) - LC\big)X^\delta + \lambda^\epsilon X^\delta & E\boldsymbol{w}^0 - LF\boldsymbol{\eta} \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top & -\lambda^\epsilon\epsilon^2 \end{bmatrix} \preceq 0, \tag{4.4e}$$

$$\begin{bmatrix} c_j^{\mathrm{s}2} & L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix} \\ \Big(L_j^{\mathrm{s}} \begin{bmatrix} Y^\delta(\boldsymbol{x}) \\ X^\delta \end{bmatrix}\Big)^\top & X^\delta \end{bmatrix} \succeq 0, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{4.4f}$$

$$\begin{bmatrix} c^{\mathrm{o}2} I^{n^{\mathrm{p}}} & MX^\delta \\ (MX^\delta)^\top & X^\delta \end{bmatrix} \succeq 0, \tag{4.4g}$$

$$\forall \boldsymbol{\zeta} := \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{x} \end{bmatrix} \in \mathcal{Z}, \quad \boldsymbol{w}^0 \in \mathrm{vert}(\hat{\mathcal{W}}), \quad \boldsymbol{\eta} \in \mathrm{vert}(\hat{\mathcal{H}}).$$

an incremental Lyapunov function with constant metric $P^\delta \in \mathbb{R}^{\mathrm{n^x} \times \mathrm{n^x}}$ and corresponding feedback law $\kappa^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ by solving the SDP (4.4). This SDP formulation differs from (2.189) in Section 2.5 by additionally optimizing for ϵ, as mentioned in Remark 20. This problem uses a given observer gain L, and treats the contraction rate $\rho > 0$ and multipliers $\lambda^\delta, \lambda^{\delta,\epsilon}, \lambda^\epsilon \geq 0$ as hyperparameters. Furthermore, the Jacobians in (4.4) are given by

$$A(\boldsymbol{\zeta}) = \left.\frac{\partial f^{\mathrm{w}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})}{\partial \boldsymbol{x}}\right|_{\zeta}, \quad B(\boldsymbol{\zeta}) = \left.\frac{\partial f^{\mathrm{w}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})}{\partial \boldsymbol{u}}\right|_{\zeta}, \tag{4.5}$$

The resulting incremental Lyapunov function,

$$V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) := (\hat{\boldsymbol{x}} - \boldsymbol{z})^\top P^\delta (\hat{\boldsymbol{x}} - \boldsymbol{z}), \tag{4.6}$$

quantifies controller error $\boldsymbol{\delta}$ and is guaranteed to contract by at least rate ρ under robust feedback law

$$\kappa^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) = \boldsymbol{v} + \int_0^1 K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big) ds \boldsymbol{\delta}, \tag{4.7}$$

where $\boldsymbol{\gamma}^{\mathrm{x}}(s) = \boldsymbol{z} + s\boldsymbol{\delta}$ is the geodesic, which is a straight line connecting $\hat{\boldsymbol{x}}$ and $\boldsymbol{z}$ with derivative $\frac{\partial \boldsymbol{\gamma}^{\mathrm{x}}(s)}{\partial s} = \boldsymbol{\delta}$. The matrices P^δ and $K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big)$ are obtained by the following

change of coordinates:

$$P^\delta = X^{\delta^{-1}}, \quad K^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big) = Y^\delta\big(\boldsymbol{\gamma}^{\mathrm{x}}(s)\big)P^\delta. \tag{4.8}$$

This contractivity property is imposed by LMI (4.4c).

Remark 25. *Note that one could use a state-dependent metric [37], [62], [79] to reduce conservatism. In this case, evaluating the geodesic $\boldsymbol{\gamma}^{\mathrm{x}}(s)$ and the Lyapunov function $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ would require solving an optimization problem. Since we require the evaluation of $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ in the terminal set constraint (4.12h), this would have resulted in a significant increase in the computational demand. In the special case of a constant feedback matrix K^δ, the control law (4.2) reduces to $\boldsymbol{v} + K^\delta \boldsymbol{\delta}$, which is also leveraged in the simulations later.*

We bound $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ from above by tube size s as

$$\sqrt{V^\delta(\hat{\boldsymbol{x}}_{\tau|t}, \boldsymbol{z}_{\tau|t})} \leq s_\tau, \ t \geq 0, \ \tau \in [0, T], \tag{4.9}$$

with $\dot{s}_\tau = -\rho s_\tau + \bar{w}^{\mathrm{o}}$. Thus, controller error $\boldsymbol{\delta}$ is bounded by tube size s. Furthermore, s grows with bounded factor $\bar{w}^{\mathrm{o}} > 0$ and contracts with rate ρ. Due to its ρ-contractivity, s cannot grow unbounded, making it suitable as a constraint tightening factor for system constraints (2.2) and obstacle avoidance constraints (2.3).

To show constraint satisfaction for $\boldsymbol{x}$, we also need to account for the difference between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$. By the uncertainty quantification (UQ) method in Section 4.3, we know that the measurement noise is bounded, i.e., $\hat{\boldsymbol{\eta}} \in \hat{\mathscr{H}}$. Combined with observer (4.3), we can construct a constant tube ϵ bounding $\boldsymbol{\epsilon}$ as

$$\sqrt{V^\delta(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t)} \leq \epsilon, \ t \geq 0. \tag{4.10}$$

Thus, we ensure satisfaction of constraints in (2.2) and (2.3) by tightening the constraints on the nominal trajectory z proportional to the two error bounds ϵ, s using suitable constraint tightening constants. The SDP (4.4) optimizes for the smallest constraint tightening constants $c_j^{\mathrm{s}}, c_j^{\mathrm{s,o}}, j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$, with

$$c_j^{\mathrm{s,o}} = \begin{cases} 0, & j \in \mathbb{N}_{[1,2n^{\mathrm{u}}]} \\ c_j^{\mathrm{s}}, & j \in \mathbb{N}_{[2n^{\mathrm{u}}+1,n^{\mathrm{s}}]} \end{cases}, \tag{4.11}$$

and c^{o}. This is imposed by the combination of (4.4a), (4.4d), (4.4e), (4.4f), and (4.4g). To avoid conservatism in certain directions of the state space, the constraint tightening constants are normalized by their corresponding constant constraint intervals $c_j^{\mathrm{c,s}}$ and $c^{\mathrm{c,o}}$ in (4.4a). In this case, $c^{\mathrm{c,o}}$ represents the minimum desired distance to obstacles. Moreover, c^ϵ is a penalty term to reduce ϵ.

Proposition 2 formalizes the properties of the ROMPC design:

Proposition 2 (ROMPC design)**.** *Consider a nominal trajectory $(\boldsymbol{z}, \boldsymbol{v})$ satisfying system constraints (2.2) tightened by $c_j^{\mathrm{s}} s_\tau + c_j^{\mathrm{s,o}} \epsilon$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$ and obstacle avoidance constraints (2.3) tightened by $c^{\mathrm{o}}(s_\tau + \epsilon)$, $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$. Then, the closed-loop trajectory $(\boldsymbol{x}, \boldsymbol{u})$ under robust feedback law (4.2) and observer (4.3) subject to disturbances $\hat{\boldsymbol{w}} \in \hat{\boldsymbol{w}}^{\mathrm{b}} \oplus \hat{\mathscr{W}}$ and measurement noise $\hat{\boldsymbol{\eta}} \in \hat{\mathscr{H}}$ satisfies (2.2) and (2.3).*

Proof. See Section 2.5. □

4.4.2. TMPC FORMULATION

Given the robust output-feedback quantities designed in the previous section, we formulate the TMPC problem at time t as follows:

$$\min_{\boldsymbol{z}_{\cdot|t},\boldsymbol{v}_{\cdot|t}} \ \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}_{T|t},\boldsymbol{x}^{\mathrm{r}}_{t+T}) + \int_0^T \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}_{\tau|t},\boldsymbol{v}_{\tau|t},\boldsymbol{r}_{t+\tau})\,d\tau, \tag{4.12a}$$

$$\text{s.t. } \boldsymbol{z}_{0|t} = \hat{\boldsymbol{x}}_t, \tag{4.12b}$$

$$s_0 = 0, \tag{4.12c}$$

$$\dot{\boldsymbol{z}}_{\tau|t} = f(\boldsymbol{z}_{\tau|t},\boldsymbol{v}_{\tau|t}) + E\hat{\boldsymbol{w}}^{\mathrm{b}}, \tag{4.12d}$$

$$\dot{s}_\tau = -\rho s_\tau + \bar{w}^{\mathrm{o}}, \tag{4.12e}$$

$$g^{\mathrm{s}}_j(\boldsymbol{z}_{\tau|t},\boldsymbol{v}_{\tau|t}) + c^{\mathrm{s}}_j s_\tau + c^{\mathrm{s,o}}_j \epsilon \le 0,\ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]} \tag{4.12f}$$

$$g^{\mathrm{o}}_{j,\tau|t}(M\boldsymbol{z}_{\tau|t}) + c^{\mathrm{o}}(s_\tau + \epsilon) \le 0, \quad j \in \mathbb{N}_{[1,n^{\mathrm{o}}]} \tag{4.12g}$$

$$(\boldsymbol{z}_{T|t}, s_T, \epsilon) \in \mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}), \tag{4.12h}$$

$$\tau \in [0,T],$$

with reference trajectory $\boldsymbol{r} = [\boldsymbol{x}^{\mathrm{r}\top}\ \boldsymbol{u}^{\mathrm{r}\top}]^\top$, stage cost $\mathscr{J}^{\mathrm{s}}(\boldsymbol{z},\boldsymbol{v},\boldsymbol{r}) = \|\boldsymbol{z}-\boldsymbol{x}^{\mathrm{r}}\|^2_Q + \|\boldsymbol{v}-\boldsymbol{u}^{\mathrm{r}}\|^2_R, Q \succ 0, R \succ 0$, terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}}) = \|\boldsymbol{z}-\boldsymbol{x}^{\mathrm{r}}\|^2_P, P \succ 0$, and prediction horizon $T > 0$. In this formulation, Q and R can be manually tuned. Terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})$ and terminal set $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ are designed as described in Section 4.4.3.

4.4.3. OFFLINE TERMINAL INGREDIENTS DESIGN

The goal of this section is to find a terminal control law that renders $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ invariant and ensures that we know a lower bound on the decrease of $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})$ over time. These properties are used to prove the recursive feasibility and trajectory tracking in Section 4.4.4, respectively.

The following proposition states a suitable design for terminal set $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$ and terminal cost $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})$:

Proposition 3 (Terminal ingredients)**.** *Terminal control law* $\kappa^{\mathrm{f}}(\hat{\boldsymbol{x}},\boldsymbol{r}) := \boldsymbol{u}^{\mathrm{r}} + \kappa^{\delta}(\hat{\boldsymbol{x}},\boldsymbol{x}^{\mathrm{r}})$ *ensures that:*

1. *there exists a terminal set scaling* $\alpha > 0$ *such that*

$$\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}}) := \left\{ \boldsymbol{z} \in \mathscr{X}, s \in \mathbb{R}, \epsilon \in \mathbb{R} \;\middle|\; \sqrt{V^{\delta}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})} + s_T + \epsilon \le \alpha \right\}, \tag{4.13}$$

   is robustly positive invariant under control law $\kappa^{\mathrm{f}}(\hat{\boldsymbol{x}},\boldsymbol{r})$ *with* $\boldsymbol{x}^{\mathrm{r}}$ *satisfying* (4.12f) *and* (4.12g) *with* s_τ *replaced by the invariant tube size* $s_\infty = \frac{\bar{w}^{\mathrm{o}}}{\rho}$,

2. *tightened system constraints* (4.12f) *and obstacle avoidance constraints* (4.12g) *are satisfied in* $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$*;*

3. *terminal cost* $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z},\boldsymbol{x}^{\mathrm{r}})$ *satisfies the following decrease property:*

$$\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T^{\mathrm{s}}+T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T^{\mathrm{s}}+T}) - \mathscr{J}^{\mathrm{f}}(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T}) \le -\int_T^{T+T^{\mathrm{s}}} \mathscr{J}^{\mathrm{s}}(\boldsymbol{z}^*_{\tau|t}, \boldsymbol{v}^*_{\tau|t}, \boldsymbol{r}_{t+\tau})d\tau, \tag{4.14}$$

with terminal cost matrix P computed using SDP (4.15).

Proof. 1) is proven by showing that terminal set constraint (4.12h) is satisfied at $T+\tau, \tau \geq 0$ given that it is satisfied at $\tau = 0$. We know that $\sqrt{V^{\delta}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})}$ contracts with factor ρ under feedback law (4.7). It can be shown that this reduction cancels out against the increase in tube size $s_{T+\tau}$ for $\tau > 0$. Since ϵ is constant, (4.12h) is satisfied for $\tau \geq 0$. This result holds for $\alpha \geq \frac{\bar{w}^{o}}{\rho} + \epsilon$. See Section 2.5 for more details. 2) Since s_T and ϵ are contained in $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$, tightened system constraints (4.12f) and obstacle avoidance constraints (4.12g) are satisfied in $\mathscr{X}^{\mathrm{f}}(\boldsymbol{x}^{\mathrm{r}})$. 3) LMI (4.15c) in SDP (4.15) enforces that $\mathscr{J}^{\mathrm{f}}(\boldsymbol{z}, \boldsymbol{x}^{\mathrm{r}})$ also decreases according to (4.14), see Chapter 3. □

$$\min_{P} \ \operatorname{trace} P, \tag{4.15a}$$

$$\text{s.t. } P \succeq 0, \tag{4.15b}$$

$$\left(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}\right)^{\top} P + P\left(A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta})K^{\delta}\right) + Q + {K^{\delta}}^{\top} R K^{\delta} \preceq 0, \quad \forall \boldsymbol{\zeta} \in \mathscr{Z}. \tag{4.15c}$$

4.4.4. THEORETICAL ANALYSIS

Theorem 6 formalizes the theoretical guarantees associated with the proposed ROHMPC framework:

Theorem 6. *Suppose the robust and TMPC terminal ingredients are designed offline according to Sections 4.4.1 and 4.4.3, the system starts at steady-state, i.e., $\dot{\boldsymbol{x}}_0 = 0$, and both the PMPC problem, see Chapter 3, and TMPC problem (4.12) are feasible at $t = 0$. Then, both problems are recursively feasible, the closed-loop system (4.2) satisfies system constraints $(\boldsymbol{x}_t, \boldsymbol{u}_t) \in \mathscr{Z}$ and avoids obstacle collisions $M\boldsymbol{x}_t \notin \mathscr{O}$ for all $t \geq 0$. Moreover, the average tracking error $\|\boldsymbol{x}_t - \boldsymbol{x}_t^{\mathrm{r}}\|$ is bounded over time.*

Proof. The PMPC is recursively feasible following the proof provided in Appendix B.2. Tightening the system and obstacle avoidance constraints in the PMPC formulation with positive invariant tube α guarantees that the reference trajectory satisfies the robust equivalent of the properties detailed in Chapter 3. Consequently, the combination of Proposition 3 and the recursive feasibility proof in Section 2.5 guarantees that the TMPC, and thus the ROHMPC framework, is recursively feasible. Therefore, the closed-loop system (4.2) robustly satisfies the system constraints and obstacle avoidance constraints. Furthermore, the average tracking error is bounded over time as shown by a combination of Proposition 3 and the trajectory tracking proof in Section 2.5. □

4.5. RESULTS

This section first presents relevant implementation details (Section 4.5.1) and continues with the results of the proposed uncertainty quantification method (Section 4.5.2), the robust output-feedback design and tightening calibration (Section 4.5.3), and ROHMPC scheme (Section 4.5.4).

4.5.1. Implementation Details

This section provides the nominal quadrotor model and its corresponding constraints in Section 4.5.1, the model uncertainty description in Section 4.5.1, and the software setup used to generate the results in the next sections in Section 4.5.1.

Quadrotor Model and Constraints

The nominal quadrotor model is based on [143]:

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{v} \\ \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^\top &= {}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R}^{\mathrm{c}}\boldsymbol{\Omega} \\ \dot{\boldsymbol{v}} &= {}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R}^{\mathrm{r}}\begin{bmatrix} 0 & 0 & \frac{T}{m} \end{bmatrix}^\top - \begin{bmatrix} 0 & 0 & g \end{bmatrix}^\top \\ I\dot{\boldsymbol{\Omega}} &= -\boldsymbol{\Omega} \times I\boldsymbol{\Omega} + \boldsymbol{\tau}, \end{aligned} \tag{4.16}$$

with states including 3D positions in inertial frame $\mathcal{I}$, ZYX Euler angles, velocities and angular velocities in the body frame, thrust and torques defined as a linear function of the inputs $\boldsymbol{u} = [t_0, t_1, t_2, t_3]^\top$ in which t_i is the thrust of rotor i, gravitational constant $g = 9.8124$, mass $m = 0.617$, inertia matrix $I = \mathrm{diag}(0.00164, 0.00184, 0.0030)$, and coordinate and rate rotation matrices ${}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R}^{\mathrm{c}}$ and ${}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R}^{\mathrm{r}}$ to convert coordinates and rates from body to inertial frame. Sine, cosine, and tangent functions are defined as $s\theta = \sin(\theta)$, $c\theta = \cos(\theta)$, $t\theta = \tan(\theta)$, for example.

The quadrotor is subject to the following constraints, derived from the trajectories used to generate the results in Section 4.5.2:

$$\begin{aligned} -4\ \mathrm{m} \leq p^x,\ p^y &\leq 4\ \mathrm{m}, \\ 0\ \mathrm{m} \leq p^z &\leq 4\ \mathrm{m}, \\ -2\ \mathrm{m/s} \leq v^x,\ v^y,\ v^z &\leq 2\ \mathrm{m/s}, \\ -0.1\ \mathrm{rad} \leq \phi,\ \theta,\ \psi &\leq 0.1\ \mathrm{rad}, \\ -0.3\ \mathrm{rad/s} \leq \Omega^x,\ \Omega^y,\ \Omega^z &\leq 0.3\ \mathrm{rad/s}, \\ 1.3936\ \mathrm{N} \leq t_0,\ t_1,\ t_2,\ t_3 &\leq 1.6336\ \mathrm{N}. \end{aligned} \tag{4.17}$$

Model Uncertainty

The model mismatch $\boldsymbol{w}$ arises due to the difference between the simulation of (4.16) and the dynamics simulated in the *RotorS* simulator, which are estimated in Section 4.5.2. Since the *RotorS* simulator is a physics-based simulator that structurally differs from the nominal model, we choose $E = I^{n^x}$ and $w \in \mathbb{R}^{n^x}$. The outputs $\boldsymbol{y}$ are given by a noisy measurement of the state $\boldsymbol{x}$ with $C = I^{n^x}$, $F = I^{n^x}$, and $\boldsymbol{\eta} \in \mathbb{R}^{n^x}$. In this case, $\boldsymbol{\eta}$ is generated as uniform noise with the following bounds:

$$\begin{aligned} -0.00005\ \mathrm{m} \leq \eta^{p^x},\ \eta^{p^y},\ \eta^{p^z} &\leq 0.00005\ \mathrm{m}, \\ -0.000628\ \mathrm{m/s} \leq v^x,\ v^y,\ v^z &\leq 0.000628\ \mathrm{m/s}, \\ -0.0005\ \mathrm{rad} \leq \phi,\ \theta,\ \psi &\leq 0.0005\ \mathrm{rad}, \\ -0.00076\ \mathrm{rad/s} \leq \Omega^x,\ \Omega^y,\ \Omega^z &\leq 0.00076\ \mathrm{rad/s}. \end{aligned} \tag{4.18}$$

SOFTWARE SETUP

The code corresponding to this work is available at https://github.com/dbenders1/rohmpc. In constructing the software pipeline, special care was taken to ensure modularity and reproducibility. One aspect to highlight is the fact that the code contains an adjusted version of the open-source stack Agilicious [40] in order to ensure deterministic *RotorS* simulation results. Moreover, the results are trivial to reproduce as a Docker container is provided, in line with the recommendation in [122].

The results presented in the next sections are generated using a Dell XPS 15 laptop with a 12-core 2.60GHz Intel i7-10750H CPU. We leverage the ACADOS sequential quadratic programming solver [144] to solve UQ problem (4.1), Mosek [145] to solve SDP (4.4), and the ForcesPro NLP solver [121] to solve the PMPC problem, see Chapter 3, and TMPC problem (4.12).

4.5.2. UNCERTAINTY QUANTIFICATION

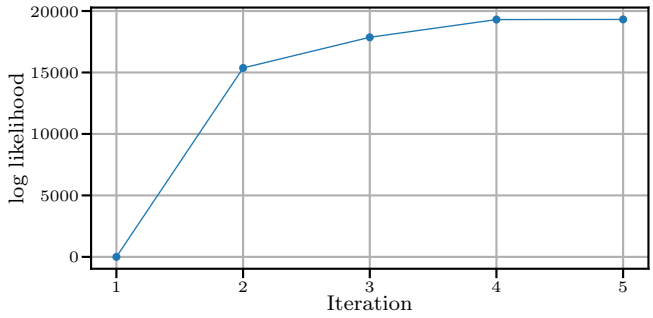
Recall that the purpose of solving UQ problem (4.1) is to find bounding boxes $\hat{\mathcal{W}}$ and $\hat{\mathcal{H}}$ that are used in the ROMPC design in Section 4.4. We solve (4.1) using IO data obtained by flying 4 different trajectories using the built-in MPC [143] in Agilicious in the *RotorS* simulator. These trajectories are designed to cover a relevant part of the state-input space and consist of clockwise and counter-clockwise circles and lemniscates with radius 1 m and frequency 0.3 Hz.

Note that there is a fundamental limitation in estimating $\boldsymbol{w}, \boldsymbol{\eta}$ using (4.1): $\boldsymbol{y}$ can be explained by both the effect of $\boldsymbol{w}$ and $\boldsymbol{\eta}$, which introduces ambiguity. As a result, the solution to (4.1) practically converges to either explaining $\boldsymbol{y}$ mainly by $\boldsymbol{w}$ or by $\boldsymbol{\eta}$, depending on the initialization of Q and R. To mitigate this issue, we start from $Q = I^{n_x}$ and $R = I^{n_\eta}$ and assume that $\mathcal{H}$ is known, i.e., it can be derived from the sensor datasheets, and add them as constraints to (4.1). In this work, we use the bounds in (4.18).

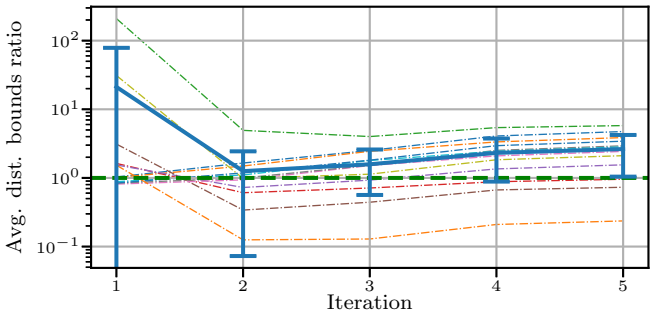
Figure 4.2a demonstrates that solving (4.1) gives a reasonable result: it maximizes the likelihood, yielding a maximum likelihood estimate, making the estimates the most likely explanation. Since the results are based on the *RotorS* simulator, we have access to the ground truth states. Therefore, we can run (4.1) with $\boldsymbol{y} = \boldsymbol{x}$ and ignore $\boldsymbol{\eta}$ to obtain the ground truth $\boldsymbol{w}$ and $\mathcal{W}$. This allows to compare the estimated disturbance bounds with the ground truth ones, giving a ratio of one in case of perfect estimation. Figure 4.2b visualizes this value by the dashed green line. Moreover, Figure 4.2b shows the average of the ratios of lower and upper bounds for each disturbance component with a dash-dotted line and the total mean ratio including one standard deviation with the blue line. Thus, 2 iterations, provide, on average, the best disturbance bound ratios. Correspondingly, we stop Algorithm 3 after 2 iterations, yielding a total computation time of 3408 s. The resulting root mean squared error of all lower and upper disturbance bounds is 0.00156.

4.5.3. ROBUST OUTPUT-FEEDBACK DESIGN

Based on $\hat{\mathcal{W}}$ and $\hat{\mathcal{H}}$ from the previous section, we solve (4.4) in 295 s. While the resulting feedback κ^δ robustly stabilizes the system, the corresponding constraint tightening is overly conservative. To reduce conservatism, we empirically determine the constants



(a)



(b)

Figure 4.2: Results after 5 iterations of Algorithm 3 based on the clockwise circle trajectory. (a) The log likelihood monotonically increases over the iterations, yielding a maximum likelihood estimate. (b) The average ratios of the lower and upper bounds of the estimated disturbance bounds with respect to the ground truth per disturbance component (dash-dotted), the total mean ratio including one standard deviation (blue solid), and the ideal disturbance bound ratio (green dashed). Initially, the ratios are often off by a factor of 100, but after 2 iterations most ratios converge relatively close to one and the standard deviation decreases significantly.

used for constraint tightening, ρ, $\bar{w}^{\mathrm{o}}$, and ϵ, corresponding to the optimized feedback κ^{δ} and Lyapunov function $V^{\delta}(\hat{\boldsymbol{x}}, \boldsymbol{z})$.

To calibrate ρ, we track a nominal stable reference trajectory with feedback controller (4.7) and observer (4.3) and record the nominal reference and the estimated states over time. Using this data, we can compute the tightest s-tube (4.9) over multiple steps, as visualized in Figure 4.3. We repeat this process for multiple grid points of ρ and pick ρ with the smallest tightening. This computation takes 3.5 s using a nominal reference duration of 10 s sampled with the simulator, estimation, and feedback frequency of 500 Hz, and gives $\rho = 12.5$.

Given a calibrated ρ, we can calibrate $\bar{w}^{\mathrm{o}}$ and ϵ by comparing the estimated state with the one-step ahead predictions of a closed-loop nominal MPC scheme and the ground truth state, respectively. In this simulation, the ground truth state is available. However, in reality one has to pick the more conservative value for ϵ given by SDP (4.4). This calibration takes 30.5 s in total, for 144 s of flight time sampled at the TMPC sampling frequency of 100 Hz for $\bar{w}^{\mathrm{o}}$ and sampled at 500 Hz for ϵ. The resulting values are $\bar{w}^{\mathrm{o}} = 0.0337$ and $\epsilon = 0.00345$. The trajectories used to calibrate $\bar{w}^{\mathrm{o}}$ and ϵ are more ag-

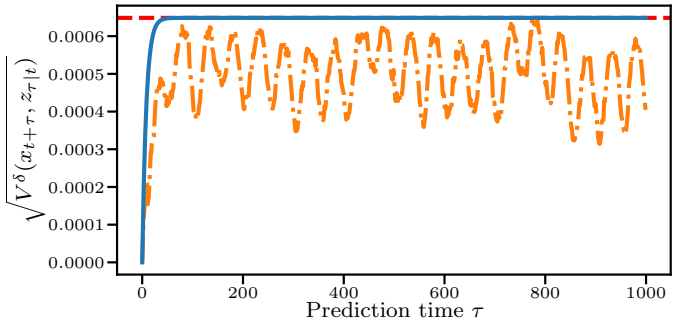


Figure 4.3: The worst-case Lyapunov error over prediction time τ (orange dash-dotted), the maximum tube size (red dashed), and the fitted tube based on the calibrated ρ (blue solid).

gressive than the ones produced by the ROHMPC framework described next, so they are reasonable to use to guarantee safety.

4.5.4. CLOSED-LOOP ROHMPC SIMULATION

Given the calibrated tightening constants ρ, $\bar{w}^{o}$, and ϵ, we can compute α used for the tightening in the PMPC and run the closed-loop ROHMPC scheme. The goal of this scheme is to fly the quadrotor from a starting position to a goal position and safely avoid the obstacle in between. Figure 4.4 visualizes the relevant part of the traversed trajectory (close to the obstacle) and shows that the closed-loop system stays within the total TMPC tube, accurately follows the plan, and safely avoids the obstacle. Furthermore, the total TMPC tube does not grow further than the tube used to tighten the constraints in the PMPC, and all tubes are contained in the obstacle-free space. Finally, the closed-loop system is empirically shown to guarantee constraint satisfaction and recursive feasibility.

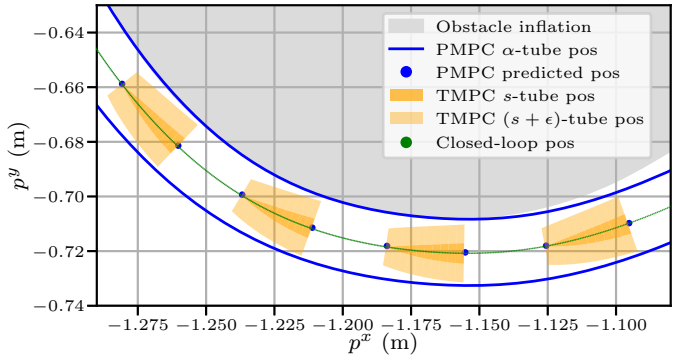


Figure 4.4: The resulting tubes, plan, and position of the ROHMPC framework in closed-loop simulation with the physics simulator. The predicted tube captures the effect of noise and disturbances, thus robustly ensuring collision avoidance in a systematic fashion.

4.6. Conclusion

Guaranteeing safety on autonomous mobile robots is a challenging topic, especially in the presence of disturbances and measurement noise. Whereas other approaches typically assume specific disturbance effects or known, sometimes unrealistic, bounds on the disturbances and do not consider measurement noise, the aim of this work is to take a step toward relaxing these assumptions and work towards reproducible and robust results on real robotic systems. Therefore, this paper proposed an RMPC design pipeline to guarantee robust constraint satisfaction and recursive feasibility. Furthermore, it verified these properties using a reproducible software stack in a closed-loop simulation of the proposed ROHMPC framework.

5

Robotic Frameworks Development: A Reproducibility Perspective

So far, Chapter 2 introduced the theoretical foundations of trajectory tracking MPC for mobile robots, and Chapters 3 and 4 built on this theory to develop novel HMPC and ROHMPC approaches for safe autonomous motion planning and control. To further strengthen the relevance of this work, the current chapter focuses on a more practical, yet equally important topic: reproducibility.

A fundamental aspect of science is the ability to build on the work of other researchers by first verifying their conclusions. The ability to achieve similar results under comparable conditions is a form of reproducibility. In general, the level of reproducibility is increasingly being questioned, including in the fields of engineering [146], [147] and robotics [148], [149]. Despite its importance for scientific integrity and collaborative progress, researchers often lack incentives to prioritize reproducible outcomes [150], including findable accessible interoperable reusable data [151], even though doing so benefits both the broader community and their own future work [152], [153].

To raise awareness of this issue, this chapter provides a meaningful interpretation of reproducibility (Section 5.1), translates it to the context of robotics research (Section 5.2), describes the frameworks developed for this thesis and evaluates them in terms of reproducibility (Section 5.3), and concludes with a summary of the main findings (Section 5.4).

This chapter is based on and taken in parts literally from:
D. Benders, Reproducibility in robotics 2025. [Online]. Available: https://github.com/dbenders1/reproducibility-in-robotics-2025.

5.1. WHAT IS REPRODUCIBILITY?

There is no universally accepted definition of reproducibility, and confusion often arises around its usage and related terms [154]. As proposed in [155], reproducibility can be categorized into three distinct types:

- "*Method reproducibility* refers to the provision of enough detail about study procedures and data so the same procedures could, in theory or in actuality, be exactly repeated."
- "*Results reproducibility* refers to obtaining the same results from the conduct of an independent study whose procedures are as closely matched to the original experiment as possible."
- "*Inferential reproducibility* refers to the drawing of qualitatively similar conclusions from either an independent replication of a study or a reanalysis of the original study."

Following the definitions in The Turing Way handbook [156], results reproducibility can be further refined into four subtypes:

- *Reproducibility:* "A result is reproducible when the same analysis steps performed on the same dataset consistently produces the same answer."
- *Replicability:* "A result is replicable when the same analysis performed on different datasets produces qualitatively similar answers."
- *Robustness:* "A result is robust when the same dataset is subjected to different analysis workflows to answer the same research question (for example one pipeline written in R and another written in Python) and a qualitatively similar or identical answer is produced. Robust results show that the work is not dependent on the specificities of the programming language chosen to perform the analysis."
- *Generalizability:* "Combining replicable and robust findings allow us to form generalisable results. Note that running an analysis on a different software implementation and with a different dataset does not provide generalised results. There will be many more steps to know how well the work applies to all the different aspects of the research question. Generalisation is an important step towards understanding that the result is not dependent on a particular dataset nor a particular version of the analysis pipeline."

Figure 5.1 provides an overview of the relationships between these types of reproducibility. The interpretation and relevance of each type vary across research domains. The next section offers an interpretation for robotics research.

5.2. REPRODUCIBILITY IN ROBOTICS RESEARCH

As discussed in [157], "robotics research spans a wide variety of application domains and tasks with very different morphologies, dynamics and control approaches and implementations". This diversity makes it essential to clearly define the scope of each study to enable precise scientific statements about specific engineering or robotics questions.

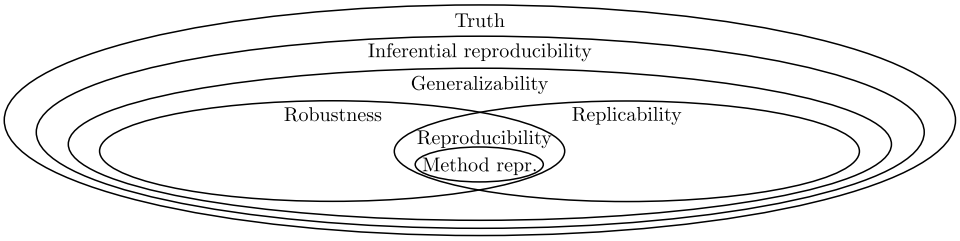


Figure 5.1: Visualization of the different reproducibility sets. The larger the set, the higher the level of reproducibility, ranging from method reproducibility to reproducibility and further to the actual truth. Note that the truth is not reachable, but it will reproduce in the most uniform way. One of the goals of research is to get as close as possible to the truth.

To provide a meaningful interpretation on the different levels of reproducibility in the context of robotics, we must first adapt the concepts of *analysis* and *data*. *Analysis* can be considered as the procedures used to derive valid conclusions from the data. The *data* encompasses all components necessary to execute an experiment, including the *data* recorded during the experiment. To better understand the data aspect in robotics research, we first need to understand the structure of typical robotic systems.

Robotics is an interdisciplinary field involving, but not limited to, machine learning, perception, planning, and control. As a result, robotic systems have become increasingly complex, consisting of numerous interdependent modules that must function cohesively to successfully perform tasks. A typical system architecture is illustrated in Figure 5.2.

The specific modules used vary by application. For instance, a state estimator may rely on the global positioning system (GPS) in outdoor environments, while indoor setups might use LiDAR scanners or depth cameras. Thus, in robotics, data includes not only the proposed algorithm but also the supporting hardware and software modules required to run it. Similarly, analysis refers to the evaluation of algorithm performance, typically using metrics such as goal achievement time, efficiency, and tracking error.

With these definitions, we can distinguish between different types of reproducibility in robotics. The most basic form is *method reproducibility*, which ensures that other researchers can execute the same algorithm without issues. Achieving this requires detailed documentation of the algorithm and its implementation. However, due to publication constraints and the difficulty of remembering every implementation detail during submission writing, a more practical approach is to include hardware specifications (e.g., CPU, GPU, RAM) in the manuscript and publish the software in a public repository.

Despite good intentions, practical experience shows that others often struggle to build, install, and run the provided code. This is a known result in robotics research [122]. To improve the likelihood of method reproducibility, the following recommendations are proposed [149], [152], [156], [158]:

- *Publish the full framework*: Method reproducibility requires that the code can be executed. Therefore, share not only the algorithm-specific code but also the complete framework in which the algorithm operates.

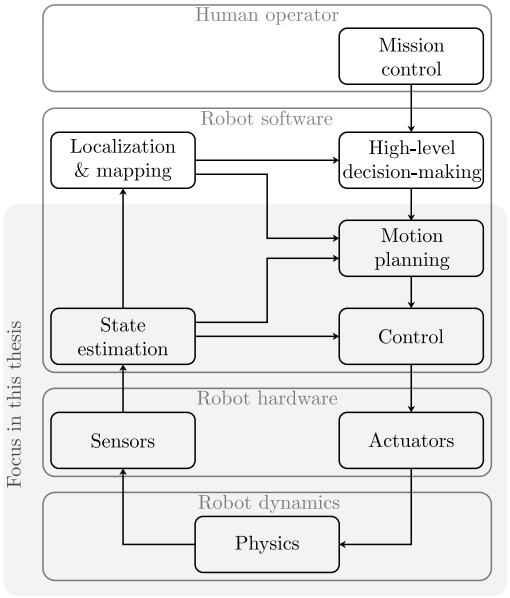


Figure 5.2: Block diagram of a typical robotic system architecture, including the human operator and the robot software, hardware, and physics. The arrows indicate the data flow between the different modules. The work in this thesis focuses on motion planning and the modules below.

- *Use object-oriented programming*: Where feasible, adopt object-oriented design principles [159], [160]. This promotes modularity, facilitates testing and debugging, and supports extensibility.
- *Employ version control*: Small code changes can significantly affect results. Use a version control system like Git from the start of the project, committing incremental updates regularly. This enables easy comparison between versions and efficient bug tracking.
- *Manage dependencies explicitly*: Minor version differences in software packages can lead to inconsistent results. Use package managers such as Conda or Pip to specify and control dependencies. In robotics, ROS packages are commonly used, but lack a dedicated package manager. To address this, include either Git submodules or a *.repos* file to link to specific commits of external packages.
- *Use containerization tools*: To ensure cross-platform compatibility, use containerization tools like Docker or Singularity. These tools encapsulate all dependencies, allowing the code to run on any system with the required hardware and containerization program. Keep containers updated to avoid future compatibility issues and ensure long-term usability.
- *Provide clear documentation*: Include a README file that explains how to install, build, and run the code. This is essential for guiding users through setup and execution.

Method reproducibility is a necessary and sufficient condition for *results reproducibility* (hereafter also referred to as reproducibility) in the deterministic case. In robotics, determinism implies that every execution of the framework yields the same result. However, achieving deterministic behavior is challenging due to several sources of stochasticity commonly present in robotic systems:

- *Hardware variability*: It is unlikely that other researchers will have an identical hardware setup. Differences in CPU architecture and scheduling can lead to timing discrepancies between modules such as the ones depicted in Figure 5.2. Without proper synchronization, these timing differences may alter module outputs or even destabilize the robot.
- *System load and scheduling*: Even with identical hardware, background processes and operating system scheduling can vary. This is especially true for computers not running a strict real-time operating system, such as the commonly used embedded computers, laptops, and desktops in robotics research.
- *Sensor noise*: Sensor measurements inherently contain noise. While simulators and algorithms often assume Gaussian noise generated via random number generators, real-world sensor noise may deviate from this assumption.
- *Random number generators*: These are not only used for simulating sensor noise but also in other modules. Any variation in the seed value can lead to different results.
- *Network effects*: Robotic systems that rely on external digital data sources are subject to network delays and packet loss. These stochastic effects can disrupt data flow and influence system behavior.
- *Environmental variability*: In real-world experiments, environmental conditions such as friction, air density, and lighting vary over runs and locations. These variations impact robot performance.

Due to these sources of stochasticity, each run of a robotic system, whether in simulation or in the real world, may yield different results. Consequently, method reproducibility does not guarantee results reproducibility. To provide meaningful insights to the research community, experiments and simulations are typically repeated multiple times, with outcomes reported using statistical measures such as the mean and standard deviation. Achieving similar statistical outcomes across runs is considered results reproducibility.

A step beyond results reproducibility is *replicability*. This refers to obtaining similar results on different systems by following a comparable algorithmic design. Crucially, these results should also be reproducible by other researchers. Replicability partly depends on the experimental design. In some cases, experiments are tailored to favor the proposed algorithm, which may not generalize to other setups. Therefore, an algorithm that performs consistently across different environments is considered replicable.

Another dimension is *robustness*, which occurs when researchers reach similar conclusions using the same setup but different evaluation metrics. Robustness indicates that the findings are not overly sensitive to the choice of metric.

A further dimension of results quality is *generalizability*, which refers to the extent to which results remain valid across different environments and evaluation metrics. Achieving generalizability often requires more than the combination of replicability or robustness. While generalizable results are desirable, they are difficult to obtain, as no single algorithm typically performs best across all scenarios. In the author's view, striving for generalizability should not be the primary goal of the research community, as it may hinder the exploratory nature of research and limit the development of diverse solutions that benefit society.

In addition to method and results reproducibility, there is inferential reproducibility, which "refers to the drawing of qualitatively similar conclusions from either an independent replication of a study or a reanalysis of the original study" [155]. Inferential reproducibility plays a limited role in robotics because conclusions are often based on quantitative metrics. However, there are aspects such as the preferred algorithm to use in a specific context that may depend on the interpretation of the individual researcher. This can be considered a form of inferential reproducibility.

Ensuring reproducible results in robotics requires substantial investment, even for achieving method reproducibility. Given the inherent complexity of robotic frameworks, the engineering effort involved is significant. As a result, research labs often specialize in a specific subfield of robotics, typically developing their own robotic solution: *a* robot designed to perform *a* task. This leads to a highly diverse set of publications, making it difficult to compare contributions across labs.

Only a limited number of studies focus on benchmark comparisons tailored to robustness, such as evaluating algorithms using standardized metrics [161], [162], or replicability, such as testing algorithm performance across different robot setups [36]. These works are particularly valuable because they provide insights that extend beyond the performance of a single algorithm.

To further improve the interpretability and comparability of results in robotics, the following aspects are essential:

- *Testbed standardization*: Comparing algorithms requires standardized robot testbeds. There exist works proposing simulators [163]–[165], complete software stacks [127], [166], [167], hardware and software stacks [40], [168]–[176], and remotely-accessible experimental setups [177]–[181]. For an overview of available ground and aerial vehicle testbeds, see [182], [183].

- *Benchmark standardization*: To enhance benchmarking of robotic setups, guidelines such as those from the European Robotics Network [184] provide structured analysis methods. Additionally, datasets with increased standardization [185], benchmarking studies comparing similar algorithms using consistent metrics [186], and benchmarking elements in robotic competitions [187] contribute to this effort. Furthermore, initiatives like [188] aim to improve reproducibility for non-expert users, bridging the gap between research and industry.

5.3. ROBOTIC FRAMEWORKS IN THIS THESIS AND THEIR REPRODUCIBILITY ANALYSIS

Building on the definitions of reproducibility in the context of robotics research, this section presents a detailed analysis of the reproducibility level of the work conducted in this thesis. Specifically, Sections 5.3.1 and 5.3.2 provide an overview of the robotic frameworks developed for the research presented in Chapters 3 and 4, respectively. These sections reflect on the reproducibility of the implemented systems and discuss potential improvements to enhance reproducibility in future work.

5.3.1. HOVERGAMES FRAMEWORK

The research presented in Chapter 3 was conducted using the self-developed HoverGames framework. The hardware foundation of this framework is based on the NXP HoverGames competition kit [128]. For this work, the original NXP flight controller was replaced with a Pixhawk 6X mini board, which provided more stable flight performance by eliminating CPU overload issues. Additionally, an NVIDIA Jetson Xavier NX embedded computer and a RealSense D455 camera were integrated into the platform.

This section focuses on the software stack running on the embedded computer and the flight controller, which are the most relevant components for evaluating reproducibility.

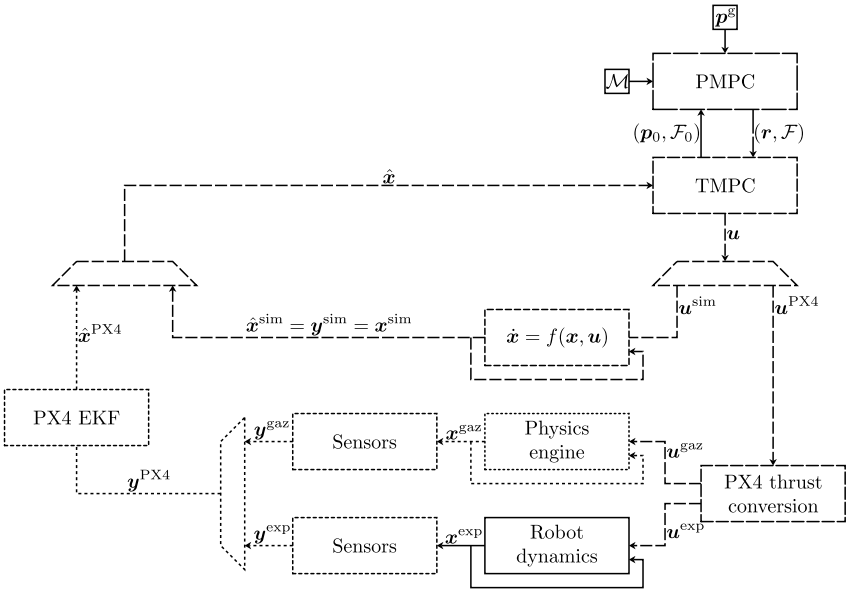


Figure 5.3: Overview of the relevant modules in the HoverGames framework used to obtain the research results in Chapter 3. Blocks represent software modules or physical processes; arrows indicate data flows. Multiplexers and de-multiplexers represent conditional data routing based on whether PX4 infrastructure is used. Solid lines denote continuous modules or data flows; dashed lines indicate data flow frequencies (2500, 250, 100, 20, and 2 Hz, respectively). All modules operate asynchronously.

Figure 5.3 presents a detailed block diagram of the HoverGames framework. It expands on the general architecture shown in Figure 5.2, with emphasis on the motion planning and control loop, as well as the module and data flow frequencies. The actuator block is excluded, as it is handled by the low-level PX4 infrastructure—including attitude and rate controllers, mixers, and the physical robot or simulation environment.

The corresponding code repository, available at https://github.com/dbenders1/hmpc, includes visual and textual documentation of the framework and provides relevant links to the PX4 documentation.

To implement the modules within the HoverGames framework, various software packages were used, most of which are built on the ROS Noetic middleware. The code repository provides an overview of all included packages. Below, the most relevant packages are described:

- *Occupancy grid mapping*: The occupancy grid map, denoted by $\mathscr{M}$ (as in Chapter 3), is generated using the custom `occupancygrid_creator` package. It constructs the map from static and dynamic obstacle data, allowing physical obstacles to be repositioned during flight for flexible testing. This package was mainly developed by Thijs Niesten and is available at https://github.com/R2CLab/occupancygrid_creator.

- *MPC implementation*: The PMPC and TMPC schemes are implemented using the modular `mpc` package available at https://github.com/dbenders1/mpc, developed in collaboration with Thijs Niesten and based on initial work by Dr. Oscar de Groot. It supports flexible MPC configurations and is used in both Chapters 3 and 4. Solver generation is handled in Python via the Forces Pro server, while runtime code is written in C++ for real-time performance. The C++ system interface is implemented using object-oriented design, with a base class for shared functionality and derived classes for PX4, Gazebo, and simple simulation interfaces. As described in Chapter 3, the PMPC includes obstacle avoidance constraints generation. The `mpc` package includes an interface to the `DecompUtil` package [67], which is used to generate single ellipsoid-based convex obstacle-free regions. The `DecompUtil` package is adjusted giving improved onboard execution efficiency. It is available at https://github.com/tud-amr/DecompUtil/tree/drone/dennis-paper-hmpc.

- *Offline MPC computations*: The `mpc-sdp` package, written in MATLAB, handles offline computations of constraint tightening, terminal set scaling, and terminal cost. It is included as a submodule in the `mpc` package for compatibility and available at https://github.com/dbenders1/mpc-sdp.

- *MPC model identification*: The `hovergames_mpc_identification` package identifies parameters of the nonlinear model used in PMPC and TMPC, including gains and time constants. It uses PX4 to send attitude commands and logs responses for parameter estimation. This package is part of the `drone_toolbox` repository, which also includes tools for PX4 control, RViz integration, safety mechanisms, and simulation assets: https://github.com/cor-drone-dev/drone_toolbox.

- *PX4 thrust conversion*: PX4 requires attitude commands with normalized thrust values to support multiple aerial platforms. Therefore, the TMPC thrust output is converted using hover thrust values from PX4's messaging system. Custom implementations of `PX4-Autopilot`, `mavlink-gbp-release`, and `mavros` handle corresponding message conversion and communication and are available at https://github.com/cor-drone-dev/PX4-Autopilot, https://github.com/cor-drone-dev/mavlink-gbp-release, and https://github.com/cor-drone-dev/mavros, respectively.

- *Controller template*: To improve the reusability of the developed quadrotor framework, an additional template repository has been created, called `drone_toolbox_ext_control_template`. It demonstrates how to implement a custom controller that integrates with the PX4 control interface provided by the `drone_toolbox` package. This template supports sending various types of commands to the PX4 autopilot and can be used as a starting point for similar projects. The repository is publicly available at https://github.com/cor-drone-dev/drone_toolbox_ext_control_template.

- *Simple simulation*: The `simple_sim` package provides a lightweight simulation environment for testing PMPC and TMPC implementations without the full PX4 stack. It enables controlled testing with zero model mismatch and is available at https://github.com/dbenders1/simple_sim.

- *Vicon integration*: The `vicon_bridge` package provides the ROS interface to the Vicon motion capture system used in the lab. It was adapted for the Mobile Robotics Lab [189] setup in collaboration with Thijs Niesten and Dr. Sihao Sun and is available at https://github.com/cor-mobile-robotics/vicon_bridge.

REPRODUCIBILITY ANALYSIS

The HoverGames framework was initially developed with the goal of creating a reusable research platform for the lab. Early efforts focused on writing well-structured software packages and providing documentation. Once the framework reached a level of maturity sufficient to produce publishable results, specifically those presented in Chapter 3, the focus shifted toward improving its reproducibility.

To support this goal, the framework adopted the *Don't Repeat Yourself* principle [190] to reduce code duplication and improve maintainability. In addition, many of the reproducibility recommendations outlined in Section 5.2 were followed. These include writing object-oriented code, using version control from the beginning of the project, managing dependencies with a package manager, applying containerization to ensure compatibility across systems, and providing multiple README files that explain how to install, build, and run the software stacks.

Despite these efforts, reproducibility was not the primary focus at the start of the project, and several aspects can still be improved:

- From a control engineering perspective, the state cannot be perfectly measured due to the presence of non-deterministic measurement noise. This leads to degraded performance and state variability across different runs.

- From both control and software engineering perspectives, differences in execution flow across controllers and estimators occur due to the asynchronous nature of the framework. These differences result in performance variations between runs and across setups, including lab experiments, Gazebo simulations, and simple simulations.
- From a software engineering perspective, the framework is non-deterministic. It includes human inputs and random number generators, which also contribute to performance variability across runs and setups.

Given the extensive documentation and public availability of the HoverGames framework, and the fact that parts of it have been tested by other researchers, the framework satisfies *method reproducibility*. Although a formal reproducibility analysis was not conducted, the conclusions in Chapter 3 were empirically validated through repeated experiments and simulations during internal review and rebuttal processes. Only specific metrics, such as goal-reaching times, showed slight deviations due to the non-deterministic nature of the framework. Therefore, the framework approaches but does not fully achieve results reproducibility.

Although results reproducibility is not fully satisfied, components of the framework have already been successfully reused in other research projects. For example, the real-time obstacle avoidance constraint generation code in `mpc` and `DecompUtil` was deployed on a ground robot in [191]. The `mpc` package and an early Python version of `simple_sim` were used to generate results in [44]. Additionally, several packages, including `occupancygrid_creator`, `mpc`, `mpc-sdp`, and `simple_sim`, required only minor adjustments for integration into the framework described in the next section. These examples demonstrate the versatility of the HoverGames framework and its potential for use across various mobile robot platforms.

5.3.2. FALCON FRAMEWORK

By the time the work in Chapter 3 was completed, several high-performing open-source quadrotor stacks had become publicly available. Among these was the Agilicious stack [40]. In line with the philosophy of building on existing work and fostering collaboration within the research group, the Falcon framework was developed based on Agilicious and used for the research presented in Chapter 4. This work placed a strong emphasis on results reproducibility and was conducted using simulations on a Dell XPS 15 laptop.

Figure 5.4 presents a block diagram of the Falcon framework. Like Figure 5.3, it highlights the motion planning and control loop, along with the module and data flow frequencies. Unlike the HoverGames framework, the Falcon framework includes additional components such as a robust control law and an observer to explicitly handle disturbances and measurement noise. It also supports two simulation setups: one using Agilicious-based Gazebo simulation with full control over measurement noise, and another using a simple simulation with control over both disturbances and measurement noise. The corresponding code is publicly available at `https://github.com/dbenders1/rohmpc`.

As with the HoverGames framework, various software packages were used. In this case, most packages are built on the ROS Melodic middleware. The most relevant pack-

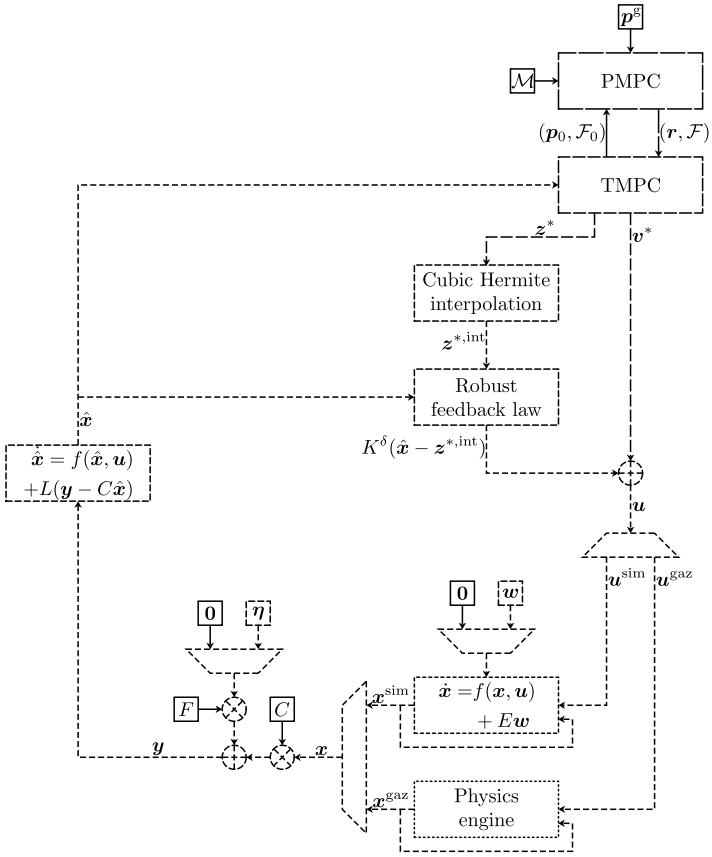


Figure 5.4: Overview of the relevant modules in the Falcon framework used to obtain the research results in Chapter 4. Blocks and arrows have similar meanings to those in Figure 5.3. Data frequencies are approximately 500000, 2000, 500, 100, and 10 Hz. Unlike the HoverGames framework, modules in this setup run synchronously.

ages are listed below:

- *Agiclean*: The Agilicious stack (`agilicious`) provides a modular and high-performance estimation and control framework for quadrotor racing. It is publicly available at https://github.com/uzh-rpg/agilicious and requires a license. Based on this stack, Dr. Sihao Sun developed an updated version called `agiclean`, which adds support for the Falcon quadrotor. This version is available at https://github.com/tud-amr/agiclean and can be accessed under the same license with approval from Sihao Sun. In this project, `agiclean` was further optimized and extended to integrate the `mpc` package for deterministic execution.

- *Gazebo simulation*: The `agiclean` package uses the `rotors_simulator` package [141] for quadrotor simulation in Gazebo. This code was modified to ensure deter-

ministic behavior and is available at https://github.com/tud-amr/rotors_simulator/tree/dennis-paper-rohmpc.

- *Reused packages*: Several packages from the HoverGames framework were reused with minor modifications. These include `occupancygrid_creator`, `mpc`, `DecompUtil`, `mpc-sdp`, and `simple_sim`. Adjustments were primarily related to model changes and robust control design.

REPRODUCIBILITY ANALYSIS

Drawing on the experience gained from developing the HoverGames framework, the Falcon framework was designed with a clear focus on achieving results reproducibility. Several improvements were introduced beyond those used in the HoverGames project:

- A consistent interface was used across all setups, including Gazebo and simple simulation, ensuring identical inputs and outputs.
- The same method was applied to generate measurement noise in every setup.
- Random number generators were initialized using fixed seeds.
- Human inputs were avoided entirely.
- All asynchronous processes were synchronized, even though this introduced additional delay. This included synchronizing ROS node startups and middleware communication between ROS and Gazebo. Modules were configured to run only after receiving their required inputs, resulting in a consistent execution graph across runs. More details on execution graphs are provided in [192].
- All steps required to reproduce the research results in Chapter 4 are documented in README files within the repository.

One aspect that was not explicitly included in the workflow was the use of unit and integration tests. Writing such tests is time-consuming and requires frequent updates to reflect changes in architecture and design. Instead, a pragmatic approach was adopted: simulations were run while printing inputs and outputs at each control loop iteration, and logs from different runs were compared. If the logs matched exactly, the code was considered deterministic. This method was applied to both the Gazebo and simple simulation setups.

Based on these practices, the Falcon framework satisfies *results reproducibility*.

5.4. CONCLUSIONS

To advance human knowledge, it is essential that researchers can reproduce and build upon the results presented in scientific publications. In robotics, reproducibility is increasingly questioned due to the complexity of modern robotic systems. Successfully conducting experiments with a robot often requires expertise across multiple disciplines, including perception, planning, control, and software engineering. As a result, achieving method reproducibility, let alone results reproducibility or generalizability,

is both time-consuming and technically challenging. Nevertheless, reproducibility is a fundamental requirement for robotics to be considered a rigorous scientific field.

Many researchers in robotics acknowledge this challenge and actively contribute to improving reproducibility through efforts such as testbed and benchmark standardization. To support reproducibility and provide context for interpreting the results presented in this thesis, Sections 5.3.1 and 5.3.2 offer detailed analyses of the reproducibility level of the robotic frameworks used in Chapters 3 and 4, respectively.

It can be concluded that the HoverGames framework satisfies method reproducibility and makes significant progress toward results reproducibility, although it does not fully achieve it. In contrast, the Falcon framework was developed with reproducibility as a central goal and successfully satisfies both method and results reproducibility, supported by the lessons learned from the HoverGames project.

6

Conclusions and Future Work

This final chapter summarizes the conclusions drawn from the work in this thesis, organised per contribution in Section 6.1. Based on these conclusions and other insights obtained during the project, Section 6.2 provides a discussion on the results and an outlook on corresponding future work directions to advance the deployment of safe mobile robots.

6.1. CONCLUSIONS

This thesis was written as part of a collaborative project between the Delft University of Technology and Dutch National Police. The overarching objective of this project was to design robotic solutions that support safe exploration of complex environments, such as the drug lab scenario introduced in Section 1. The specific focus of this research was the development of autonomous navigation algorithms for mobile robots, grounded in MPC theory. The main conclusions drawn from this work are presented in the following sections, structured according to the individual contributions made throughout the thesis.

6.1.1. STEP-BY-STEP GUIDE ON NMPC FOR SAFE MOBILE ROBOT NAVIGATION

Recent developments in NMPC within the control community have demonstrated promising potential for ensuring safety in the presence of model uncertainties. Despite these advances, the adoption of such methods in robotics remains limited. Discussions with colleagues in the field suggest that one of the key barriers is the need for a deep and balanced understanding of both theoretical foundations and practical implementation. Even minor inaccuracies in either aspect can compromise the guarantees of a safe MPC scheme.

To address this gap, Chapter 2 introduced a step-by-step guide on NMPC tailored for safe mobile robot navigation. The guide begins with fundamental concepts of MPC and safety, and progressively explores more advanced topics, including RMPC and ROMPC. It is designed as a practical resource for researchers and practitioners in robotics, equipping them with the essential knowledge required to implement safe NMPC strategies in real-world applications.

The guide has already shown positive impact during the course of this research. For instance, it facilitated technical discussions related to the work presented in Chapter 4, and has been actively used by a collaborator working on CRAMPC in the context of ASVs [46].

6.1.2. EMBEDDED HMPC FOR AUTONOMOUS NAVIGATION

The work presented in Chapter 3 marks the first set of validated results in this thesis. It was motivated by the observation that conventional MPC schemes face several limitations, including high computational demands and the risk of producing infeasible solutions. To address these challenges, the chapter introduced a novel HMPC framework. This framework integrates a slow PMPC for long-term trajectory generation with a fast TMPC for accurate tracking.

The PMPC operates independently of the TMPC during runtime, enabled by an offline co-design process involving terminal ingredients from the TMPC. It continuously updates the reference trajectory, allowing the system to plan toward a steady state without compromising progress. Given the assumption of a static environment, the PMPC offers recursive feasibility guarantees. Furthermore, through the design of appropriate terminal ingredients, it can be shown that the TMPC, and, when extended with a suitably-designed PMPC, the entire HMPC scheme, is recursively feasible and capable

of avoiding collisions in the absence of model uncertainty.

The resulting HMPC scheme demonstrated safer and approximately four times faster goal-reaching performance compared to a standard SMPC approach with similar characteristics. It successfully operated in real time on an NVIDIA Jetson Xavier NX embedded computer, whereas the SMPC failed to meet real-time constraints. Additionally, the HMPC produced smoother trajectories and exhibited reduced sensitivity to model inaccuracies. These conclusions are supported by results from simulations without model uncertainty, Gazebo simulations, and lab experiments.

A key limitation of this work is that the theoretical guarantees do not extend to scenarios involving model uncertainty. This limitation serves as the primary motivation for the next contribution.

6.1.3. A PRACTICAL PIPELINE FOR ROBUST MPC SYNTHESIS

Building on the conclusions from the previous chapter, the work presented in Chapter 4 aimed to develop an accessible pipeline that enables robotics practitioners to explore ROMPC schemes. This pipeline begins with the identification of model uncertainty using IO data from a sufficiently excited nonlinear robotic system.

As anticipated, ambiguity arises when attempting to distinguish between additive disturbances and measurement noise, since both can account for similar output behavior. To address this, the pipeline adopts the least restrictive assumptions, relying on reasonable measurement noise bounds derived from sensor datasheets. These bounds are incorporated as constraints in an MHE-like formulation, which is used to identify model uncertainty.

The identified uncertainty bounds then serve as inputs to the next offline step: computing a robust feedback control law and corresponding constraint tightening constants using contraction metrics. The results indicate that this computation is nontrivial and becomes increasingly challenging as the model uncertainty bounds grow. In practice, small uncertainty bounds are required to ensure feasibility of the resulting online MPC scheme.

Finally, the chapter demonstrates that, given appropriate model uncertainty bounds and corresponding robust ingredients, the proposed ROHMPC scheme guarantees recursive feasibility and obstacle avoidance in the presence of model uncertainty. These conclusions are supported by reproducible simulation results.

6.1.4. REPRODUCIBILITY IN ROBOTICS RESEARCH

As an additional contribution, Chapter 5 provided a well-supported argument for the importance of reproducibility in robotics research. The chapter discussed several challenges that hinder reproducibility, including the inherent complexity of robotic systems, the absence of standardized testing environments, and the practical difficulties in sharing code and data.

To address these issues, the chapter proposed a set of guidelines informed by both existing literature and personal experience. These guidelines aim to support researchers in making their work more transparent and reproducible. In line with this objective, the chapter also offered a detailed analysis of the robotic hardware and software stacks used to produce the results presented in Chapters 3 and 4. Particular attention was given

to evaluating the level of reproducibility of these works, with the intention of enabling others to reproduce and build upon the presented results.

6.2. FUTURE WORK

Although this thesis has taken several steps toward bridging the gap between safe MPC theory and its application in robotics, advancing autonomous and safe mobile robot navigation algorithms, and reflecting on reproducibility in robotics research, many challenges remain. The following sections provide a discussion of the results obtained and outline potential directions for future work that could further support the deployment of safe mobile robotic systems.

6.2.1. IMPROVING KNOWLEDGE TRANSFER

Although the step-by-step guide in Chapter 2 provides a solid foundation for understanding safe NMPC, there remains room for improvement in its accessibility and educational value. As a preliminary test, the guide was shared with several research engineers in the department. Feedback indicated that, without prior exposure to MPC fundamentals, some concepts were difficult to grasp. To enhance reader understanding, future iterations of the guide could benefit from the inclusion of visual aids, such as figures illustrating the concept of linear MPC constraints, which would help contextualize the theoretical material and make it more approachable for a broader public.

6.2.2. EFFICIENT GOAL-REACHING USING MPC

Although the PMPC scheme introduced in Chapter 3 enables long-term goal-oriented planning, its current formulation may lead to inefficient behavior in certain scenarios. For instance, when a goal lies behind a rectangular obstacle and the corresponding halfspace constraint is orthogonal to the direct path, the robot may become stuck due to the lack of incentive to navigate around the obstacle. To address this, future work could explore integrating a globally optimized path, such as a spline generated using methods like [193], into the PMPC framework. This would allow the use of model predictive contouring control [93], [100] or trajectory tracking formulations [38], particularly when the spline is time-parameterized [99]. Given that the PMPC computation time is well below the real-time threshold, such enhancements are feasible.

A second factor influencing goal-reaching efficiency is the shape of the obstacle avoidance constraints. In this work, these constraints are implemented as stage-wise polygonal regions based on previously computed plans, inspired by [67]. The orientation of each linear halfspace constraint depends on the direction of the straight line between discrete stages in the planning horizon. Incorporating a continuously parameterized global path and leveraging recent methods such as [68] could help generate smoother avoidance constraints and improve navigation performance. However, such approaches may not be suitable for all scenarios. In the context of unknown environment exploration, for example, global planning is of limited use. In such cases, planning an optimistic trajectory into unexplored space, along with a contingency plan for fallback tracking [19], may offer a more practical solution.

6.2.3. EXTENSION TO 3D

Although the quadrotors used in Chapter 3 and Chapter 4 operate in 3D space, the obstacle avoidance constraints were limited to two dimensions for simplicity of implementation and illustration. While the underlying generation scheme from [67] supports 3D environments, extending the current implementation to full 3D would introduce additional computational complexity and require incorporating the vertical component into the constraint tightening scheme. Addressing this limitation is a relevant direction for future work, particularly for researchers working with micro aerial vehicles.

6.2.4. APPLICATION IN REAL-WORLD OUTDOOR SCENARIOS

While HMPC has been verified through lab experiments, ROHMPC has only been validated in a simplified Gazebo simulation. To reduce model mismatch before integrating the tightening calibration into the RMPC pipeline, the simulation model was intentionally simplified by excluding effects such as motor spin-up and spin-down dynamics and rotor drag. Although this simplification helped isolate the core behavior of the control scheme, it limits the realism of the validation. Due to time constraints, the final results could not be tested against a more representative model. A relevant next step is to complete this validation using a more accurate simulation and experimental setup that better reflect real-world dynamics.

Another important assumption concerns the availability of perfect pose estimates. In the HMPC lab experiments, accurate positioning was achieved using a motion capture system. However, outdoor deployments typically rely on onboard sensors such as inertial measurement units, cameras, and LiDAR scanners. These sensors are subject to noise, drift, and latency, which can significantly affect state estimation and, consequently, the robot's performance. Investigating the impact of such uncertainties and developing robust estimation and control strategies to mitigate their effects is a relevant direction for future research, especially for autonomous systems operating in unstructured or GPS-denied environments.

A third limitation is the assumption of a static and known environment, which does not reflect the dynamic nature of many real-world settings. The presence of moving obstacles and other agents can render the proposed schemes infeasible. Addressing this limitation requires extending the framework to handle dynamic environments, potentially by incorporating probabilistic models or reactive planning strategies. For a more detailed discussion on MPC in dynamic environments, the reader is referred to [194].

6.2.5. REDUCING CONSERVATISM

The results in Chapter 4 demonstrate that the ROHMPC scheme can guarantee recursive feasibility and obstacle avoidance in the presence of model uncertainty. However, the scheme tends to be conservative, requiring small model uncertainty bounds to ensure feasibility. Reducing this conservatism is an important direction for future work.

One approach is to reduce the model uncertainty bounds themselves. This can be achieved either by improving the accuracy of the dynamical model or by considering state- and input-dependent uncertainty bounds. The former could be addressed using robust adaptive MPC strategies that leverage set-membership estimation techniques

[62]. The latter builds on the approach described in Chapter 4, with the added flexibility of treating the model uncertainty bound and tube size as decision variables within the optimization problem. This allows the controller to act more cautiously in regions with higher expected disturbances, while enabling more aggressive behavior in regions with lower uncertainty [61].

An alternative strategy is to replace hard constraints with probabilistic ones. Risk-aware control methods, such as the one proposed in [195], allow for optimizing performance while maintaining safety with a specified probability. If the underlying uncertainty distribution can be estimated, for example using GPs, this information can be incorporated into the MPC formulation. The level of acceptable risk can be tuned by the user and should be chosen based on the specific application context [16].

6.2.6. THE POTENTIAL OF ARTIFICIAL INTELLIGENCE

This thesis focused on the understanding and applicability of MPC-based algorithms for safe mobile robot navigation. While MPC provides a structured framework for enforcing safety guarantees, it has two main limitations: a fixed objective formulation and significant computational cost. Artificial intelligence (AI)-based methods, often referred to as learning-based or data-driven approaches, have shown potential to reduce computation time and improve performance. For example, reinforcement learning [196] can be used to train neural network policies, with MPC guiding the policy search [197]. However, these trained policies typically lack formal safety guarantees [198], [199].

To address this, recent work has explored ways to retain safety while leveraging learning. According to [200], three main strategies include: (a) improving the MPC prediction model using data [75], (b) refining the objective or constraints [201], and (c) combining learning-based policies with safe fallback mechanisms such as MPC [202], extended with CBFs [203], or using HJ reachability analysis [204]. MPC can also serve as an expert in supervised learning [205]. Finally, diffusion-based approximate MPC can address the limitation of a fixed objective and capture multi-modalities [206].

Given the promise of these approaches, future work could explore integrating the proposed MPC schemes with AI methods to handle complex objectives and enhance performance while ensuring safety.

6.2.7. WHAT ALGORITHM TO CHOOSE?

A central motivation in robotics research is to equip practitioners with tools that enable robotic systems to perform reliably in real-world tasks. The field offers a wide range of algorithmic approaches, each with its own strengths and limitations. In addition to the AI-based methods discussed previously, there are also other promising approaches that are more closely related to the work presented in this thesis. These include techniques described in Section 3.1.1, which share a focus on model-based control and/or formal safety guarantees. This diversity of methods raises an important question: *What algorithm should be chosen for a specific application?*

To answer this question meaningfully, it is essential to compare algorithms in a standardized and reproducible manner. Such comparisons should not only consider experimental design but also the hardware and software stacks used to run the experiments. As discussed in Chapter 5, reproducibility in robotics research remains a challenge. For-

tunately, the increasing availability of open-source platforms, such as [40], [168], and [173], offers an opportunity for the community to converge on a few common frameworks. This convergence would facilitate fair and transparent comparison studies, like those in [143] and [161], ultimately supporting more informed decision-making when selecting algorithms for specific robotic applications.

6.2.8. DOES MY ROBOT SOLUTION MAKE SENSE AT ALL?

Related to the previous point, it is important for every roboticist, whether researcher or practitioner, to recognize that their development efforts extend beyond technical impact. Robotic solutions inevitably influence society, whether in the intended way or not. This places an active responsibility on developers to critically reflect on the broader implications of their work. The questions surrounding this responsibility are complex and often evolve over time, requiring continuous re-evaluation through thoughtful interactions with relevant stakeholders.

Some key questions worth considering include: *Is a robot suitable for this operation? Do I want to develop a robot for this operation? Who else has an interest in this operation? Do they also support the use of a robot in this context?* And if so: *What kind of robotic solution would be acceptable to those stakeholders?* These questions highlight the importance of aligning technical development with societal values and expectations, ensuring that robotic systems are not only effective but also responsible and contextually appropriate.

6.2.9. RELEVANCE OF THIS THESIS IN THE CONTEXT OF POLICE OPERATIONS

The overarching motivation for the research presented in this thesis is to support the safe exploration of unknown and potentially hazardous environments, such as the drug lab scenario introduced in Chapter 1. A central question therefore arises: *How do the contributions of this thesis support practical police operations?*

Addressing this question requires consideration of several dimensions: the technical relevance of the proposed algorithms, the feasibility of deploying these algorithms in real operational settings, and the broader ethical, legal, and societal implications associated with the use of robotic systems in law-enforcement contexts.

From a technical perspective, this thesis has focused on developing MPC-based algorithms for safe autonomous navigation. The extensive body of literature across top robotics conferences and journals demonstrating the effectiveness of MPC for autonomous systems reflects the maturity and relevance of this methodology for safety-critical operations. The research in this thesis specifically focused on NMPC, i.e., MPC leveraging a nonlinear system model. A nonlinear system model can be used to represent a wide range of robotic platforms, including aerial vehicles, as demonstrated in this thesis, ground vehicles, and vessels, as presented in [45, Chap. 6]. Consequently, the developed methods have the potential to support various types of police interventions.

These algorithms, however, rest on two core assumptions: a static environment and incremental stabilizability of the system. The former means that the theoretical guarantees of recursive feasibility do not necessarily hold when the environment is dynamic. In

such situations, the robot may be unable to ensure safety, making robustness to dynamic environments an important future research direction, see Section 6.2.4. The incremental stabilizability assumption is satisfied by many robotic platforms but may impose constraints, for example, requiring a minimum velocity for certain ground vehicles. Thus, while highly applicable across many scenarios, the algorithms are not universally suitable under all conditions.

Regarding practical feasibility, the experimental results demonstrate that the proposed HMPC scheme can run in real time on an embedded computer. This is a key requirement for operational deployment, where onboard computation is often the only viable option. The robust extension, ROHMPC, has so far only been validated in a simplified simulation environment. Although it introduces additional constraint tightening, making it somewhat more computationally demanding, it is still expected to remain within real-time limits. Nonetheless, further validation is needed before its real-world applicability can be assessed. Moreover, as noted in Section 6.2.4, transitioning to exclusively onboard sensing for state estimation represents another important step toward reliable deployment in field operations. Thus, these results suggest that the proposed algorithms have realistic potential for deployment in police scenarios, even though further development and testing are required.

Finally, the deployment of robotic systems in police operations raises important ethical, legal, and societal considerations. Ethically, questions arise concerning safety, which is partly addressed in this thesis, but also fairness, accountability, and transparency of robotic decision-making. Legally, issues of liability and privacy are important: if a robot causes harm, *who is responsible*? *And how can privacy be protected when robots gather data in sensitive environments?* Societally, public acceptance is a key concern, with underlying concerns about safety, misuse in law enforcement, and potential job changes or replacements.

In the author's view, addressing these challenges requires a collaborative, multidisciplinary approach grounded in continuous and open dialogue. Establishing direct and sustained links between developers, end-users, and both internal and external policymakers is a practical and effective starting point. These links can be in the form of regular meetings, but also in the form of workshops, to enhance the mutual understanding of the technical possibilities and limitations, the practical needs and constraints, and the ethical, legal, and societal implications. This mutual understanding is essential for ensuring that the development of robotic solutions is aligned with the needs and values of all stakeholders, thereby supporting the responsible and effective deployment of robotic systems in police operations.

A

Proofs Related to Chapter 2

This appendix contains all mathematical proofs required to show the properties of the MPC schemes introduced in Chapter 2. Each proof is presented in a separate section, and the sections are ordered according to the order in which the proofs are referenced in Chapter 2.

A

A.1. PROOF OF TERMINAL SET (2.31) INVARIANCE

In words, (2.31) tries to find the largest value for α such that system constraints (2.2) are satisfied for all states satisfying terminal set constraint (2.14). Let us define the vector pointing from reference state $\boldsymbol{x}^{\mathrm{r}}$ to state $\boldsymbol{x}$ by

$$\boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{x}^{\mathrm{r}}. \tag{A.1}$$

Terminal set constraint (2.14) can now be rewritten as

$$\|\boldsymbol{\delta}\|^2_{P(\boldsymbol{r})} = \boldsymbol{\delta}^\top P(\boldsymbol{r})\boldsymbol{\delta} \leq \alpha^2. \tag{A.2}$$

Furthermore, let us define the error in the control input by

$$\boldsymbol{u} - \boldsymbol{u}^{\mathrm{r}} \overset{(2.12)}{=} K(\boldsymbol{r})(\boldsymbol{x} - \boldsymbol{x}^{\mathrm{r}}) = K(\boldsymbol{r})\boldsymbol{\delta}. \tag{A.3}$$

Thus, in the terminal set, it should hold that

$$L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} \boldsymbol{\delta} \leq l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{A.4}$$

Since the terminal set is described by (A.2), we would like the following implication to hold to satisfy (2.15b) in Assumption 1:

$$\boldsymbol{\delta}^\top P(\boldsymbol{r})\boldsymbol{\delta} \leq \alpha^2 \Rightarrow L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} \boldsymbol{\delta} \leq l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}, \ \forall \boldsymbol{r} \in \bar{\mathcal{Z}}, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{A.5}$$

By defining $\tilde{\boldsymbol{\delta}} := P(\boldsymbol{r})^{\frac{1}{2}} \boldsymbol{\delta}$ we obtain

$$\tilde{\boldsymbol{\delta}}^\top \tilde{\boldsymbol{\delta}} \leq \alpha^2 \Rightarrow L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \tilde{\boldsymbol{\delta}} \leq l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}, \ \forall \boldsymbol{r} \in \bar{\mathcal{Z}}, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{A.6}$$

Squaring and taking the norm of the left side of the implication gives

$$\tilde{\boldsymbol{\delta}}^\top \tilde{\boldsymbol{\delta}} \leq \alpha^2 \Rightarrow \|L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \tilde{\boldsymbol{\delta}}\|^2 \leq \|l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}} \boldsymbol{r}\|^2, \ \forall \boldsymbol{r} \in \bar{\mathcal{Z}}, \ j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}. \tag{A.7}$$

We can now write

$$\|L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \tilde{\boldsymbol{\delta}}\|^2 \overset{(a)}{\leq} \|L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \|^2 \|\tilde{\boldsymbol{\delta}}\|^2 \tag{A.8a}$$

$$\overset{(b)}{=} \|L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \|^2 \tilde{\boldsymbol{\delta}}^\top \tilde{\boldsymbol{\delta}} \tag{A.8b}$$

$$\overset{(A.2)}{\leq} \|L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}} \|^2 \alpha^2 \tag{A.8c}$$

$$\overset{(c)}{=} \|\big(L_j^{\mathrm{r,s}} \begin{bmatrix} K(\boldsymbol{r}) \\ I^{n^{\mathrm{x}}} \end{bmatrix} P(\boldsymbol{r})^{-\frac{1}{2}}\big)^\top \|^2 \alpha^2 \tag{A.8d}$$

$$\overset{(d)}{=} \|P(\boldsymbol{r})^{-\frac{1}{2}} \big[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}\top}\big] L_j^{\mathrm{r,s}\top} \|^2 \alpha^2, \tag{A.8e}$$

where (a) follows from the Cauchy-Schwarz inequality, (b) follows from the definition of the inner product, (c) follows from the fact that the norm of a matrix is equal to the norm of its transpose, and (d) follows from the definition of the transpose and the fact that $P(\boldsymbol{r})$ is symmetric. Thus, the following implication holds:

$$\|P(\boldsymbol{r})^{-\frac{1}{2}}\left[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}\top}\right]L_j^{\mathrm{r,s}\top}\|^2\alpha^2 \le (l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}}\boldsymbol{r})^2 \Rightarrow \|L_j^{\mathrm{r,s}}\begin{bmatrix}K(\boldsymbol{r})\\ I^{n^{\mathrm{x}}}\end{bmatrix}P(\boldsymbol{r})^{-\frac{1}{2}}\tilde{\boldsymbol{\delta}}\|^2 \le (l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}}\boldsymbol{r})^2. \tag{A.9}$$

Taking the square root of both sides gives

$$\|P(\boldsymbol{r})^{-\frac{1}{2}}\left[K(\boldsymbol{r})^\top I^{n^{\mathrm{x}}\top}\right]L_j^{\mathrm{r,s}\top}\|\alpha \le (l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}}\boldsymbol{r}) \Rightarrow \|L_j^{\mathrm{r,s}}\begin{bmatrix}K(\boldsymbol{r})\\ I^{n^{\mathrm{x}}}\end{bmatrix}P(\boldsymbol{r})^{-\frac{1}{2}}\tilde{\boldsymbol{\delta}}\| \le (l_j^{\mathrm{s}} - L_j^{\mathrm{r,s}}\boldsymbol{r}). \tag{A.10}$$

Therefore, the maximum value for α such that the system constraints are satisfied for all states satisfying terminal set constraint (2.14) is given by solving linear program (2.31). □

A.2. PROOF OF TUBE SIZE UPPER BOUND (2.68)

Intuitively, (2.68) says that the upper bound on the candidate tube size starts $s_{T^{\mathrm{s}}}$ lower than $s_{T^{\mathrm{s}}+\tau}$ and contracts exponentially towards $s_{T^{\mathrm{s}}+\tau}$ with factor $e^{-\rho\tau}$ from there. Thus, the upper bound on the candidate tube size contracts with factor ρ to the previously optimal tube size. This makes sense since the tube dynamics (2.42e) saturate exponentially to a maximum value of $\frac{\bar{w}}{\rho}$.

Let us prove this by induction. At $\tau = 0$, (2.68) is trivially satisfied. Given that (2.68) holds at $\tau = 0$, we can prove that it also holds for $\tau > 0$:

$$\frac{\partial}{\partial\tau}(s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}}) \overset{(2.42e)}{=} -\rho s_\tau + \bar{w} + \rho s_{T^{\mathrm{s}}+\tau} - \bar{w} - \rho e^{-\rho\tau}s_{T^{\mathrm{s}}} \tag{A.11a}$$

$$= -\rho(s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}}) \tag{A.11b}$$

$$\overset{(2.68)}{\ge} 0. \tag{A.11c}$$

Since the value of $s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}}$ starts negative, its derivative is positive for $s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}} < 0$ and zero for $s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}} = 0$, $s_\tau - s_{T^{\mathrm{s}}+\tau} + e^{-\rho\tau}s_{T^{\mathrm{s}}}$ exponentially contracts to zero. Therefore, we can conclude that (2.68) holds for $\tau \ge 0$. □

A.3. PROOF OF TERMINAL SET (2.76) INVARIANCE

To prove that terminal set (2.76) is invariant for $\tau \ge 0$, we need to show that if the terminal set constraint is satisfied at prediction time T at time t, i.e.,

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T})} + s_T \le \alpha, \tag{A.12}$$

the terminal set constraint should also be satisfied for $T+\tau, \tau \ge 0$ at time t, i.e.,

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} + s_{T+\tau} \le \alpha. \tag{A.13}$$

Since there are no disturbances in this part of the prediction horizon, we know that terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})$ ensures contraction of $\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})}$ by at least ρ as given by (2.47). Given terminal set satisfaction (A.12), this gives the following upper bound:

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq e^{-\rho\tau}(\alpha - s_T). \tag{A.14}$$

If the terminal set is invariant, the right-hand side of this equation is upper-bounded by $\alpha - s_{T+\tau}$ as derived from (A.13):

$$e^{-\rho\tau}(\alpha - s_T) \leq \alpha - s_{T+\tau}, \tag{A.15}$$

which is trivially satisfied for $\tau = 0$. Working out this inequality for $\tau > 0$, similar to (2.82), gives

$$e^{-\rho\tau}(\alpha - s_T) \leq \alpha - s_{T+\tau} \tag{A.16a}$$
$$(1 - e^{-\rho\tau})\alpha \geq -e^{-\rho\tau} s_T + s_{T+\tau} \tag{A.16b}$$
$$(1 - e^{-\rho\tau})\alpha \geq -e^{-\rho\tau}(1 - e^{-\rho T})\frac{\bar{w}}{\rho} + (1 - e^{-\rho(T+\tau)})\frac{\bar{w}}{\rho} \tag{A.16c}$$
$$(1 - e^{-\rho\tau})\alpha \geq (-e^{-\rho\tau} + e^{-\rho(T+\tau)} + 1 - e^{-\rho(T+\tau)})\frac{\bar{w}}{\rho} \tag{A.16d}$$
$$(1 - e^{-\rho\tau})\alpha \geq (1 - e^{-\rho\tau})\frac{\bar{w}}{\rho} \tag{A.16e}$$
$$\alpha \geq \frac{\bar{w}}{\rho}. \tag{A.16f}$$

Thus, terminal set (2.76) is invariant provided that $\alpha \geq \frac{\bar{w}}{\rho}$. □

A.4. PROOF OF RPI LMI (2.138)

The goal of this appendix is to design an LMI ensuring that sublevel set $V^\delta(\boldsymbol{x}, \boldsymbol{z}) \leq \delta^2$ is RPI for disturbed dynamics (2.39) and $\boldsymbol{w} = \boldsymbol{w}^{\mathrm{b}} + \boldsymbol{w}^0 \in \boldsymbol{w}^{\mathrm{b}} \oplus \mathcal{W}$, with $\mathcal{W}$ given by (2.40). Given state-independent P^δ, $V^\delta(\boldsymbol{x}, \boldsymbol{z})$ is given by (2.133). Consequently, we want the following property to hold:

$$\frac{d}{dt} V^\delta(\boldsymbol{x}, \boldsymbol{z}) \leq 0, \quad \forall \boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \geq \delta^2, \quad \forall \boldsymbol{w}^0 \in \mathcal{W}, \tag{A.17}$$

with $\boldsymbol{\delta}$ defined in (2.48).

Using the state-independency of P^δ, we can remove the integral and simplify the upper bound on $\frac{d}{dt} V^\delta(\boldsymbol{x}, \boldsymbol{z})$ in (2.124b) as follows:

$$\boldsymbol{\delta}^\top (A^{\mathrm{cl}^\top} P^\delta + P^\delta A^{\mathrm{cl}})\boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta E \boldsymbol{w}^0 + (E\boldsymbol{w}^0)^\top P^\delta \boldsymbol{\delta} \leq 0, \tag{A.18}$$

or, alternatively,

$$\begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix}^\top \begin{bmatrix} A^{\mathrm{cl}^\top} P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix} \leq 0. \tag{A.19}$$

We want this inequality to hold for all $\boldsymbol{\delta}$ such that $\boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \geq \delta^2$ and all $\boldsymbol{w}^0 \in \mathcal{W}$. Since $\mathcal{W}$ is a polytopic set and $\boldsymbol{w}^0$ appear linearly in the above inequality, it suffices to check the inequality for the vertices of $\mathcal{W}$, i.e., the inequality should hold for all $\boldsymbol{w}^0 \in \text{vert}(\mathcal{W})$.

To enforce these properties, we can use the S-procedure, which states that $\lambda F_1 - F_2 \succeq 0, \lambda \geq 0$ is a sufficient condition for implication $\boldsymbol{x}^\top F_1 \boldsymbol{x} \leq 0 \implies \boldsymbol{x}^\top F_2 \boldsymbol{x} \leq 0$ to hold [126]. In this case, we leverage the S-procedure to enforce that the following implication holds:

$$\boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \geq \delta^2 \implies \begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix}^\top \begin{bmatrix} A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix} \leq 0. \tag{A.20}$$

Re-ordering the terms on the left side of this implication gives

$$\delta^2 - \boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \leq 0, \tag{A.21}$$

which can be written in matrix form as

$$\begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix}^\top \begin{bmatrix} -P^\delta & \mathbf{0} \\ \mathbf{0} & \delta^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix} \leq 0. \tag{A.22}$$

Now, we can apply the S-procedure with

$$\boldsymbol{x} \coloneqq \begin{bmatrix} \boldsymbol{\delta} \\ 1 \end{bmatrix}, \tag{A.23a}$$

$$F_1 \coloneqq \begin{bmatrix} -P^\delta & \mathbf{0} \\ \mathbf{0} & \delta^2 \end{bmatrix}, \tag{A.23b}$$

$$F_2 \coloneqq \begin{bmatrix} A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & 0 \end{bmatrix}, \tag{A.23c}$$

to obtain

$$\lambda \begin{bmatrix} -P^\delta & \mathbf{0} \\ \mathbf{0} & \delta^2 \end{bmatrix} - \begin{bmatrix} A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & 0 \end{bmatrix} \succeq 0. \tag{A.24}$$

Working out this LMI gives

$$-\begin{bmatrix} A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} + \lambda P^\delta & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & -\lambda \delta^2 \end{bmatrix} \succeq 0, \tag{A.25a}$$

$$\begin{bmatrix} A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} + \lambda P^\delta & P^\delta E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top P^\delta & -\lambda \delta^2 \end{bmatrix} \preceq 0, \tag{A.25b}$$

with $\lambda \geq 0$. Pre- and post-multiplying with $\begin{bmatrix} P^{\delta^{-1}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$ yields

$$\begin{bmatrix} P^{\delta^{-1}} (A^{\text{cl}^\top} P^\delta + P^\delta A^{\text{cl}} + \lambda P^\delta) P^{\delta^{-1}} & P^{\delta^{-1}} (P^\delta E \boldsymbol{w}^0) \\ \left((E\boldsymbol{w}^0)^\top P^\delta\right) P^{\delta^{-1}} & -\lambda \delta^2 \end{bmatrix} \preceq 0, \tag{A.26a}$$

$$\begin{bmatrix} P^{\delta^{-1}} A^{\text{cl}^\top} + A^{\text{cl}} P^{\delta^{-1}} + \lambda P^{\delta^{-1}} & E \boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^\top & -\lambda \delta^2 \end{bmatrix} \preceq 0. \tag{A.26b}$$

A

Filling in the definition of A^{cl} gives

$$\begin{bmatrix} P^{\delta^{-1}}\big(A(\boldsymbol{\zeta})+B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)^{\top}+\big(A(\boldsymbol{\zeta})+B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)P^{\delta^{-1}}+\lambda P^{\delta^{-1}} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^{\top} & -\lambda\delta^2 \end{bmatrix} \preceq 0, \tag{A.27a}$$

$$\begin{bmatrix} P^{\delta^{-1}}A(\boldsymbol{\zeta})^{\top}+P^{\delta^{-1}}K^{\delta}(\boldsymbol{x})^{\top}B(\boldsymbol{\zeta})^{\top}+A(\boldsymbol{\zeta})P^{\delta^{-1}}+B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})P^{\delta^{-1}}+\lambda P^{\delta^{-1}} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^{\top} & -\lambda\delta^2 \end{bmatrix} \preceq 0, \tag{A.27b}$$

which, under the same coordinate transformation as in Section 2.4.3 with state-independent $X^{\delta}=P^{\delta^{-1}}$ and state-dependent $Y^{\delta}(\boldsymbol{x})=K^{\delta}(\boldsymbol{x})P^{\delta}$, and leveraging the symmetry of P^{δ}, gives the following LMI:

$$\begin{bmatrix} X^{\delta}A(\boldsymbol{\zeta})^{\top}+Y^{\delta}(\boldsymbol{x})^{\top}B(\boldsymbol{\zeta})^{\top}+A(\boldsymbol{\zeta})X^{\delta}+B(\boldsymbol{\zeta})Y^{\delta}(\boldsymbol{x})+\lambda X^{\delta} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^{\top} & -\lambda\delta^2 \end{bmatrix} \preceq 0, \tag{A.28a}$$

$$\begin{bmatrix} A(\boldsymbol{\zeta})X^{\delta}+B(\boldsymbol{\zeta})Y^{\delta}(\boldsymbol{x})+\big(A(\boldsymbol{\zeta})X^{\delta}+B(\boldsymbol{\zeta})Y^{\delta}(\boldsymbol{x})\big)^{\top}+\lambda X^{\delta} & E\boldsymbol{w}^0 \\ (E\boldsymbol{w}^0)^{\top} & -\lambda\delta^2 \end{bmatrix} \preceq 0. \tag{A.28b}$$

□

A.5. PROOF OF LIPSCHITZ CONTINUITY OBSTACLE AVOIDANCE CONSTRAINTS LMI (2.139f)

The goal of this appendix is to derive the LMI ensuring Lipschitz continuity of the obstacle avoidance constraints using tightening constant c^{o} and incremental Lyapunov function $V^{\delta}(\boldsymbol{x},\boldsymbol{z})$ according to (2.44c). Following similar steps as in [74], we first derive a Lipschitz bound on general constraints $g_j(\boldsymbol{x},\boldsymbol{u})\le 0$, i.e., a bound of the following form for $j\in\mathbb{N}_{[1,n]}$ where n is the number of constraints:

$$g_j(\boldsymbol{x},\boldsymbol{u})-g_j(\boldsymbol{z},\boldsymbol{v})\le c_j\|\boldsymbol{x}-\boldsymbol{z}\|_{P^{\delta}}. \tag{A.29}$$

The linearization of these constraints with respect to states $\boldsymbol{x}$ and inputs $\boldsymbol{u}$ is given by

$$\frac{\partial g_j}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}+\frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}\boldsymbol{\delta}, \tag{A.30}$$

with $\boldsymbol{\delta}$ defined in (2.48). This gives the following Lipschitz bound:

$$\frac{\partial g_j}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}+\frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}\le c_j\|\boldsymbol{\delta}\|_{P^{\delta}}, \tag{A.31}$$

in which we can rewrite the expression for $\|\boldsymbol{\delta}\|_{P^{\delta}}$ as

$$\|\boldsymbol{x}-\boldsymbol{z}\|_{P^{\delta}}=\sqrt{\boldsymbol{\delta}^{\top}P^{\delta}\boldsymbol{\delta}}=\sqrt{\boldsymbol{\delta}^{\top}P^{\delta\frac{1}{2}}P^{\delta\frac{1}{2}}\boldsymbol{\delta}}=\sqrt{(P^{\delta\frac{1}{2}}\boldsymbol{\delta})^{\top}(P^{\delta\frac{1}{2}}\boldsymbol{\delta})}=\|P^{\delta\frac{1}{2}}\boldsymbol{\delta}\|, \tag{A.32}$$

to obtain

$$\Big(\frac{\partial g_j}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}+\frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}}\Big)\boldsymbol{\delta}\le c_j\|P^{\delta\frac{1}{2}}\boldsymbol{\delta}\|. \tag{A.33}$$

Defining $\tilde{\boldsymbol{\delta}} := P^{\delta^{\frac{1}{2}}}\boldsymbol{\delta}$ gives the following inequality:

$$(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\tilde{\boldsymbol{\delta}} \leq c_j\|\tilde{\boldsymbol{\delta}}\|. \tag{A.34}$$

An upper bound on the left-hand side of this equation is given by

$$(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\tilde{\boldsymbol{\delta}} \leq \|(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\tilde{\boldsymbol{\delta}}\| \tag{A.35a}$$

$$\leq \|(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\|\|\tilde{\boldsymbol{\delta}}\|. \tag{A.35b}$$

Thus, a sufficient condition for (A.34) to hold is

$$\|(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\|\|\tilde{\boldsymbol{\delta}}\| \leq c_j\|\tilde{\boldsymbol{\delta}}\| \tag{A.36a}$$

$$\|(\frac{\partial g_j}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g_j}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}})P^{\delta^{-\frac{1}{2}}}\| \leq c_j. \tag{A.36b}$$

In the following, we derive the LMI that should hold for the obstacle avoidance constraints specifically. first note that obstacle avoidance constraints (2.3) are already linear and do not depend on inputs $\boldsymbol{u}$. Therefore, we can write the following equality:

$$\frac{\partial g^{\mathrm{o}}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} + \frac{\partial g^{\mathrm{o}}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x}} = L^{\mathrm{o}}M. \tag{A.37}$$

Filling this expression in in (A.36) gives the following inequality:

$$\|L^{\mathrm{o}}MP^{\delta^{-\frac{1}{2}}}\| \leq c^{\mathrm{o}}. \tag{A.38}$$

As commented under (2.3), the polytopic obstacle avoidance constraints can always be normalized such that $\|L^{\mathrm{o}}\| = 1$. Therefore, we can write

$$\|L^{\mathrm{o}}MP^{\delta^{-\frac{1}{2}}}\| \leq \|L^{\mathrm{o}}\|\|MP^{\delta^{-\frac{1}{2}}}\| = \|MP^{\delta^{-\frac{1}{2}}}\|. \tag{A.39}$$

Thus, the following implication holds:

$$\|MP^{\delta^{-\frac{1}{2}}}\| \leq c^{\mathrm{o}}I^{n^{\mathrm{p}}} \implies \|L^{\mathrm{o}}MP^{\delta^{-\frac{1}{2}}}\| \leq c^{\mathrm{o}}. \tag{A.40}$$

Working out the left-hand side and applying coordinate change $X^\delta := P^{\delta^{-1}}$ yields

$$\|MP^{\delta^{-\frac{1}{2}}}\| \preceq c^{\mathrm{o}} I^{n^{\mathrm{p}}} \tag{A.41a}$$

$$\|MP^{\delta^{-\frac{1}{2}}}\|^2 \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41b}$$

$$\|(MP^{\delta^{-\frac{1}{2}}})^\top\|^2 \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41c}$$

$$\|P^{\delta^{-\frac{1}{2}}} M^\top\|^2 \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41d}$$

$$(P^{\delta^{-\frac{1}{2}}} M^\top)^\top (P^{\delta^{-\frac{1}{2}}} M) \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41e}$$

$$MP^{\delta^{-\frac{1}{2}}} P^{\delta^{-\frac{1}{2}}} M^\top \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41f}$$

$$MP^{\delta^{-1}} M^\top \preceq c^{\mathrm{o}2} I^{n^{\mathrm{p}}} \tag{A.41g}$$

$$c^{\mathrm{o}2} I^{n^{\mathrm{p}}} - MP^{\delta^{-1}} M^\top \succeq 0 \tag{A.41h}$$

$$c^{\mathrm{o}2} I^{n^{\mathrm{p}}} - MP^{\delta^{-1}} P^\delta P^{\delta^{-1}} M^\top \succeq 0 \tag{A.41i}$$

$$c^{\mathrm{o}2} I^{n^{\mathrm{p}}} - MX^\delta X^{\delta^{-1}} X^\delta M^\top \succeq 0. \tag{A.41j}$$

Taking the Schur complement of this LMI gives

$$\begin{bmatrix} c^{\mathrm{o}2} I^{n^{\mathrm{p}}} & MX^\delta \\ (MX^\delta)^\top & X^\delta \end{bmatrix} \succeq 0. \tag{A.42}$$

□

A.6. PROOF OF TERMINAL SET (2.166) INVARIANCE

The proof in this appendix follows similar steps as the one in Appendix A.3. To prove that terminal set (2.166) is invariant for $\tau \geq 0$, we need to show that if the terminal set constraint is satisfied at prediction time T at time t, i.e.,

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T})} + s_T + \epsilon \leq \alpha, \tag{A.43}$$

the terminal set constraint should also be satisfied for $T+\tau, \tau \geq 0$ at time t, i.e.,

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} + s_{T+\tau} + \epsilon \leq \alpha. \tag{A.44}$$

Since there are no disturbances in this part of the prediction horizon, we know that terminal control law $\kappa^{\mathrm{f}}(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})$ ensures contraction of $\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})}$ by at least ρ as given by (2.47). Given terminal set satisfaction (A.43), this gives the following upper bound:

$$\sqrt{V^\delta(\boldsymbol{z}^*_{T+\tau|t}, \boldsymbol{x}^{\mathrm{r}}_{t+T+\tau})} \leq e^{-\rho\tau}(\alpha - s_T - \epsilon). \tag{A.45}$$

If the terminal set is invariant, the right-hand side of this equation is upper-bounded by $\alpha - s_{T+\tau} - \epsilon$ as derived from (A.44):

$$e^{-\rho\tau}(\alpha - s_T - \epsilon) \leq \alpha - s_{T+\tau} - \epsilon, \tag{A.46}$$

which is trivially satisfied for $\tau = 0$. Working out this inequality for $\tau > 0$, similar to (A.16), gives

$$e^{-\rho\tau}(\alpha - s_T - \epsilon) \le \alpha - s_{T+\tau} - \epsilon \tag{A.47a}$$
$$(1 - e^{-\rho\tau})\alpha \ge -e^{-\rho\tau} s_T + s_{T+\tau} + (1 - e^{-\rho\tau})\epsilon \tag{A.47b}$$
$$(1 - e^{-\rho\tau})\alpha \ge -e^{-\rho\tau}(1 - e^{-\rho T})\frac{\bar{w}^{\mathrm{o}}}{\rho} + (1 - e^{-\rho(T+\tau)})\frac{\bar{w}^{\mathrm{o}}}{\rho} + (1 - e^{-\rho\tau})\epsilon \tag{A.47c}$$
$$(1 - e^{-\rho\tau})\alpha \ge (1 - e^{-\rho\tau} + e^{-\rho(T+\tau)} + e^{-\rho(T+\tau)})\frac{\bar{w}^{\mathrm{o}}}{\rho} + (1 - e^{-\rho\tau})\epsilon \tag{A.47d}$$
$$(1 - e^{-\rho\tau})\alpha \ge (1 - e^{-\rho\tau})\frac{\bar{w}^{\mathrm{o}}}{\rho} + (1 - e^{-\rho\tau})\epsilon \tag{A.47e}$$
$$\alpha \ge \frac{\bar{w}^{\mathrm{o}}}{\rho} + \epsilon. \tag{A.47f}$$

Thus, terminal set (2.166) is invariant provided that $\alpha \ge \frac{\bar{w}^{\mathrm{o}}}{\rho} + \epsilon$. □

A.7. PROOF OF ϵ-RPI LMI (2.185)

The goal of this appendix is to design an LMI ensuring that sublevel set $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}}) \le \epsilon^2$ is RPI for the error between estimated state dynamics $\dot{\hat{\boldsymbol{x}}}$, given by (2.179) with $\boldsymbol{y} \in \mathbb{R}^{n^{\mathrm{y}}}$ and $\boldsymbol{\eta} \in \mathscr{H}$ defined in (2.146) and $\mathscr{H}$ defined in (2.147), and disturbed dynamics $\dot{\boldsymbol{x}}$, given by (2.39) with $\boldsymbol{w} = \boldsymbol{w}^{\mathrm{b}} + \boldsymbol{w}^{0} \in \boldsymbol{w}^{\mathrm{b}} \oplus \mathscr{W}$ and $\mathscr{W}$ defined in (2.40).

Given state-independent P^{δ}, $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is given by

$$V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}}) := (\boldsymbol{x} - \hat{\boldsymbol{x}})^{\top} P^{\delta} (\boldsymbol{x} - \hat{\boldsymbol{x}}). \tag{A.48}$$

For the ϵ-sublevel set of $V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ to be RPI, we want the following property to hold:

$$\frac{d}{dt} V^{\delta}(\boldsymbol{x}, \hat{\boldsymbol{x}}) \le 0, \quad \forall \boldsymbol{\epsilon}^{\top} P^{\delta} \boldsymbol{\epsilon} \ge \epsilon^2, \quad \forall \boldsymbol{w}^{0} \in \mathscr{W} \quad \forall \boldsymbol{\eta} \in \mathscr{H}, \tag{A.49}$$

with $\boldsymbol{\epsilon}$ defined in (2.159). Using the following additional shorthand notations $\dot{\boldsymbol{\epsilon}}$ and A^{cl}:

$$\begin{aligned} \dot{\boldsymbol{\epsilon}} &= \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}}, \\ &= f\big(\boldsymbol{x}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w} - \Big(f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + L\big(F\boldsymbol{\eta} + C(\boldsymbol{x} - \hat{\boldsymbol{x}})\big)\Big) \\ &= f\big(\boldsymbol{x}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{0} - L(F\boldsymbol{\eta} + C\boldsymbol{\epsilon}), \end{aligned} \tag{A.50}$$
$$A := A(\boldsymbol{\gamma}^{\mathrm{x}}, \boldsymbol{\gamma}^{\mathrm{u}}), \tag{A.51}$$

we can upper-bound the time-derivative of $V^\delta(\boldsymbol{x},\hat{\boldsymbol{x}})$ as

$$\begin{aligned}
\frac{d}{dt}V^\delta(\boldsymbol{x},\hat{\boldsymbol{x}}) &= (\dot{\boldsymbol{\epsilon}})^\top P^\delta \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^\top P^\delta \dot{\boldsymbol{\epsilon}} \\
&= 2\boldsymbol{\epsilon}^\top P^\delta \dot{\boldsymbol{\epsilon}} && \text{(A.52a)} \\
&= 2\boldsymbol{\epsilon}^\top P^\delta \Big(f\big(\boldsymbol{x},\kappa(\hat{\boldsymbol{x}},\boldsymbol{z},\boldsymbol{v})\big) - f\big(\hat{\boldsymbol{x}},\kappa(\hat{\boldsymbol{x}},\boldsymbol{z},\boldsymbol{v})\big) + E\boldsymbol{w}^0 - L(F\boldsymbol{\eta}+C\boldsymbol{\epsilon})\Big) && \text{(A.52b)} \\
&= 2\boldsymbol{\epsilon}^\top P^\delta \Big(f\big(\boldsymbol{x},\kappa(\hat{\boldsymbol{x}},\boldsymbol{z},\boldsymbol{v})\big) - f\big(\hat{\boldsymbol{x}},\kappa(\hat{\boldsymbol{x}},\boldsymbol{z},\boldsymbol{v})\big)\Big) - 2\boldsymbol{\epsilon}^\top P^\delta LC\boldsymbol{\epsilon} \\
&\quad + 2\boldsymbol{\epsilon}^\top P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) && \text{(A.52c)} \\
&\overset{\text{(a)}}{\leq} 2\boldsymbol{\epsilon}^\top P^\delta A\boldsymbol{\epsilon} - 2\boldsymbol{\epsilon}^\top P^\delta LC\boldsymbol{\epsilon} + 2\boldsymbol{\epsilon}^\top P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) && \text{(A.52d)} \\
&= 2\boldsymbol{\epsilon}^\top P^\delta (A-LC)\boldsymbol{\epsilon} + 2\boldsymbol{\epsilon}^\top P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) && \text{(A.52e)} \\
&= \boldsymbol{\epsilon}^\top \big((A-LC)^\top P^\delta + P^\delta (A-LC)\big)\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^\top P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) \\
&\quad + (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top P^\delta \boldsymbol{\epsilon}, && \text{(A.52f)}
\end{aligned}$$

where (a) is obtained by knowing that an $s \in [0,1]$ exists, such that this expression holds by the MVT.

Setting the desired upper bound from (A.49) gives

$$\begin{aligned}
\frac{d}{dt}V^\delta(\boldsymbol{x},\hat{\boldsymbol{x}}) &\leq \boldsymbol{\epsilon}^\top \big((A-LC)^\top P^\delta + P^\delta (A-LC)\big)\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^\top P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) + (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top P^\delta \boldsymbol{\epsilon} \\
&\leq 0, && \text{(A.53)}
\end{aligned}$$

or, alternatively,

$$\begin{bmatrix} \boldsymbol{\epsilon} \\ 1 \end{bmatrix}^\top \begin{bmatrix} (A-LC)^\top P^\delta + P^\delta (A-LC) & P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top P^\delta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon} \\ 1 \end{bmatrix} \leq 0. \tag{A.54}$$

It suffices to enforce this inequality for $\boldsymbol{w}^0 \in \text{vert}(\mathscr{W})$ and $\boldsymbol{\eta} \in \text{vert}(\mathscr{H})$ since $\mathscr{W}$ and $\mathscr{H}$ are convex polytopic sets.

Following the S-procedure similar to the derivation in Appendix A.4 results in the following LMI:

$$\begin{bmatrix} (A-LC)^\top P^\delta + P^\delta (A-LC) + \lambda^\epsilon P^\delta & P^\delta (E\boldsymbol{w}^0 - LF\boldsymbol{\eta}) \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top P^\delta & -\lambda^\epsilon \epsilon^2 \end{bmatrix} \leq 0, \tag{A.55}$$

with multiplier $\lambda^\epsilon \geq 0$ and for $\boldsymbol{w}^0 \in \text{vert}(\mathscr{W})$ and $\boldsymbol{\eta} \in \text{vert}(\mathscr{H})$.

From here, following the same steps as presented in Appendix A.4 results in the following ϵ-RPI LMI:

$$\begin{bmatrix} X^\delta \big(A(\boldsymbol{\zeta}) - LC\big)^\top + \big(A(\boldsymbol{\zeta}) - LC\big) X^\delta + \lambda^\epsilon X^\delta & E\boldsymbol{w}^0 - LF\boldsymbol{\eta} \\ (E\boldsymbol{w}^0 - LF\boldsymbol{\eta})^\top & -\lambda^\epsilon \epsilon^2 \end{bmatrix} \leq 0, \tag{A.56}$$

with $\lambda^\epsilon \geq 0$ and for $\boldsymbol{w}^0 \in \text{vert}(\mathscr{W})$ and $\boldsymbol{\eta} \in \text{vert}(\mathscr{H})$.

Note that this LMI is similar to the RPI LMI derived in Appendix A.4, but with additional terms proportional to L in the top-left block and with $E\boldsymbol{w}^0 - LF\boldsymbol{\eta}$ instead of $E\boldsymbol{w}^0$ in the top-right and bottom-left blocks. □

A.8. PROOF OF δ-RPI LMI (2.188)

The goal of this appendix is to derive the LMI ensuring that sublevel set $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq \delta^2$ is RPI for the error between estimated state dynamics $\dot{\hat{\boldsymbol{x}}}$, given by (2.179) with $\boldsymbol{y} \in \mathbb{R}^{n^y}$ and $\boldsymbol{\eta} \in \mathscr{H}$ defined in (2.146) and $\mathscr{H}$ defined in (2.147), and observer error $\boldsymbol{\epsilon}$, defined in (2.159), satisfying $\boldsymbol{\epsilon}^\top P^\delta \boldsymbol{\epsilon} \leq \epsilon^2$, as enforced by LMI (2.185).

For the δ-sublevel set of $V^\delta(\boldsymbol{x}, \hat{\boldsymbol{x}})$ to be RPI, we want the following property to hold:

$$\frac{d}{dt} V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq 0, \quad \forall \boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \geq \delta^2, \quad \forall \boldsymbol{\epsilon}^\top P^\delta \boldsymbol{\epsilon} \leq \epsilon^2, \quad \forall \boldsymbol{\eta} \in \mathscr{H}, \tag{A.57}$$

with $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ defined in (2.158) and (2.159), respectively. Using the following shorthand notation for A^{cl}:

$$A^{\mathrm{cl}} := A(\boldsymbol{\zeta}) + B(\boldsymbol{\zeta}) K^\delta(\boldsymbol{x}), \tag{A.58}$$

we can upper-bound the time-derivative of $V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z})$ as

$$\begin{aligned}
\frac{d}{dt} V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) &= \dot{\boldsymbol{\delta}}^\top P^\delta \boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta \dot{\boldsymbol{\delta}} && \text{(A.59a)}\\
&= 2\boldsymbol{\delta}^\top P^\delta \dot{\boldsymbol{\delta}} && \text{(A.59b)}\\
&= 2\boldsymbol{\delta}^\top P^\delta \Big(f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) + E\boldsymbol{w}^{\mathrm{b}} + LF\boldsymbol{\eta} + LC\boldsymbol{\epsilon} - f(\boldsymbol{z}, \boldsymbol{v}) - E\boldsymbol{w}^{\mathrm{b}} \Big) && \text{(A.59c)}\\
&= 2\boldsymbol{\delta}^\top P^\delta \Big(f\big(\hat{\boldsymbol{x}}, \kappa(\hat{\boldsymbol{x}}, \boldsymbol{z}, \boldsymbol{v})\big) - f(\boldsymbol{z}, \boldsymbol{v}) \Big) + 2\boldsymbol{\delta}^\top P^\delta (LF\boldsymbol{\eta} + LC\boldsymbol{\epsilon}) && \text{(A.59d)}\\
&\overset{\text{(a)}}{\leq} 2\boldsymbol{\delta}^\top P^\delta A^{\mathrm{cl}} \boldsymbol{\delta} + 2\boldsymbol{\delta}^\top P^\delta (LF\boldsymbol{\eta} + LC\boldsymbol{\epsilon}) && \text{(A.59e)}\\
&= \boldsymbol{\delta}^\top A^{\mathrm{cl}\top} P^\delta \boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta A^{\mathrm{cl}} \boldsymbol{\delta} + (LF\boldsymbol{\eta} + LC\boldsymbol{\epsilon})^\top P^\delta \boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta (LF\boldsymbol{\eta} + LC\boldsymbol{\epsilon}) && \text{(A.59f)}\\
&= \boldsymbol{\delta}^\top (A^{\mathrm{cl}\top} P^\delta + P^\delta A^{\mathrm{cl}}) \boldsymbol{\delta} + (LC\boldsymbol{\epsilon})^\top P^\delta \boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta LC\boldsymbol{\epsilon}\\
&\quad + (LF\boldsymbol{\eta})^\top P^\delta \boldsymbol{\delta} + \boldsymbol{\delta}^\top P^\delta LF\boldsymbol{\eta}, && \text{(A.59g)}\\
&= \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix}^\top \begin{bmatrix} A^{\mathrm{cl}\top} P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^x} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix}, && \text{(A.59h)}
\end{aligned}$$

where (a) is obtained by knowing that an $s \in [0, 1]$ exists, such that this expression holds by the MVT.

Setting the desired upper bound from (A.57) gives

$$\frac{d}{dt} V^\delta(\hat{\boldsymbol{x}}, \boldsymbol{z}) \leq \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix}^\top \begin{bmatrix} A^{\mathrm{cl}\top} P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^x} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix} \leq 0. \tag{A.60}$$

It suffices to enforce this inequality for $\boldsymbol{\eta} \in \mathscr{H}$ since $\mathscr{H}$ is a convex polytopic set.

A

Applying the S-procedure to enforce that the inequality holds $\forall \boldsymbol{\delta}^\top P^\delta \boldsymbol{\delta} \geq \delta^2$ with

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix}, \tag{A.61a}$$

$$F_1 := \begin{bmatrix} -P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \delta^2 \end{bmatrix}, \tag{A.61b}$$

$$F_2 := \begin{bmatrix} {A^{\mathrm{cl}}}^\top P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & 0 \end{bmatrix}, \tag{A.61c}$$

results in the following LMI:

$$\lambda^\delta \begin{bmatrix} -P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \delta^2 \end{bmatrix} - \begin{bmatrix} {A^{\mathrm{cl}}}^\top P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & 0 \end{bmatrix} \succeq 0 \tag{A.62a}$$

$$\begin{bmatrix} {A^{\mathrm{cl}}}^\top P^\delta + P^\delta A^{\mathrm{cl}} & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & 0 \end{bmatrix} - \lambda^\delta \begin{bmatrix} -P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \delta^2 \end{bmatrix} \preceq 0 \tag{A.62b}$$

$$\begin{bmatrix} {A^{\mathrm{cl}}}^\top P^\delta + P^\delta A^{\mathrm{cl}} + \lambda^\delta P^\delta & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & -\lambda^\delta \delta^2 \end{bmatrix} \preceq 0, \tag{A.62c}$$

with $\lambda^\delta \geq 0$.

We want this LMI to hold $\forall \boldsymbol{\epsilon}^\top P^\delta \boldsymbol{\epsilon} \leq \epsilon^2$. Applying the S-procedure again, this time with

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ 1 \end{bmatrix}, \tag{A.63a}$$

$$F_1 := \begin{bmatrix} 0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0} \\ 0^{n^{\mathrm{x}}} & P^\delta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\epsilon^2 \end{bmatrix}, \tag{A.63b}$$

$$F_2 := \begin{bmatrix} {A^{\mathrm{cl}}}^\top P^\delta + P^\delta A^{\mathrm{cl}} + \lambda^\delta P^\delta & P^\delta LC & P^\delta LF\boldsymbol{\eta} \\ (LC)^\top P^\delta & 0 & \mathbf{0} \\ (LF\boldsymbol{\eta})^\top P^\delta & \mathbf{0} & -\lambda^\delta \delta^2 \end{bmatrix}, \tag{A.63c}$$

results in the following LMI:

$$\lambda^{\delta,\epsilon}\begin{bmatrix}0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & -\epsilon^2\end{bmatrix} - \begin{bmatrix}A^{\mathrm{cl}\top}P^{\delta}+P^{\delta}A^{\mathrm{cl}}+\lambda^{\delta}P^{\delta} & P^{\delta}LC & P^{\delta}LF\boldsymbol{\eta}\\ (LC)^{\top}P^{\delta} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top}P^{\delta} & \mathbf{0} & -\lambda^{\delta}\delta^2\end{bmatrix} \succeq 0 \tag{A.64a}$$

$$\begin{bmatrix}A^{\mathrm{cl}\top}P^{\delta}+P^{\delta}A^{\mathrm{cl}}+\lambda^{\delta}P^{\delta} & P^{\delta}LC & P^{\delta}LF\boldsymbol{\eta}\\ (LC)^{\top}P^{\delta} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top}P^{\delta} & \mathbf{0} & -\lambda^{\delta}\delta^2\end{bmatrix} - \lambda^{\delta,\epsilon}\begin{bmatrix}0^{n^{\mathrm{x}}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & -\epsilon^2\end{bmatrix} \preceq 0 \tag{A.64b}$$

$$\begin{bmatrix}A^{\mathrm{cl}\top}P^{\delta}+P^{\delta}A^{\mathrm{cl}}+\lambda^{\delta}P^{\delta} & P^{\delta}LC & P^{\delta}LF\boldsymbol{\eta}\\ (LC)^{\top}P^{\delta} & -\lambda^{\delta,\epsilon}P^{\delta} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top}P^{\delta} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix} \preceq 0, \tag{A.64c}$$

with $\lambda^{\delta},\lambda^{\delta,\epsilon}\geq 0$. Pre- and post-multiplying by $\begin{bmatrix}P^{\delta^{-1}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix}$ yields

$$\begin{bmatrix}P^{\delta^{-1}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}A^{\mathrm{cl}\top}P^{\delta}+P^{\delta}A^{\mathrm{cl}}+\lambda^{\delta}P^{\delta} & P^{\delta}LC & P^{\delta}LF\boldsymbol{\eta}\\ (LC)^{\top}P^{\delta} & -\lambda^{\delta,\epsilon}P^{\delta} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top}P^{\delta} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix}\begin{bmatrix}P^{\delta^{-1}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix} \preceq 0 \tag{A.65a}$$

$$\begin{bmatrix}P^{\delta^{-1}}A^{\mathrm{cl}\top}P^{\delta}+A^{\mathrm{cl}}+\lambda^{\delta} & LC & LF\boldsymbol{\eta}\\ P^{\delta^{-1}}(LC)^{\top} & -\lambda^{\delta,\epsilon} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top}P^{\delta} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix}\begin{bmatrix}P^{\delta^{-1}} & 0^{n^{\mathrm{x}}} & \mathbf{0}\\ 0^{n^{\mathrm{x}}} & P^{\delta^{-1}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & 1\end{bmatrix} \preceq 0 \tag{A.65b}$$

$$\begin{bmatrix}P^{\delta^{-1}}A^{\mathrm{cl}\top}+A^{\mathrm{cl}}P^{\delta^{-1}}+\lambda^{\delta}P^{\delta^{-1}} & LCP^{\delta^{-1}} & LF\boldsymbol{\eta}\\ P^{\delta^{-1}}(LC)^{\top} & -\lambda^{\delta,\epsilon}P^{\delta^{-1}} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix} \preceq 0. \tag{A.65c}$$

Filling in the definition of A^{cl} gives

$$\begin{bmatrix}P^{\delta^{-1}}\big(A(\boldsymbol{\zeta})+B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)^{\top}+\big(A(\boldsymbol{\zeta})+B(\boldsymbol{\zeta})K^{\delta}(\boldsymbol{x})\big)P^{\delta^{-1}}+\lambda^{\delta}P^{\delta^{-1}} & LCP^{\delta^{-1}} & LF\boldsymbol{\eta}\\ P^{\delta^{-1}}(LC)^{\top} & -\lambda^{\delta,\epsilon}P^{\delta^{-1}} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix} \preceq 0, \tag{A.66}$$

which, under the coordinate change $X^{\delta}:=P^{\delta^{-1}}$, $Y^{\delta}(\boldsymbol{x}):=K^{\delta}(\boldsymbol{x})X^{\delta}$ yields the following LMI:

$$\begin{bmatrix}A(\boldsymbol{\zeta})X^{\delta}+B(\boldsymbol{\zeta})Y^{\delta}(\boldsymbol{x})+\big(A(\boldsymbol{\zeta})X^{\delta}+B(\boldsymbol{\zeta})Y^{\delta}(\boldsymbol{x})\big)^{\top}+\lambda^{\delta}X^{\delta} & LCX^{\delta} & LF\boldsymbol{\eta}\\ (LCX^{\delta})^{\top} & -\lambda^{\delta,\epsilon}X^{\delta} & \mathbf{0}\\ (LF\boldsymbol{\eta})^{\top} & \mathbf{0} & \lambda^{\delta,\epsilon}\epsilon^2-\lambda^{\delta}\delta^2\end{bmatrix} \preceq 0. \tag{A.67}$$

□

A

A.9. PROOF OF NORM LMI (2.196d)

The goal of this appendix is to show that (2.197) is enforced by LMI (2.196d).

Squaring (2.197) yields

$$\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|^2 \leq l^2. \tag{A.68}$$

Working out this inequality in matrix form gives

$$\begin{aligned}
\|P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}\|^2 &= (P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}})^\top(P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}}) && \text{(A.69a)}\\
&\overset{(a)}{=} P^{\delta^{-\frac{1}{2}}}(LC)^\top P^{\delta\frac{1}{2}}P^{\delta\frac{1}{2}}LCP^{\delta^{-\frac{1}{2}}} && \text{(A.69b)}\\
&= P^{\delta^{-\frac{1}{2}}}(LC)^\top P^{\delta}LCP^{\delta^{-\frac{1}{2}}} && \text{(A.69c)}\\
&\leq l^2 I^{n^x}, && \text{(A.69d)}
\end{aligned}$$

where (a) holds by symmetry of P^δ. Multiplying both sides with $P^{\delta\frac{1}{2}}$ from left and right yields

$$(LC)^\top P^\delta LC \preceq l^2 P^\delta, \tag{A.70}$$

or, equivalently,

$$l^2 P^\delta - (LC)^\top P^\delta P^{\delta^{-1}} P^\delta LC \succeq 0. \tag{A.71}$$

Taking the Schur complement of this LMI by leveraging $P^\delta \succeq 0$ gives

$$\begin{bmatrix} P^\delta & P^\delta LC \\ (P^\delta LC)^\top & l^2 P^\delta \end{bmatrix} \succeq 0, \tag{A.72}$$

which is equivalent to LMI (2.196d). □

B

PROOFS RELATED TO CHAPTER 3

This appendix contains the mathematical proofs of Theorems 4 and 5 presented in Section 3.5.2. The proofs are detailed in Sections B.1 and B.2, respectively.

B.1. PROOF THEOREM 4

Proof. The proof consists of three parts with similar steps as presented in Sections 2.3.2 and 2.3.2. The parts are included in this section for completeness. **Part I** proposes the previously optimal solution shifted by $T^{\mathrm{s}} = T^{\mathrm{s,t}}$, appended with the state after applying κ^{f}, as candidate solution with horizon $T = T^{\mathrm{t}}$ for the next TMPC run. **Part II** shows that all system and obstacle constraints are satisfied for the candidate solution. Finally, **Part III** shows asymptotic convergence of the state to the reference trajectory.

Part I. Candidate Solution
For convenience, define for $\tau \in [T, T + T^{\mathrm{s}}]$:

$$\boldsymbol{u}^*_{\tau|t} := \kappa^{\mathrm{f}}(\boldsymbol{x}^*_{\tau|t}, \boldsymbol{r}^{\mathrm{r}}_{\tau|t}), \tag{B.1}$$

and $\boldsymbol{x}^*_{\tau|t}$ according to (3.2c) by applying (B.1).

Consider the following candidate solution:

$$\tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}} = \boldsymbol{u}^*_{\tau+T^{\mathrm{s}}|t},\ \tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}} = \boldsymbol{x}^*_{\tau+T^{\mathrm{s}}|t},\ \tau \in [0, T], \tag{B.2}$$

with the reference trajectory defined by (3.15) in Property 4.

Part II. Recursive Feasibility
Part II-I. System Constraints Satisfaction For $\tau \in [0, T - T^{\mathrm{s}})$, $j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}$, the system constraints given the candidate solution satisfy:

$$g^{\mathrm{s}}_j(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}}) \overset{(B.2)}{=} g^{\mathrm{s}}_j(\boldsymbol{x}^*_{\tau+T^{\mathrm{s}}|t}, \boldsymbol{u}^*_{\tau+T^{\mathrm{s}}|t}) \overset{(3.2d)}{\leq} 0. \tag{B.3}$$

Part II-II. Obstacle Avoidance Similarly, by Property 5 (b), for $\tau \in (0, T - T^{\mathrm{s}}]$, $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$, the obstacle avoidance constraints given the candidate satisfy:

$$g^{\mathrm{o}}_{j,\tau+T^{\mathrm{s}}|t}(\boldsymbol{p}^*_{\tau+T^{\mathrm{s}}|t}) \overset{(3.2e)(3.21)}{\leq} 0 \overset{(B.2)}{\Longrightarrow} g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\tilde{\boldsymbol{p}}_{\tau|t+T^{\mathrm{s}}}) \leq 0, \tag{B.4}$$

thus ensuring obstacle avoidance by Property 5 (a).

Part II-III. Terminal Constraints Satisfaction Given candidate (B.2) and reference (3.15), the following holds for $\tau \in [T - T^{\mathrm{s}}, T]$:

$$\mathscr{J}^{\mathrm{f,t}}(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}, \boldsymbol{x}^{\mathrm{r}}_{\tau|t+T^{\mathrm{s}}}) \overset{(B.2)}{=} \mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}^*_{\tau+T^{\mathrm{s}}|t}, \boldsymbol{x}^{\mathrm{r}}_{\tau|t+T^{\mathrm{s}}}) \overset{(3.16a)}{\leq} \mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}^*_{T|t}, \boldsymbol{x}^{\mathrm{r}}_{T|t}) \overset{(3.2f)}{\leq} \alpha^2, \tag{B.5}$$

i.e. the terminal set $(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}} - \boldsymbol{x}^{\mathrm{r}}_{\tau|t+T^{\mathrm{s}}}) \in \mathscr{X}^{\mathrm{t,f}}$ is positive invariant. Therefore, by Proposition 1, the following holds for $\tau \in [T - T^{\mathrm{s}}, T]$:

$$g^{\mathrm{s}}_j(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}, \tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}}) \overset{(3.16b)}{\leq} 0, \qquad j \in \mathbb{N}_{[1,n^{\mathrm{s}}]}, \tag{B.6}$$

$$g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\tilde{\boldsymbol{p}}_{\tau|t+T^{\mathrm{s}}}) \overset{(3.16c)}{\leq} 0, \qquad j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}. \tag{B.7}$$

Part III. Asymptotic Convergence
Asymptotic convergence is shown using Barbalat's Lemma. By (3.16a), the following holds:

$$\mathscr{J}^{\mathrm{f,t}}(\tilde{\boldsymbol{x}}_{T|t+T^{\mathrm{s}}}-\boldsymbol{x}^{\mathrm{r}}_{T|t+T^{\mathrm{s}}})-\mathscr{J}^{\mathrm{f,t}}(\boldsymbol{x}^{*}_{T|t}-\boldsymbol{x}^{\mathrm{r}}_{T|t})+\int_{\tau=T-T^{\mathrm{s}}}^{T}\mathscr{J}^{\mathrm{s,t}}(\boldsymbol{x}^{*}_{\tau+T^{\mathrm{s}}|t},\boldsymbol{x}^{*}_{\tau+T^{\mathrm{s}}|t},\boldsymbol{r}_{\tau+T^{\mathrm{s}}|t})d\tau\leq 0. \tag{B.8}$$

Therefore, the following inequality is established:

$$\begin{aligned}\mathscr{J}^{*,\mathrm{t}}(\boldsymbol{x}_{t+T^{\mathrm{s}}},\boldsymbol{x}^{\mathrm{r}}_{\cdot|t+T^{\mathrm{s}}})&\leq\int_{\tau=0}^{T}\mathscr{J}^{\mathrm{s,t}}(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}},\tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}},\boldsymbol{r}_{\tau|t+T^{\mathrm{s}}})d\tau+\mathscr{J}^{\mathrm{f,t}}(\tilde{\boldsymbol{x}}_{T|t+T^{\mathrm{s}}}-\boldsymbol{x}^{\mathrm{r}}_{T|t+T^{\mathrm{s}}})\\&\overset{(B.8)}{\leq}\mathscr{J}^{*,\mathrm{t}}(\boldsymbol{x}_{t},\boldsymbol{x}^{\mathrm{r}}_{\cdot|t})-\int_{\tau=0}^{T^{\mathrm{s}}}\mathscr{J}^{\mathrm{s,t}}(\boldsymbol{x}^{*}_{\tau|t},\boldsymbol{x}^{*}_{\tau|t},\boldsymbol{r}_{\tau|t})d\tau,\end{aligned} \tag{B.9}$$

which proves:

$$\mathscr{J}^{*,\mathrm{t}}(\boldsymbol{x}_{t+T^{\mathrm{s}}},\boldsymbol{x}^{\mathrm{r}}_{\cdot|t+T^{\mathrm{s}}})-\mathscr{J}^{*,\mathrm{t}}(\boldsymbol{x}_{t},\boldsymbol{x}^{\mathrm{r}}_{\cdot|t})\leq -c^{\mathscr{J},\mathrm{d}}\int_{\tau=t}^{t+T^{\mathrm{s}}}\|\boldsymbol{x}_{t}-\boldsymbol{x}^{\mathrm{r}}_{t}\|^{2}_{Q}d\tau, \tag{B.10}$$

using $Q,R>0$ with $c^{\mathscr{J},\mathrm{d}}>0$. Iterating this inequality and using the fact that $\mathscr{J}^{*,\mathrm{t}}(\boldsymbol{x}_{t},\boldsymbol{x}^{\mathrm{r}}_{\cdot|t})$ is uniformly bounded yields:

$$\lim_{t\to\infty}\int_{0}^{t}\|\boldsymbol{x}_{\tau}-\boldsymbol{x}^{\mathrm{r}}_{\tau}\|^{2}_{Q}d\tau<\infty. \tag{B.11}$$

Asymptotic convergence follows by invoking Barbalat's Lemma. □

B.2. PROOF THEOREM 5

Proof. The proof consists of two parts. **Part I** proposes the previously optimal solution shifted by $T^{\mathrm{s}}=T^{\mathrm{s,p}}$, appended with an input ensuring steady state, as candidate solution with horizon $T=T^{\mathrm{p}}$ for the next PMPC run. **Part II** shows feasibility of the candidate solution given the previously optimal solution.

Part I. Candidate Solution
Consider the following candidate solution, valid by (3.19f):

$$\tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}}=\begin{cases}\boldsymbol{u}^{*}_{\tau+T^{\mathrm{s}}|t}, & \tau\in[0,T-T^{\mathrm{s}}),\\ \boldsymbol{u}^{*}_{T|t}, & \tau\in[T-T^{\mathrm{s}},T],\end{cases} \tag{B.12a}$$

$$\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}}=\begin{cases}\boldsymbol{x}^{*}_{\tau+T^{\mathrm{s}}|t}, & \tau\in[0,T-T^{\mathrm{s}}),\\ \boldsymbol{x}^{*}_{T|t}, & \tau\in[T-T^{\mathrm{s}},T].\end{cases} \tag{B.12b}$$

Part II. Recursive Feasibility
Part II-I. System Constraints Satisfaction For $\tau\in[0,T-T^{\mathrm{s}})$, $j\in\mathbb{N}_{[1,n^{\mathrm{s}}]}$, the system constraints given the candidate solution are:

$$g^{\mathrm{s}}_{j}(\tilde{\boldsymbol{x}}_{\tau|t+T^{\mathrm{s}}},\tilde{\boldsymbol{u}}_{\tau|t+T^{\mathrm{s}}})\overset{(B.12)}{=}g^{\mathrm{s}}_{j}(\boldsymbol{x}^{*}_{\tau+T^{\mathrm{s}}|t},\boldsymbol{u}^{*}_{\tau+T^{\mathrm{s}}|t})\overset{(3.19d)}{\leq}0. \tag{B.13}$$

Part II-II. Obstacle Avoidance Constraints Satisfaction Similarly, by Property 5 (b) for $\tau \in (0, T - T^{\mathrm{s}}]$, $j \in \mathbb{N}_{[1,n^{\mathrm{o}}]}$, the obstacle avoidance constraints given the candidate satisfy:

$$g^{\mathrm{o}}_{j,\tau+T^{\mathrm{s}}|t}(\boldsymbol{p}^{*}_{\tau+T^{\mathrm{s}}|t}) \overset{(3.19e)(3.21)(3.13)}{\leq} 0 \overset{(B.12b)}{\Longrightarrow} g^{\mathrm{o}}_{j,\tau|t+T^{\mathrm{s}}}(\tilde{\boldsymbol{p}}_{\tau|t+T^{\mathrm{s}}}) \leq 0. \tag{B.14}$$

Part II-III. Terminal Constraints Satisfaction Given the steady-state condition in candidate (B.12), (B.13) and (B.14) hold for $\tau \in [T - T^{\mathrm{s}}, T]$ as well. □

BIBLIOGRAPHY

[1] RTL. "Drugslab ontmanteld in woonwijk: 125.000 XTC-pillen met 1 miljoen straatwaarde | De wijkagent", Youtube. (2025), [Online]. Available: https://www.youtube.com/watch?v=gTNLs3jWOTk.

[2] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[3] B. Padmaja, C. V. Moorthy, N. Venkateswarulu, and M. M. Bala, "Exploration of issues, challenges and latest developments in autonomous cars", *Journal of Big Data*, vol. 10, no. 1, p. 61, 2023.

[4] E. O. Sodiya, U. J. Umoga, O. O. Amoo, and A. Atadoga, "AI-driven warehouse automation: A comprehensive review of systems", *GSC Advanced Research and Reviews*, vol. 18, no. 2, pp. 272–282, 2024.

[5] J. Iqbal, R. U. Islam, S. Z. Abbas, A. A. Khan, and S. A. Ajwad, "Automating industrial tasks through mechatronic systems-a review of robotics in industrial perspective.", *Tehnicki vjesnik/Technical Gazette*, vol. 23, no. 3, 2016.

[6] V. Grubišić and B. Crnokić, "A systematic review of robotics' transformative role in education", in *International Conference on Digital Transformation in Education and Artificial Intelligence Application*, Springer, 2024, pp. 257–272.

[7] Y. Gao and S. Chien, "Review on space robotics: Toward top-level science through space exploration", *Science Robotics*, vol. 2, no. 7, eaan5074, 2017.

[8] J. Delmerico *et al.*, "The current state and future outlook of rescue robotics", *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[9] C. P. Ezenkwu and A. Starkey, "Machine autonomy: Definition, approaches, challenges and research gaps", in *Intelligent Computing: Proceedings of the 2019 Computing Conference, Volume 1*, Springer, 2019, pp. 335–358.

[10] J. M. Beer, A. D. Fisk, and W. A. Rogers, "Toward a framework for levels of robot autonomy in human-robot interaction", *Journal of human-robot interaction*, vol. 3, no. 2, p. 74, 2014.

[11] M. Beetz *et al.*, "Trustworthiness and well-being: The ethical, legal, and social challenge of robotic assistance", *Robots in Care and Everyday Life*, p. 1, 2023.

[12] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications", *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019.

[13] S. Thakar, S. Srinivasan, S. Al-Hussaini, *et al.*, "A survey of wheeled mobile manipulation: A decision-making perspective", *Journal of Mechanisms and Robotics*, vol. 15, no. 2, p. 020 801, 2023.

[14] S. Garg *et al.*, "Semantics for robotic mapping, perception and interaction: A survey", *Foundations and Trends® in Robotics*, vol. 8, no. 1–2, pp. 1–224, 2020.

[15] N. Sethi, "Semantic target search and exploration using mavs in cluttered environments", M.S. thesis, Delft University of Technology, The Netherlands, 2003. [Online]. Available: https://resolver.tudelft.nl/uuid:c30dd07b-d2ca-4fed-98c9-7f4b874c7f26.

[16] H. Christensen *et al.*, "A roadmap for US robotics—from internet to robotics 2020 edition", *Foundations and Trends® in Robotics*, vol. 8, no. 4, pp. 307–424, 2021.

[17] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[19] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments", *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922–938, 2021.

[20] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor", in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, vol. 3, 2004, pp. 2451–2456.

[21] P. Foehn and D. Scaramuzza, "Onboard state dependent LQR for agile quadrotors", in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 6566–6572.

[22] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: Theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[23] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles", *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[24] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots", *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[25] A. Orthey, C. Chamzas, and L. E. Kavraki, "Sampling-based motion planning: A comparative review", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, pp. 285–310, 2023.

[26] L. Li, J. Li, and S. Zhang, "State-of-the-art trajectory tracking of autonomous vehicles", *Mechanical Sciences*, vol. 12, no. 1, pp. 419–432, 2021.

[27] C. Papachristos, T. Dang, S. Khattak, F. Mascarich, N. Khedekar, and K. Alexis, "Modeling, control, state estimation and path planning methods for autonomous multirotor aerial robots", *Foundations and Trends® in Robotics*, vol. 7, no. 3, pp. 180–250, 2018.

[28] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control", *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

[29] A. Bemporad and D. Barcelli, "Decentralized model predictive control", *Networked control systems*, pp. 149–178, 2010. DOI: 10.1007/978-0-85729-033-5_5.

[30] R. Negenborn and J. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities", *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014. DOI: 10.1109/MCS.2014.2320397.

[31] M. Neunert *et al.*, "Fast nonlinear model predictive control for unified trajectory optimization and tracking", in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 1398–1404.

[32] M. Kögel, M. Ibrahim, C. Kallies, and R. Findeisen, "Safe hierarchical model predictive control and planning for autonomous systems", *International Journal of Robust and Nonlinear Control*, vol. 35, no. 7, pp. 2658–2676, 2025.

[33] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles", in *2008 American Control Conference*, IEEE, 2008, pp. 3719–3724.

[34] B. Elsayed and R. Findeisen, "Generic motion primitives-based safe motion planner under uncertainty for autonomous navigation in cluttered environments", in *2023 XXIX International Conference on Information, Communication and Automation Technologies (ICAT)*, IEEE, 2023, pp. 1–6.

[35] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems", *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[36] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots", *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.

[37] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, "Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics", *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, 2022.

[38] J. Köhler, M. A. Müller, and F. Allgöwer, "A nonlinear model predictive control framework using reference generic terminal ingredients", *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3576–3583, 2019, extended preprint on arXiv: arXiv:1909.12765v2.

[39] ——, "Analysis and design of model predictive control frameworks for dynamic operation—an overview", *Annual Reviews in Control*, vol. 57, p. 100 929, 2024.

[40] P. Foehn *et al.*, "Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight", *Science robotics*, vol. 7, no. 67, eabl6259, 2022.

[41] D. J. MacKay, "Introduction to gaussian processes", *NATO ASI series F computer and systems sciences*, vol. 168, pp. 133–166, 1998.

[42] J. Probst, "Data-driven optimal control for safe quadrotor navigation in windy environments", M.S. thesis, Delft University of Technology, The Netherlands, 2023. [Online]. Available: https://resolver.tudelft.nl/uuid:aaee3499-06cd-4aa7-82ae-6798ec37ce5d.

[43] S. Carena, "Wind estimation using gaussian process regression for safe drone control", M.S. thesis, Politecnico di Torino, Italy, 2024. [Online]. Available: https://webthesis.biblio.polito.it/secure/33884/1/tesi.pdf.

[44] A. Tsolakis, D. Benders, O. De Groot, R. Negenborn, V. Reppa, and L. Ferranti, "COLREGs-aware trajectory optimization for autonomous surface vessels", *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 269–274, 2022.

[45] A. Tsolakis, "Rule-compliant and fault-tolerant motion planning: With application to autonomous surface vehicles", Dissertation, Delft University of Technology, 2025. DOI: 10.4233/uuid:6d3b427d-e1fc-488c-a2f8-9fa011d63520.

[46] J. Köhler, P. Kötting, R. Soloperto, F. Allgöwer, and M. A. Müller, "A robust adaptive model predictive control framework for nonlinear uncertain systems", *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8725–8749, 2021.

[47] F. Bos, A. A. Meera, D. Benders, and M. Wisse, "Free energy principle for state and input estimation of a quadcopter flying in wind", in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 5389–5395.

[48] A. Bemporad and M. Morari, "Robust model predictive control: A survey", in *Robustness in identification and control*, Springer, 2007, pp. 207–226.

[49] B. Kouvaritakis and M. Cannon, *Model predictive control*. Springer, 2016, vol. 38, pp. 13–56.

[50] F. Allgower, R. Findeisen, and Z. K. Nagy, "Nonlinear model predictive control: From theory to application", *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.

[51] D. Q. Mayne, "Model predictive control: Recent developments and future promise", *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[52] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.

[53] M. B. Saltık, L. Özkan, J. H. Ludlage, S. Weiland, and P. M. Van den Hof, "An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects", *Journal of Process Control*, vol. 61, pp. 77–102, 2018.

[54] T. P. Nascimento, C. E. Dórea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: A survey", *Robotica*, vol. 36, no. 5, pp. 676–696, 2018.

[55] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart, "Model predictive control for micro aerial vehicles: A survey", in *2021 European Control Conference (ECC)*, IEEE, 2021, pp. 1556–1563.

[56] Y. Shi and K. Zhang, "Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives", *Annual Reviews in Control*, vol. 52, pp. 170–196, 2021.

[57] H. Wei and Y. Shi, "MPC-based motion planning and control enables smarter and safer autonomous marine vehicles: Perspectives and a tutorial survey", *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 8–24, 2022.

[58] T. Faulwasser, "Optimization-based solutions to constrained trajectory-tracking and path-following problems", Dissertation, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany, Oct. 2012. DOI: 10.25673/3900.

[59] J. Köhler, M. A. Müller, and F. Allgöwer, "A novel constraint tightening approach for nonlinear robust model predictive control", in *2018 Annual American control conference (ACC)*, IEEE, 2018, pp. 728–734.

[60] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control", in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 811–817.

[61] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems", *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2020, extended preprint on arXiv: arXiv:1910.12081v2.

[62] A. Sasfi, M. N. Zeilinger, and J. Köhler, "Robust adaptive MPC using control contraction metrics", *Automatica*, vol. 155, p. 111 169, 2023.

[63] M. Kögel and R. Findeisen, "Robust output feedback MPC for uncertain linear systems with reduced conservatism", *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 685–10 690, 2017.

[64] J. Köhler, F. Allgöwer, and M. A. Müller, "A simple framework for nonlinear robust output-feedback MPC", in *2019 18th European control conference (ECC)*, IEEE, 2019, pp. 793–798.

[65] G. Chou, N. Ozay, and D. Berenson, "Safe output feedback motion planning from images via learned perception modules and contraction theory", in *International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2022, pp. 349–367.

[66] S. Brown, M. Khajenejad, A. Suresh, and S. Martínez, "Robust output-feedback MPC for nonlinear systems with applications to robotic exploration", *IEEE Control Systems Letters*, vol. 9, pp. 90–95, 2025.

[67] S. Liu *et al.*, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments", *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[68] J. Arrizabalaga, Z. Manchester, and M. Ryll, "Differentiable collision-free parametric corridors", in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, pp. 1839–1846.

[69] O. de Groot, B. Brito, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Scenario-based trajectory optimization in uncertain dynamic environments", *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5389–5396, 2021.

[70] P. O. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)", *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[71] I. Barbalat, "Systemes d'équations différentielles d'oscillations non linéaires", *Rev. Math. Pures Appl*, vol. 4, no. 2, pp. 267–270, 1959.

[72] D. Angeli, "A lyapunov approach to incremental stability properties", *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.

[73] J. Köhler, M. A. Müller, and F. Allgöwer, "Nonlinear reference tracking: An economic model predictive control perspective", *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 254–269, 2018.

[74] J. Nubert, "Learning-based approximate model predictive control with guarantees", M.S. thesis, ETH Zürich, Zürich, Switzerland, 2019.

[75] M. Dubied, A. Lahr, M. N. Zeilinger, and J. Köhler, "A robust and adaptive MPC formulation for gaussian process models", *arXiv preprint arXiv:2507.02098*, 2025.

[76] S. Yu, M. Reble, H. Chen, and F. Allgöwer, "Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control", *Automatica*, vol. 50, no. 9, pp. 2269–2280, 2014.

[77] D. Q. Mayne, E. C. Kerrigan, E. Van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control", *International journal of robust and nonlinear control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[78] F. Bayer, M. Bürger, and F. Allgöwer, "Discrete-time incremental ISS: A framework for robust NMPC", in *2013 European control conference (ECC)*, IEEE, 2013, pp. 2068–2073.

[79] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust feedback motion planning via contraction theory", *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 655–688, 2023.

[80] J. Köhler, M. N. Zeilinger, A. Carron, and L. Hewing, *Advanced model predictive control - nonlinear robust MPC - part 1 [lecture notes]. Intelligent Control Systems, ETH Zürich*. 2024.

[81] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

[82] D. L. Marruedo, T. Alamo, and E. F. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties", in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, IEEE, vol. 4, 2002, pp. 4619–4624.

[83] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska, "Robust MPC via min-max differential inequalities", *Automatica*, vol. 77, pp. 311–321, 2017.

[84] G. Pin, D. M. Raimondo, L. Magni, and T. Parisini, "Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties", *IEEE Transactions on automatic control*, vol. 54, no. 7, pp. 1681–1687, 2009.

[85] I. R. Manchester and J.-J. E. Slotine, "Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design", *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.

[86] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems", *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[87] J. D. Schiller, S. Muntwiler, J. Köhler, M. N. Zeilinger, and M. A. Müller, "A lyapunov function for robust stability of moving horizon estimation", *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7466–7481, 2023.

[88] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments", *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4616–4623, 2021.

[89] M. Lodel, B. Brito, A. Serra-Gómez, L. Ferranti, R. Babuška, and J. Alonso-Mora, "Where to look next: Learning viewpoint recommendations for informative trajectory planning", in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 4466–4472.

[90] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear MPC for trajectory tracking applied to rotary wing micro aerial vehicles", *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463–3469, 2017.

[91] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight", *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.

[92] M. A. Santos, A. Ferramosca, and G. V. Raffo, "Nonlinear model predictive control schemes for obstacle avoidance", *Journal of Control, Automation and Electrical Systems*, vol. 34, no. 5, pp. 891–906, 2023.

[93] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments", *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[94] D. Saccani, L. Cecchin, and L. Fagiano, "Multitrajectory model predictive control for safe UAV navigation in an unknown environment", *IEEE Transactions on Control Systems Technology*, vol. 31, no. 5, pp. 1982–1997, 2023.

[95] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives", in *2012 American Control Conference (ACC)*, IEEE, 2012, pp. 4239–4244.

[96] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars", *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[97] M. Ibrahim, J. Matschek, B. Morabito, and R. Findeisen, "Hierarchical model predictive control for autonomous vehicle area coverage", *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 79–84, 2019.

[98] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[99] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments", in *Robotics Research: The 16th International Symposium ISRR*, Springer, 2016, pp. 649–666.

[100] L. Quan, Z. Zhang, X. Zhong, C. Xu, and F. Gao, "EVA-planner: Environmental adaptive quadrotor planning", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 398–404.

[101] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight", *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.

[102] M. M. Tobenkin, I. R. Manchester, and R. Tedrake, "Invariant funnels around trajectories using sum-of-squares programming", *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9218–9223, 2011.

[103] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing", in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 2397–2403.

[104] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight", *Science robotics*, vol. 6, no. 56, eabh1221, 2021.

[105] J. Li, M. Ran, and L. Xie, "Design and experimental evaluation of a hierarchical controller for an autonomous ground vehicle with large uncertainties", *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1215–1227, 2021.

[106] L. Numerow, A. Zanelli, A. Carron, and M. N. Zeilinger, "Inherently robust suboptimal MPC for autonomous racing with anytime feasible SQP", *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6616–6623, 2024.

[107] M. Krinner, A. Romero, L. Bauersfeld, M. Zeilinger, A. Carron, and D. Scaramuzza, "MPCC++: Model predictive contouring control for time-optimal flight with safety constraints", in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, Jul. 2024.

[108] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification", *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[109] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning", *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[110] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors", *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[111] M. K. M Jaffar and M. Otte, "Retracted: PiP-X: Online feedback motion planning/replanning in dynamic environments using invariant funnels", *The International Journal of Robotics Research*,

[112] M. M. Nicotra and E. Garone, "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems", *IEEE Control Systems Magazine*, vol. 38, no. 4, pp. 89–107, 2018.

[113] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multirate control: From low-level actuation to high-level planning", *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6627–6640, 2022.

[114] N. Csomay-Shanklin, A. J. Taylor, U. Rosolia, and A. D. Ames, "Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions", in *2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 2022, pp. 3732–3739.

[115] M. Chen *et al.*, "Fastrack: A modular framework for real-time motion planning and guaranteed safe tracking", *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5861–5876, 2021.

[116] D. Althoff, M. Althoff, and S. Scherer, "Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 3470–3477.

[117] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning", in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 387–394.

[118] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust tracking with model mismatch for fast and safe planning: An SOS optimization approach", in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*, Springer, 2020, pp. 545–564.

[119] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems", *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[120] E. AG, *FORCESPRO*, https://forces.embotech.com, 2014–2023.

[121] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs", *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020.

[122] E. Cervera, "Try to start it! the challenge of reusing code in robotics research", *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 49–56, 2018.

[123] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[124] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[125] P. J. Huber, "Robust estimation of a location parameter", in *Breakthroughs in statistics: Methodology and distribution*, Springer, 1992, pp. 492–518.

[126] S. Boyd, "Convex optimization", *Cambridge UP*, 2004.

[127] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms", in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 6235–6240.

[128] NXP Semiconductors. "HoverGames". (2020–2024), [Online]. Available: https://nxp.gitbook.io/hovergames (visited on 06/14/2024).

[129] G. Fragapane, R. De Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda", *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021.

[130] S. Hwang, I. Jang, D. Kim, and H. J. Kim, "Safe motion planning and control for mobile robots: A survey", *International Journal of Control, Automation and Systems*, vol. 22, no. 10, pp. 2955–2969, 2024.

[131] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Springer International Publishing, 2015.

[132] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances", in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, IEEE, vol. 2, 1998, pp. 1384–1389.

[133] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Dynamic tube MPC for nonlinear systems", in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 1655–1662.

[134] S. V. Raković, L. Dai, and Y. Xia, "Homothetic tube model predictive control for nonlinear systems", *IEEE Transactions on Automatic Control*, vol. 68, no. 8, pp. 4554–4569, 2022.

[135] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview", *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021.

[136] J. Lee, J. Feng, M. Humt, M. G. Müller, and R. Triebel, "Trust your robots! Predictive uncertainty estimation of neural networks with sparse gaussian processes", in *Conference on Robot Learning*, PMLR, 2022, pp. 1168–1179.

[137] A. McHutchon and C. Rasmussen, "Gaussian process training with input noise", *Advances in neural information processing systems*, vol. 24, 2011.

[138] L. Lindemann, Y. Zhao, X. Yu, G. J. Pappas, and J. V. Deshmukh, "Formal verification and control with conformal prediction", *arXiv preprint arXiv:2409.00536*, 2024.

[139] M. Lauricella and L. Fagiano, "Set membership identification of linear systems with guaranteed simulation accuracy", *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5189–5204, 2020.

[140] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2024.

[141] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework", in *Robot Operating System (ROS) The Complete Reference (Volume 1)*, Springer, 2016, pp. 595–625.

[142] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems", *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005.

[143] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight", *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.

[144] R. Verschueren *et al.*, "acados—a modular open-source framework for fast embedded optimal control", *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[145] E. D. Andersen and K. D. Andersen, "The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm", in *High performance optimization*, Springer, 2000, pp. 197–232.

[146] M. Baker, "Reproducibility crisis", *nature*, vol. 533, no. 26, pp. 353–66, 2016.

[147] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science", *Patterns*, vol. 4, no. 9, 2023.

[148] F. Bonsignorio and A. P. Del Pobil, "Toward replicable and measurable robotics research [from the guest editors]", *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 32–35, 2015.

[149] E. Cervera, "Run to the source: The effective reproducibility of robotics code repositories", *IEEE Robotics & Automation Magazine*, vol. 31, no. 2, pp. 125–134, 2023.

[150] B. A. Nosek *et al.*, "Promoting an open research culture", *Science*, vol. 348, no. 6242, pp. 1422–1425, 2015.

[151] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, "The fair guiding principles for scientific data management and stewardship", *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[152] F. Markowetz, "Five selfish reasons to work reproducibly", *Genome biology*, vol. 16, pp. 1–4, 2015.

[153] S. Zhou, L. Brunke, A. Tao, *et al.*, "What is the impact of releasing code with publications?: Statistics from the machine learning, robotics, and control communities", *IEEE Control Systems Magazine*, vol. 44, no. 4, pp. 38–46, 2024.

[154] H. E. Plesser, "Reproducibility vs. replicability: A brief history of a confused terminology", *Frontiers in neuroinformatics*, vol. 11, p. 76, 2018.

[155] S. N. Goodman, D. Fanelli, and J. P. Ioannidis, "What does research reproducibility mean?", *Science translational medicine*, vol. 8, no. 341, 341ps12–341ps12, 2016.

[156] The Turing Way Community, *The Turing Way: A handbook for reproducible, ethical and collaborative research*, version 1.0.2, Jul. 2022. DOI: 10.5281/zenodo.7625728.

[157] F. Bonsignorio, J. Hallam, and A. Del Pobil, "Defining the requisites of a replicable robotics experiment", in *RSS2009 Workshop on Good Experimental Methodologies in Robotics*, 2009.

[158] J. A. Hernández and M. Colom, "Repeatability, reproducibility, replicability, reusability (4R) in journals' policies and software/data management in scientific publications: A survey, discussion, and perspectives", *arXiv preprint arXiv:2312.11028*, 2023.

[159] P. Wegner, "Concepts and paradigms of object-oriented programming", *ACM Sigplan Oops Messenger*, vol. 1, no. 1, pp. 7–87, 1990.

[160] J. d. O. Guimaraes, "The object oriented model and its advantages", *ACM SIGPLAN OOPS Messenger*, vol. 6, no. 1, pp. 40–49, 1995.

[161] H. Yu, G. C. E. de Croon, and C. De Wagter, "Avoidbench: A high-fidelity vision-based obstacle avoidance benchmarking suite for multi-rotors", in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9183–9189.

[162] H. Krasowski and M. Althoff, "Commonocean: Composable benchmarks for motion planning on oceans", in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2022, pp. 1676–1682.

[163] C. A. Dimmig *et al.*, "Survey of simulators for aerial robots: An overview and in-depth systematic comparisons [survey]", *IEEE Robotics & Automation Magazine*, vol. 32, no. 2, pp. 153–166, 2024.

[164] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars", in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, IEEE, 2021, pp. 62–70.

[165] Y. Li *et al.*, "Choose your simulator wisely: A review on open-source simulators for autonomous driving", *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 5, pp. 4861–4876, 2024.

[166] S. Kato *et al.*, "Autoware on board: Enabling autonomous vehicles with embedded systems", in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 287–296.

[167] S. Ochs *et al.*, "One stack to rule them all: To drive automated vehicles, and reach for the 4th level", *arXiv preprint arXiv:2404.02645*, 2024.

[168] T. Baca *et al.*, "The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles", *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 26, 2021.

[169] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering", in *2017 22nd international conference on methods and models in automation and robotics (MMAR)*, IEEE, 2017, pp. 37–42.

[170] K. Mohta *et al.*, "Fast, autonomous flight in GPS-denied and cluttered environments", *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[171] F. Camara, C. Waltham, G. Churchill, and C. Fox, "Openpodcar: An open source vehicle for self-driving car research", *arXiv preprint arXiv:2205.04454*, 2022.

[172] A. Carron *et al.*, "Chronos and CRS: Design of a miniature car-like robot and a software framework for single and multi-agent robotics and control", *arXiv preprint arXiv:2209.12048*, 2022.

[173] L. Lyons, T. Niesten, and L. Ferranti, "DART: A compact platform for autonomous driving research", in *2024 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2024, pp. 129–136.

[174] N. Hyldmar, Y. He, and A. Prorok, "A fleet of miniature cars for experiments in cooperative driving", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3238–3244.

[175] U. Rosolia. "The berkeley autonomous race car (BARC) project". (2025), [Online]. Available: https://urosolia.github.io/BARC_Project (visited on 03/24/2025).

[176] M. Boulet, O. Guldner, M. Lin, and S. Karaman. "MIT RACECAR". (2025), [Online]. Available: https://mit-racecar.github.io (visited on 03/24/2025).

[177] D. Pickem *et al.*, "The robotarium: A remotely accessible swarm robotics research testbed", in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1699–1706.

[178] J. Tani *et al.*, "Integrated benchmarking and design for reproducible and accessible evaluation of robotic agents", in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 6229–6236.

[179] A. Mokhtarian, P. Scheffe, S. Kowalewski, and B. Alrifaee, "Remote teaching with the cyber-physical mobility lab", *IFAC-PapersOnLine*, vol. 55, no. 17, pp. 386–391, 2022.

[180] Fly4Future. "AeroSTREAM open science: Open remote laboratory". (2025), [Online]. Available: https://fly4future.com/aerostream-open-remote-laboratory (visited on 03/24/2025).

[181] U. of Plymouth. "Autonomous systems test laboratory: Supporting teaching, learning and research in robotic vehicles and control systems". (2025), [Online]. Available: https://www.plymouth.ac.uk/facilities/autonomous-systems-test-laboratory (visited on 03/24/2025).

[182] A. Mokhtarian *et al.*, "A survey on small-scale testbeds for connected and automated vehicles and robot swarms: A guide for creating a new testbed", *IEEE Robotics & Automation Magazine*, 2024.

[183] A. Jiménez-González, J. R. Martinez-de Dios, and A. Ollero, "Testbeds for ubiquitous robotics: A survey", *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1487–1501, 2013.

[184] European Robotics Network. "GEM guidelines". (2025), [Online]. Available: http://www.heronrobots.com/EuronGEMSig/downloads/GemSigGuidelinesBeta.pdf (visited on 03/24/2025).

[185] Z. Wang *et al.*, "All robots in one: A new standard and unified dataset for versatile, general-purpose embodied agents", *arXiv preprint arXiv:2408.10899*, 2024.

[186] B. Boroujerdian, H. Genc, S. Krishnan, W. Cui, A. Faust, and V. Reddi, "Mavbench: Micro aerial vehicle benchmarking", in *2018 51st annual IEEE/ACM international symposium on microarchitecture (MICRO)*, IEEE, 2018, pp. 894–907.

[187] F. Amigoni *et al.*, "Competitions for benchmarking: Task and functionality scoring complete performance assessment", *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 53–61, 2015.

[188] F. Lier, P. Lücking, J. De Leeuw, S. Wachsmuth, S. Šabanović, and R. Goldstone, "Can we reproduce it? toward the implementation of good experimental methodology in interdisciplinary robotics research", in *ICRA 2017 workshop on reproducible research in robotics: current status and road ahead*, 2017.

[189] Delft University of Technology. "Mobile Robotics Lab". (2024–2025), [Online]. Available: https://www.tudelft.nl/me/over/afdelingen/cognitive-robotics-cor/research/mobile-robotics-lab (visited on 10/10/2025).

[190] D. Thomas and A. Hunt, *The Pragmatic Programmer: your journey to mastery*. Addison-Wesley Professional, 2019.

[191] D. Martinez-Baselga, O. de Groot, L. Knoedler, J. Alonso-Mora, L. Riazuelo, and L. Montano, "Hey robot! Personalizing robot navigation through model predictive control with a large language model", in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025, pp. 11 002–11 009.

[192] D. van der Heijden, "Designing simulators for robot learning", Dissertation, Delft University of Technology, 2025. DOI: 10.4233/uuid:d41281cc-d8cc-450d-b863-2a41d9d4a203.

[193] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "EGO-planner: An ESDF-free gradient-based local planner for quadrotors", *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[194] O. de Groot, "Probabilistic motion planning in dynamic environments", Dissertation, Delft University of Technology, 2024. DOI: 10.4233/uuid:c809ede8-3b9a-41c6-9b16-9a8a430351c8.

[195] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression", *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[196] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 1. MIT press Cambridge, 1998, vol. 1.

[197] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search", in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 528–535.

[198] L. Brunke *et al.*, "Safe learning in robotics: From learning-based control to safe reinforcement learning", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[199] S. W. Chen, T. Wang, N. Atanasov, V. Kumar, and M. Morari, "Large scale model predictive control with neural networks and primal active sets", *Automatica*, vol. 135, p. 109 947, 2022.

[200] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control", *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.

[201] K. Wang, T. Li, L. Xu, Q. Hu, and K. You, "Safety meets speed: Accelerated neural MPC with safety guarantees and no retraining", *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11 411–11 418, 2025.

[202] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems", *Automatica*, vol. 129, p. 109 597, 2021.

[203] K. P. Wabersich and M. N. Zeilinger, "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control", *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 2638–2651, 2022.

[204] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems", *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[205] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees", *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[206] P. M. Julbe, J. Nubert, H. Hose, S. Trimpe, and K. J. Kuchenbecker, "Diffusion-based approximate MPC: Fast and consistent imitation of multi-modal action distributions", *arXiv preprint arXiv:2504.04603*, 2025.

ACKNOWLEDGMENTS

Ten years at TU Delft, what a journey... One thing is certain: time really flies when you are having fun. Reaching this moment feels like a good time to look back on everything that has happened over de past decade, specifically during my PhD: the projects I have worked on, the incredible people I have had the pleasure to meet, and the friendships that have grown along the way. After finishing my bachelor's in Electrical Engineering and master's in Embedded Systems, I was lucky enough to be offered the chance to apply for this PhD position on safe mobile robot navigation in collaboration with the Dutch National Police. This collaboration was quite new at the time, at least for me, with an almost overwhelming range of possible directions to explore. Now, about five years later, I am happy with the choices I made and the results that came out of this project.

Of course, this would not have been possible without the people around me. My researcher mindset even has its affect on this section, because I am trying hard to be as complete as possible. But after meeting, working with, and enjoying life together with so many people throughout the PhD, it is nearly impossible not to forget someone. If your name is not mentioned, I am truly sorry. Feel free to mentally add yourself to this text. For everyone, I hope you will read on until the very last sentence. Each person below deserves to be acknowledged!

First of all, I want to thank some close collaborators from the Dutch National Police, in particular **Marius**, **Klaas Jan**, **Paul**, and **Rob**. I have really enjoyed our conversations and I am happy that, already during the PhD journey, you offered me a position in the group where I currently work with great pleasure. I would also like to thank the other people from the Dutch National Police who made it possible to bring robotics research a step further, not only for me, but also for other researchers in the department.

A PhD is a hard journey to complete. Therefore, I honestly appreciate the supervision provided by you, Laura, Javier, and Robert. **Laura**, thank you for guiding me through this journey. Your trust in me and the academic lessons you have taught me are invaluable. This is something I discover over and over again. I have respect for the fact that you gave birth to a baby and continued your academic career. **Javier**, thank you for being my supervisor as well. You have contributed to my academic attitude and provided me the opportunity to be part of a research group, which resulted in valuable and joyful discussions with the other skilled researchers in the group. **Robert**, thank you for being my supervisor too and providing relevant feedback during the journey. I remember our conversations during the last phase of my PhD which inspired me to keep up curiosity, to not only to keep learning about technical systems but also about life in general.

Without the committee I would not be able to defend this work. Thank you for providing valuable feedback on my thesis **Martijn Wisse**, **Guido de Croon**, **Timm Faulwasser**, and **Jaap Knotter**. Thanks to you too, **Jens Kober**, for your willingness to be reserve member in the committee. A big thanks to my paranymphs, **Simone** and **Thijs**, for standing by my side and for your help in organizing my defense day. And thank you,

Fae, for creating the beautiful cover design. I am impressed by your design skills. I also want to acknowledge the effort of the reviewers of my academic submissions. I know an academic career can be stressful. Thank you for taking the time to read and process my submissions and provide detailed feedback. At this place, it also feels appropriate to appreciate the world-wide development of chat agents, in particular Bing chat, that saved me multiple weeks of web searches, gave me intuition of specific mathematical equations, and helped me polish parts of the text in this thesis.

Now, I want to thank several close collaborators. First of all, thank you **Max**, for being my direct colleague during this research project. I enjoyed our inspiring daily conversations with topics ranging from research to societies to psychology. Also, thank you for collaborating on building the lab and preparing our project demos. Yes I agree, we can be proud of that! The lab built-up was also successful because of the serious effort put in it by you, Thijs. Thanks a lot **Thijs**, your positive energy came at the right time for me during the PhD. I will never forget this energy in our first online meeting during my stay in Zürich for a summer school. I have really enjoyed working together on our quadrotor platform and the real-time embedded MPC code. I will also never forget the moment we flew our quadrotor for the first time in our lab: fantastic! I also want to thank you, **Tasos**, for showing your interest in the methods I developed and for collaborating on a few projects of these methods applied to ASVs. I have enjoyed our conversations about the MPC code and life in general. And also, **Oscar**, thanks for your help at the start of my PhD, for our collaboration, and for just being a nice person to talk to. Also, **Sihao**, I am glad you joined our research group as self-funded postdoc. Thanks for the joyful informal conversations and for introducing me to the Agilicious framework. I also want to thank you for being a good example of how to find the tricky balance between (1) publishing high-impact research papers while (2) maintaining intuition for the operation of real quadrotor systems and their practical implementations while (3) providing above average guidance to master and bachelor students while (4) being just a nice person to talk to and collaborate with, a unique skillset. Last, but certainly not least, I want to thank you, **Johannes**, for the incredibly valuable insights you have provided me during our collaborative projects. Our collaboration was special, since we have never met in person until this time of writing. I look forward to the moment we have planned to meet in person: the day before my PhD defense. I am glad you are able to come to Delft. Despite the online collaboration, we managed to keep productivity high and get the T-RO paper published. I am incredibly grateful for meeting you and I really enjoyed working on the projects together. Please keep up the robotics research, I believe it will eventually serve many people.

Although a lot of time is spent on reading, implementing, and writing during the PhD, I have had countless work-related and informal conversations with colleagues. Be prepared to read a list of colleagues from the Cognitive Robotics department whom I had the pleasure to meet during my PhD. In particular, I want to thank you, **Luzia**. Your passion for research is contagious and your willingness to learn the Dutch language is admirable. I want to thank you for providing me the opportunity to join Linda and you on a trip in the United States. I have really enjoyed sightseeing with the three of us. Thanks **Linda**, for being there on the trip, for our nice conversations in the corridor and sharing your passion for music. I am still looking forward to watching the trip video. **Saray**, thank

you for your enthusiasm in the office, and your proactive attitude in organizing delicious dinners at your place. Also, thank you for keeping up the number of sandwiches during the lunch breaks in the social room. The Dutch culture deserves a place in the department. Thank you, **Anna**, for your approachable, honest, and open attitude to discuss different matter ranging from the PhD to hikes across Europe. I really enjoyed our conversations during the breaks. Thanks **Khaled**, for our joyful conversations during the summer school in Zürich and for the day-to-day fun we had. **Andreu**, thanks for being an office mate towards the end of my PhD and for sharing all kinds of interesting facts about literally anything in the world. **Clarence**, I was happy to be your office mate as well. Thanks for giving advice on research and quadrotor-related topics, and for sharing special Chinese candies and teas. Thanks, **Álvaro**, for the genuine conversations we had from time to time. And thanks for helping me building the very first HoverGames platform, I enjoyed the process of doing this together. Also thanks to you, **Julian**, not only for serving many delicious cakes, but also for seriously taking over the role as PhD representative in the department. Thanks **Giovanni**. GPs are cool! That is what I will remember from my interactions with you, I hope I understood well. You came across so passionate and I am sure you have helped countless researchers and students with your positive and ambitious attitude. Thanks to you, **Lasse**, for sharing your thought-through view of the world, research and what not. Your intrinsic motivation for research and engineering seems so strong that I have no doubt you will reach far in your future career. Thanks **Lorenzo**, for the conversations we had from time to time, about solvers, real-time code, and hardware experiments. Thanks **Patrik**, for the nice conversations we had when you visited our department and for showing me around for one weekend after the summer school in Prague. I will never forget these experiences. Also thanks to **Ajith**, **Alex** (2x), **Amy**, **Andrés**, **Arkady**, **Ashwin**, **Barys**, **Bas**, **Bence**, **Björn**, **Bruno**, **Burak**, **Carlos**, **Chadi**, **Chuhan**, **Corrado**, **Cosimo**, **Daniel** (2x), **David**, **Diego**, **Ebrahim**, **Ekaterina**, **Elia**, **Forough**, **Gillian**, **Gustavo**, **Hai**, **Holger**, **Italo**, **Irene**, **Irshadh**, **Jelle**, **Jens**, **Joris**, **Julian**, **Juri**, **Karin**, **Khaldon**, **Laurence**, **Manuel**, **Mariano**, **Martin**, **Max (Spahn)**, **Maximilian** (2x), **Micah**, **Nicky**, **Nils**, **Olger**, **Renchi**, **Rodrigo**, **Roel**, **Sagar**, **Thijs**, **Tom**, **Tomás**, **Tomáš**, **Vishal**, **Vít**, **Xinwei**, **Yasemin**, **Yke**, **Yujie**, **Zhaochong**, **Zhaoting**, and the other CoR colleagues for bringing in positive energy in the department.

The department would have a hard time continuing its work without the following people. Starting with the technical support, **André**, **Maurits**, **Kseniia**, and **Thomas**, I had a great time building the lab with you. I want to thank **Gijs**, **Ronald**, and **Mario** as well for their willingness to give software engineering advice for my work. A big thanks goes to the secretariat, **Hanneke**, **Noortje**, **Ellen**, **Loulou**, and **Arlette**, and department managers **Rosanne**, **Nancy**, and **Jolanda**, for your unconditional support to everyone in the department. I have always appreciated your effort. Thanks **Karin**, your organisational effort for the master robotics is admirable. I enjoyed our conversations in the corridor and at the coffee machine. Also a big thanks to the department heads, **Hans** and **Martijn**, for leading the department and acknowledging the engineering effort necessary to generate high-impact results. A special thanks to you, **Chris**, for the informal motivational talks that gave me a positive boost every single time we chatted. Your positive energy is really attractive, and I am sure it will help lots of other people.

During my PhD, I have enjoyed the guidance of several students. First of all, I am

grateful to have had the possibility to supervise you during your master thesis projects, **Johanna**, **Michiel**, and **Simone**. You have provided me with so many new insights, both on a theoretical and on a personal level. You have also allowed me to grow as a supervisor, to learn how to set up these projects and to bring them to a successful end, together with you. Second, I want to thank all bachelor students whom I had the privilege to supervise over the years: **Bobby**, **Christiaan**, **Cyp**, **Danish**, **Diederick**, **Emma**, **Ewout**, **Fabian**, **Gijs**, and **Pieter-Jan**. In this context, I want to thank you, **Giovanni**, **Bas**, and **Max**, for your enthusiastic co-supervision. I have enjoyed being in this process together with you. I also want to thank the following honours students for their enthusiasm: **Adam**, **Alejandro**, **Alfonso**, and **Vaclav**. Finally, I want to thank the master robotics students I mentored during the first year: **Abel**, **Alon**, **Amin**, **Emma**, **Luoqi**, **Max**, **Pim**, **Severin**, and **Vassil**. I sometimes come across updates of your lives as professionals. It fascinates me, I wish you all the best. And who knows, maybe our paths cross again at some point.

I would also like to thank the people I met because of the collaboration with the Dutch National Police: thank you **Alexander**, **Bas**, **Bianca**, **Irene**, **Isabelle**, **Jelte**, **Maaike**, **Marina**, **Ron**, and **Theo**. The National Police AI lab gathering would not have been possible without your support from TU Delft side, thank you **Celine**, **Moniek**, and **Will**. Furthermore, I want to thank the people from the police who joined for the police writing days: **Coen**, **Geer**, **Robert**, and **Sophie**.

Gedurende mijn PhD is er een groep mensen geweest die ik met regelmaat heb gezien in mijn vrije tijd. Onder deze groep mensen vallen vrienden van de bachelor elektrotechniek: **Bastiaan**, **Casper**, **Matti**, and **Remon**. Thanks jongens, ik geniet elke keer weer van onze halfjaarlijkse diners en vind het interessant om te horen waar jullie nu mee bezig zijn. Ook wil ik graag alle leden van koor **Twentysomething** bedanken voor de mooie momenten die we met elkaar hebben meegemaakt gedurende ons bestaan (okee, behalve de allereerste repetitie voor mij dan). In het bijzonder wil ik graag iedereen bedanken die heeft deelgenomen aan de intermezzo's, commissies en activiteiten waar ik bij betrokken was. Ik vind het erg bijzonder hoe onze groep ondanks, maar misschien wel dankzij, de groei en diversiteit zo hecht blijft. Ik kijk uit naar nog vele optredens samen met veel muzikaliteit! Ook wil ik graag iedereen van korfbalvereniging **KCC** bedanken, in het bijzonder iedereen die heeft meegespeeld in de teams KCC 3 en 5 in de jaren dat ik daar actief was. Helaas heb ik halverwege mijn PhD (tijdelijk) besloten te stoppen met korfbal ten goede van de succesvolle afronding van mijn PhD. Echter, in de jaren dat ik nog speelde heeft het me altijd erg veel energie gebracht. Hier ben ik me de laatste jaren des te meer bewust van geworden. Dank jullie wel daarvoor!

Nu wil ik graag twee goede vrienden bedanken. Dank je wel, **Tobias**. We kennen elkaar al van jongs af aan. Het is inspirerend om te zien hoe onze wegen andere kanten op zijn gegaan. Desondanks klikt het meteen weer als we elkaar spreken. Ik ben onder de indruk van jouw down-to-earth mentaliteit en jouw kunde om je doelen na te streven, ook al gaan dingen niet zoals je ze gepland had. Het helpt mij om mijn eigen uitdagingen in perspectief te plaatsen. Ik kijk uit naar nog vele gesprekken onder het genot van een drankje! En dank je wel, **Bastiaan**. Dank je wel voor onze goede vriendschap sinds het eerstejaarsweekend van de bachelor. I geniet ervan om met jou te praten over van alles en nog wat in het leven, vooral op het vlak van muziek. Ik kijk uit naar veel meer

improvisatiesessies en andere muzikale samenwerkingen!

Last, but certainly not least, wil ik graag mijn **familie** bedanken. Jullie zijn een constante factor geweest gedurende mijn leven. Ik geniet van de grappen die gemaakt worden en serieuze gesprekken die gevoerd worden tijdens momenten van samenkomst door het jaar heen. In het bijzonder wil ik u, **oma Jeanne**, bedanken. Dank u wel voor de mooie herinneringen die we samen hebben opgebouwd. Uw onvoorwaardelijke interesse in alles wat ik meemaak vind ik heel bijzonder. Ik hoop dat ik op zijn minst een deel daarvan terug kan geven aan u. **Simone**, jou wil ik ook graag bedanken. Onze band als broer en zus vind ik heel bijzonder, omdat we zoveel met elkaar kunnen delen. Dit betreft ook zeker mijn PhD. Ik vind het heel fijn hoe je hier op veel momenten in mee hebt gedacht. Dank je wel hiervoor, ik hoop samen nog vele mooie herinneringen te kunnen maken! Als laatste, **pap** en **mam**, ik weet eerlijk gezegd niet goed hoe ik jullie ooit kan bedanken voor alles wat jullie voor me betekenen. De onvoorwaardelijke liefde en steun die jullie aan Simone en mij geven is onbeschrijflijk. Tijdens mijn PhD heb ik veel gesprekken met jullie gevoerd over alle uitdagingen die op mijn pad kwamen. Hier open over kunnen praten en jullie vertrouwen in mij hebben ervoor gezorgd dat ik de eindstreep heb gehaald. Simone, pap en mam, deze PhD heb ik gehaald dankzij jullie steun. Daarvoor blijf ik jullie voor altijd dankbaar!

Dennis
Krimpen aan den IJssel, January 2026

CURRICULUM VITÆ

Dennis Benders was born in November 1997 in Krimpen aan den IJssel, The Netherlands. He obtained the B.Sc. degree in Electrical Engineering and M.Sc. degree in Embedded Systems from the Delft University of Technology, Delft, The Netherlands, in 2018 and 2020, respectively. In January 2021, he started a Ph.D. as part of the Reliable Robot Control and Autonomous Multi-Robots research labs in the department of Cognitive Robotics at the Delft University of Technology. During his Ph.D, he worked on autonomous and safe hierarchical model predictive control algorithms for safe motion planning and control on mobile robots under supervision of Dr. Laura Ferranti, Prof. dr. Javier Alonso-Mora, and Prof. dr. Robert Babuška. The research project was executed in collaboration with one of the robotics groups at the Dutch National Police, where he is a robotics engineer since September 2025.

His research interests include motion planning and control for autonomous mobile robots, with special emphasis on real-time and open-source code.

List of Publications

Publications

1. D. Benders, J. Köhler, T. Niesten, R. Babuška, J. Alonso-Mora and L. Ferranti, "Embedded Hierarchical MPC for Autonomous Navigation", in IEEE Transactions on Robotics, vol. 41, pp. 3556-3574, 2025, doi: 10.1109/TRO.2025.3567529.

2. D. Benders, L. Ferranti, and J. Köhler, "A step-by-step guide on nonlinear model predictive control for safe mobile robot navigation", *arXiv preprint arXiv:2507.17856*, Aug. 2025.

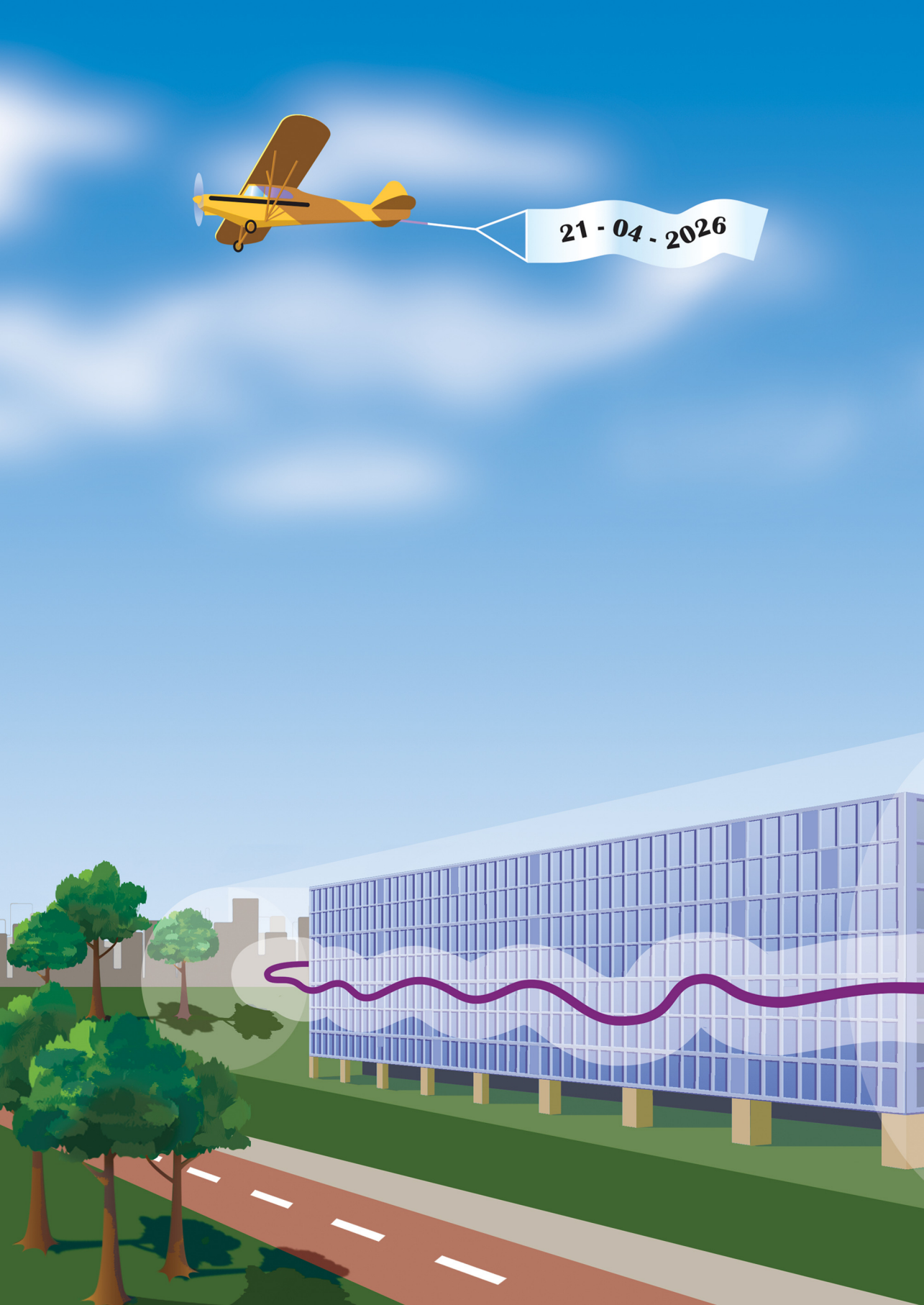
3. D. Benders, J. Köhler, R. Babuška, J. Alonso-Mora and L. Ferranti, "From Data to Safe Mobile Robot Navigation: An Efficient and Modular Robust MPC Design Pipeline", *arXiv preprint arXiv:2508.07045*, Aug. 2025, under review.

4. D. Benders, Reproducibility in robotics 2025. [Online]. Available: https://github.com/dbenders1/reproducibility-in-robotics-2025.

Co-authored Publications

1. F. Bos, A. A. Meera, D. Benders and M. Wisse, "Free Energy Principle for State and Input Estimation of a Quadcopter Flying in Wind", 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 5389-5395, doi: 10.1109/ICRA46639.2022.9812415.

2. A. Tsolakis, D. Benders, O. De Groot, R. Negenborn, V. Reppa, and L. Ferranti, "Colregs-aware trajectory optimization for autonomous surface vessels", IFAC-PapersOnLine, vol. 55, no. 31, pp. 269–274, 2022, doi: 10.1016/j.ifacol.2022.10.441.


21 - 04 - 2026