Title

Safe Control for Mobile Robots via Reference Governor Techniques

Permalink

https://escholarship.org/uc/item/79f4m63f

Author

Li, Zhichao

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Safe Control for Mobile Robots via Reference Governor Techniques

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Zhichao Li

Committee in charge:

Professor Nikolay Atanasov, Chair
Professor Massimo Franceschetti
Professor Sylvia Herbert
Professor Sonia Martinez Diaz
Professor Yang Zheng

2023

The Dissertation of Zhichao Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To my parents and wife. Thanks for their constant love and support.

TABLE OF CONTENTS

Dissertation Approval Page iii

Dedication iv

Table of Contents v

List of Figures viii

List of Tables x

Acknowledgements xi

Vita xiii

Abstract of the Dissertation xiv

Chapter 1 Introduction 1
1.1 Motivation 1
1.2 Related Works 3
1.2.1 Graph Search Algorithms 3
1.2.2 Safe Control and Trajectory Tracking 4
1.2.3 Dynamics Learning 7
1.2.4 Environmental Adaptive Motion Control and Planning 8
1.3 Overview 9

Chapter 2 Safe Output Tracking via Reference Governor 10
2.1 Problem Formulation 10
2.2 Output Regulation without Constraints 11
2.3 Safe Output Regulation via PBF 13
2.4 Safe Output Tracking using a Reference Governor 14
2.5 Evaluation 20
2.6 Summary 25

Chapter 3 Robust and Safe Autonomous Navigation for Systems with Learned SE(3) Hamiltonian Dynamics 26
3.1 Overview 26
3.2 Problem Statement 27
3.3 Learning $SE(3)$ Hamiltonian Dynamics from Data 29
3.3.1 Translation-Equivariant $SE(3)$ Hamiltonian Dynamics Learning 30
3.3.2 Model Estimation Error as a Disturbance 31
3.4 Stabilization of Hamiltonian Dynamics with Matched Disturbances 34
3.4.1 Passivity-based Control 34
3.4.2 Robustness Analysis 37

3.4.3 Safety Analysis 39
3.5 Safe and Stable Tracking using a Reference Governor 41
3.6 Application to Hamiltonian Dynamics in $\mathbb{R}^n$ 45
3.7 Evaluation 47
3.7.1 Evaluation of $SE(3)$ Hamiltonian Dynamics Learning 49
3.7.2 Evaluation of Robust Safe Tracking Control of a Learned 2D Hexarotor Hamiltonian Model 49
3.7.3 Evaluation of Robust Safe Tracking Control of a Learned 3D Fully-actuated Hexarotor Hamiltonian Model 52
3.7.4 Evaluation of Robust Safe Tracking Control of a Learned 3D Quadrotor Hamiltonian Model 54
3.7.5 Evaluation of Our Approach against Unmodeled Noise 56
3.8 Summary 57

Chapter 4 Environment Aware Safe Tracking using Planning and Control Co-Design . 59
4.1 Overview 59
4.2 Directional Metric and Motion Prediction 60
4.2.1 State-Dependent Directional Metric 60
4.2.2 Trajectory Bounds Estimation for LTI System 62
4.2.3 Trajectory Bounds Estimation for Unicycle-like Nonlinear Systems 65
4.3 Dynamic Safety Margin using SDDM 68
4.3.1 DSM from Ellipsoid Trajectory Bounds 68
4.3.2 DSM from Reachable Set Approximation 69
4.3.3 DSM for Adaptive Control Law 70
4.3.4 Evaluation of Dynamic Safety Margin 71
4.4 Safety Aware Planning 74
4.4.1 Costmap Design 75
4.4.2 Planner Edge Cost Design 76
4.4.3 Evaluation of Costmap Designs 78
4.5 Safe Tracking via Reference Governor 79
4.5.1 Governor Control Policy Design 80
4.5.2 Governor Control Policy Extension for Moving Obstacles 82
4.6 Evaluation 86
4.6.1 Hardware Overview. 86
4.6.2 Software Architecture 87
4.6.3 Experiment Setup and Parameters 88
4.6.4 Baseline Comparison 89
4.6.5 Hardware Demonstration in Clutter Unknown Environment 91
4.6.6 Moving Obstacles Experiments 92
4.7 Summary 94

Chapter 5 Concluding Remarks 97
5.1 Conclusion 97
5.2 Future Work 97

Appendix A Proofs of Chapter 2 98

Appendix B Proofs of Chapter 3 101
B.1 Proof of Theorem 3 101

Bibliography 106

LIST OF FIGURES

Figure 2.1. The structure of the reference-governor controller 15

Figure 2.2. A mobile robot with omnidirectional wheels 20

Figure 2.3. Simulation result of output-tracking control of an omnidirectional mobile robot navigating in an unknown environment 22

Figure 2.4. Simulation result of output-tracking control of an omnidirectional robot in an unknown environment 23

Figure 3.1. Architecture of $SE(3)$ Hamiltonian neural ODE network 32

Figure 3.2. Structure of the reference-governor tracking controller 41

Figure 3.3. Dynamics learning using $SE(3)$ Hamiltonian neural ODE network for hexarotor 46

Figure 3.4. Comparison of position prediction errors between our method vs. GP model 48

Figure 3.5. Path tracking comparison our approach vs. GP-MPC 48

Figure 3.6. Safe navigation of a hexarotor system using learned model in a warehouse 51

Figure 3.7. Tracking error of a hexarotor system 52

Figure 3.8. Tracking controller performance for hexarotor in warehouse simulation with the ground truth model subject to a disturbance 53

Figure 3.9. Safe navigation of quadrotor system using learned model in a warehouse . 55

Figure 3.10. Tracking error of a quadrotor system 56

Figure 3.11. Robustness result against unmodeled noise. 57

Figure 4.1. Directional metric vs symmetric metric 61

Figure 4.2. Trajectory bounds comparison between a Euclidean metric and a quadratic norm based SDDM 65

Figure 4.3. Cone shape reachable set prediction for closed-loop unicycle system 67

Figure 4.4. Corridor Simulation 71

Figure 4.5. Output peak curve along trajectory 72

Figure 4.6. Simulation of the robot-governor system navigate in an environment with circular obstacles 73

Figure 4.7. Snapshot of the robot-governor system navigating a cluttered maze 74

Figure 4.8. Simulation of robot-governor path following controller in C-shape environment 75

Figure 4.9. Jackal unmanned ground vehicle 76

Figure 4.10. Jackal costmap design 77

Figure 4.11. Jackal race world ROS simulation with three different cost curve designs. 79

Figure 4.12. Geometric relationship of local safe zone, local projected goal and optimized local goal 81

Figure 4.13. System architecture overview of environment aware safe tracking control. 87

Figure 4.14. Safety test on ten random selected goals. 90

Figure 4.15. The result maze adaptivity experiment 91

Figure 4.16. Hardware experiment in a large-scale collaborative working space 92

Figure 4.17. Costmap of a large-scale collaborative working space 93

Figure 4.18. Result of moving obstacle simulation with six dynamic obstacles at different running speed and sizes 94

Figure 4.19. Hardware experiment with Jackal robot and two moving obstacle 95

LIST OF TABLES

Table 4.1. Three different costmap design parameters. 79

Table 4.2. Simulation quantitative result of three costmap designs. 79

Table 4.3. Parameter table of control gains. 89

Table 4.4. Parameter table of control gains used in moving obstacle simulation. 93

ACKNOWLEDGEMENTS

I would like to first express my deepest gratitude to my advisor Professor Atanasov for his unwavering support and guidance throughout the entire journey of this dissertation. His constant encouragement has been the driving force behind my research and my motivation to excel. In addition, I extend my thanks to the members of my dissertation committee for their invaluable insights and constructive feedback. Their contributions have played a pivotal role in shaping this research.

I would also like to thank all my colleagues at Exisential Robotic Lab, without whom my research would have no doubt taken five times as long. It is their support that helped me in an immeasurable way. Special thanks to Thai Duong, for the intense insightful and discussions in research, helping me doing experiments in day and night and bring many joys using his humorous conversations. Thanks to Yinzhuang Yi and Zhuolin Niu, for spending heroic effort helping me finish my last project and drag me out from lab and hiking outdoors to keep me fit. Thanks to Zhirui Dai, thank you for helping me out with programming and answering stupid and challenging questions with great patience.

I also like to thank my friends at UC San Diego, Weijian Xu, Songting Xu, Pengluo Wang, Yangsheng Hu, Dan Li, Huan Yu. Thank you for helping me go through this long journey, cheer me up when I am down, help me out when I am in trouble or ill and share happiness in beautiful moments. I feel so grateful.

Finally, I would like to thank my parents and my wife for their constant support and unconditional love. Thank you for always believing in me, encouraging me, and giving me great freedom to explore my life. My accomplishment and success cannot happen without your love.

Chapter 2, in full, is a reprint of “Governor-parameterized barrier function for safe output tracking with locally sensed constraints” by Zhichao Li and Nikolay Atanasov, which appears in Automatica, Elsevier Press, 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 3, is based on material from the following papers. First, “Safe autonomous

navigation for systems with learned SE(3) Hamiltonian dynamics” authored by Zhichao Li*, Thai Duong* and Nikolay Atanasov, which appears in Learning for Dynamics and Control Conference (L4DC), 2022. The dissertation author was the co-primary investigator and author of this paper, (*equal contribution). Second, “Robust and Safe Autonomous Navigation for Systems with Learned SE(3) Hamiltonian Dynamics” by Zhichao Li, Thai Duong and Nikolay Atanasov, which appears in Open Journal of Control Systems, IEEE, 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 4, is based on material the following two papers. First, “Fast and safe path-following control using a state-dependent directional metric” authored by Zhichao Li, Omur Arslan and Nikolay Atanasov, which appears in International Conference on Robotics and Automation (ICRA), IEEE, 2020. The dissertation author was the primary investigator and author of this paper. Second, “EAST: Environment Aware Safe Tracking using Planning and Control Co-Design” by Zhichao Li, Yinzhuang Yi, Zhuolin Niu and Nikolay Atanasov, is currently being prepared for submission. The dissertation author was the primary investigator and author of this material.

VITA

2013	Bachelor of Engineering, Northwestern Polytechnical University
2016	Master of Science, Arizona State University
2016–2023	Research Assistant, University of California San Diego
2023	Doctor of Philosophy, University of California San Diego

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering (Intelligent Systems, Robotics, and Control)

Studies in Guidance Navigation and Control System
Professor Chao Xing

Studies in Control Systems
Professor Armando Rodriguez

Studies in Robotics
Professor Nikolay Atanasov

ABSTRACT OF THE DISSERTATION

Safe Control for Mobile Robots via Reference Governor Techniques

by

Zhichao Li

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2023

Professor Nikolay Atanasov, Chair

This dissertation considers the problem of safe navigation for autonomous mobile robots working in partially known and unknown environments with static and non-adversarial moving obstacles. Given a geometric path generated by standard path planner, we develop reference-governor based tracking control policy to continuous generate proper set-points along the path for downstream low-level stabilizing controller. This new method systematically puts planning, motion prediction and safety metric design together to achieve environmentally adaptive and safe navigation. Our algorithm balances optimality in travel distance and safety regarding passing clearance. Robots adapt progress speed adaptively according to the sensed environment, being fast in wide open areas and slowdown in narrow passages and taking necessary maneuvers to avoid dangerous incoming obstacles. Directional distance measure, motion prediction and custom costmap are integrated properly to evaluate system risk accurately with respect to local geometry of surrounding environments. Using such risk estimation, reference governor technique and control barrier function are worked together to enable adaptive and safe path tracking in dynamical environments. We validate our algorithm extensively both in simulations and hardware platforms in challenging real-world environments.

Chapter 1

Introduction

In recent years, there has been a rapid emergence of robot applications, thanks to advancements in sensing technology and computation power. Robot systems are becoming increasingly integrated into the human society in applications, such as autonomous transportation, factory inspection, cleaning services, and medical robotics. Reliable robot navigation in unstructured environments is a crucial aspect for enabling these services. As a result, there is a growing focus on researching and developing certifiably safe yet efficient navigation algorithms.

1.1 Motivation

There are extensive works on classical motion planning computing optimal paths that connect two states in a robot's work space or configuration space. However, only a few of these algorithms prioritize safety in defining optimality and the robot's dynamics are often ignored or greatly simplified. In algorithms like ARA* [70], RRT* [52], optimality is often measured in terms of travel distance neglecting robot dynamics. Path clearance is often set to a constant (bigger than the robot's circumscribed radius). If this constant is small, the resulting optimal paths usually hug obstacle boundaries tightly and introduce collision risk for downstream tracking controllers (due to model uncertainty and external disturbances). If the constant is large, it will cause a planner to return no path to areas connected by narrow passages. On the other extreme, maximum clearance [11] can provide the most safe path, however, a much larger travel distance may not be favorable for energy-efficiency reasons. Ideally, we want a planner that can balance energy (travel distance) and safety (path clearance) and is able to reach feasible regions without being overly close to obstacles.

In tracking controller design, the performance is usually measured in terms of tracking error and speed. Such design works reasonably well for machines that operate in known, controlled and static environments, like robot arm in manufacturer, airplane on the fly. But, there are many robots applications (for example, floor cleaning robot or autonomous cars) have to work in safe-critical and highly constrained dynamical environments. For these applications, solely using tracking error and speed as objective becomes insufficient and less-favorable. In nature, a intelligent agent tends to control its motion speed according to sensory data. For example, a cat can go through a narrow tunnel or passage cautiously while running fast in wide open area. An experienced driver will pay most attention to objects in front and adjust safe distance properly according to speed, meanwhile being alert to lateral cars but not overly-reacting. From these observations, we see that a good tracking control policy must can estimate risk quickly in real-time and adjust its motion speed adapt to local environments. Plus, while pursuing along the given path, it must be flexible to take necessary detour to reduce collision risk imposed by moving objects. We would like to robot being aware to surrounding environments and can run sufficiently fast when possible in wide-open area and slow down in highly constrained spaces. Furthermore, it needs to pay attention to nearby moving obstacles and able to avoid potential collision and resume to original task quickly.

To address above issues, we develop an environment aware and safe tracking by integrating planning and control design. First, we design a generic customizable cost function connecting planning states and integrated in A* planning algorithm. By using distance-based costmap, the planner generate natural (human-like) path balancing safety and clearance, favoring custom clearance in wide open space while being able to go through tight passage. Next, by adopting accurate motion prediction and directional metric, we estimate system running risk with respect to locally sensed environments. Using this estimation, we develop new adaptive reference-governor to track optimized path. It regulates robot motion speed and continuously generate proper signal for low level controller, fast in empty space and slow down when go through places while being reasonable fast in narrow passage. Finally, to handle moving obstacles, we use control barrier functions (CBF) to optimized reference signal (allowing it being away from given path) in minimally invasive fashion. Following this optimized goal, robot can be alert to incoming obstacle and take necessary detours maneuver. When gaining enough safety, robot can resume to original path quickly.

1.2 Related Works

1.2.1 Graph Search Algorithms

The problem of finding a feasible or optimal path (trajectory) connecting start and goal points (states) has been investigated extensively [62]. Sampling-based methods like Probabilistic Roadmap (PRM) [53], Rapidly-exploring Random Tree (RRT) [61] can quickly find a feasible path even in large scale or high-dimensional space. Some algorithm like RRT* [52] can asymptotically converge to the optimal solution. But the non-deterministic property due to sampling procedure and unpredictable intermediate result makes its usage limited especially when application has fixed planning time requirements. Search-based motion planning algorithms like A* [41] or jump point search (JPS) [40] can provide optimal path in a more consistent manner and planning time is predictable. However, search-based algorithms can face scalability issues in large-scale workspace or high-dimensional complex configuration space. If a hard planning time constraints exists, algorithms like ARA* [70] and RTAA [56] can be used. In most graph search algorithms, optimality is often equivalent to path length measured in workspace (for mobile robots) and safety is either ignored or simply considered in terms of fixed path clearance. To help further discussion, this type of graph search motion planning algorithm will be referred to geometric planner.

1.2.2 Safe Control and Trajectory Tracking

Model predictive control (MPC) [14, 39, 17, 95, 76] is a well established approach for constrained control that approximates an infinite-horizon optimal control problem with a sequence of finite-horizon problems. With a suitable choice of terminal cost and constraints for the finite-horizon problems, MPC guarantees recursive feasibility and asymptotic stability for linear systems subject to polytopic state and control constraints [14]. Even if the system and constraints are nonlinear, implicit MPC techniques solve the finite-horizon problems online using sequential quadratic programming approximation [39]. In the context of trajectory tracking, the safe set of system states is commonly approximated via polytopes, ellipsoids, or other convex regions and collision checking is performed using linearized system models. In [33], robust invariant sets are first computed offline, by treating the nonlinear part of the system dynamics as a bounded disturbance, and then used to tighten the state and input constraints in a nominal nonlinear MPC problem. Collision avoidance among multiple vehicles as well as static and dynamic obstacles is considered in [90]. It is shown that this problem can be stated as a receding-horizon linear program with mixed integer and linear constraints that account for the unsafe regions.

Reference governor control (RG) [10, 36, 37, 34, 58] is an approach for enforcing input and state constraints for a nonlinear system tracking a reference trajectory. Unlike MPC, this technique may directly use the nonlinear system dynamics. It assumes that a control law that stabilizes the system to an equilibrium point in the absence of constraints is available. A virtual system, called a reference governor, is used to adaptively decide the rate of change of the reference signal so that the pre-stabilized system can track it without violating the constraints. The governor update rule is a key component of the design, typically involving constrained scalar optimization [59]. A forward invariant set, such as the maximum state or output admissible set for linear systems [36] or a Lyapunov function level set for nonlinear systems [37], is required to solve the optimization problem efficiently. The explicit reference governor (ERG) formulation

[35] relies on Lyapunov function level sets to modify the reference governor velocity and enforce the constraints without resorting to any online optimization. With an appropriate choice of dynamic safety margin and navigation field, ERG control [78] can guarantee safe trajectory tracking with a priori known obstacles. ERG control can be conservative because it enforces constraints via Lyapunov function level sets, which implicitly requires all sublevel sets to be forward invariant. Our prior work [67] shows that directional trajectory bounds can be computed only for the actual initial state of a linear system, allowing fast and safe tracking control.

Reachability-based methods [19, 91, 21, 44] for safe control rely on precise reachable set approximations for each system state. There are many ways to compute reachable sets but funnels and Hamilton-Jacobi reachability techniques have been particularly effective in trajectory tracking applications. A *funnel* is a tight outer approximation of the reachable set around a reference trajectory. Sequential funnel composition [19] is a seminal work that composes controllers, by guaranteeing that the goal point of each lies within the domain of attraction of the next, to achieve complex system behaviors. Recent advances in sum-of-square (SOS) programming allow efficient and accurate computation of funnels and their successful application in safe navigation [96, 75]. Funnel computations are tied to a reference trajectory, requiring the design of a funnel library [75], which faces a trade-off between the number of pre-computed funnels and the computational complexity of composing them online. Contraction theory [91] generalizes the funnel design to allow adaptation to *any* feasible nominal trajectory online. Hamilton-Jacobi reachability techniques [8] approximate the complete reachable set of a nonlinear system by solving a partial differential equation using dynamic programming. Obtaining solutions for high-dimensional systems may be challenging without decomposition methods that split the system into several low-order subsystems [21]. FaSTrack [44] is a safe motion planning framework that employs a safe controller with a pre-computed tracking error function. The tracking error function is defined over the relative state between a simplified planning model and an accurate dynamics model, engaged in a pursuit-evasion game. Then, a feasible path may be computed efficiently online using the simplified model while augmenting the obstacles with the worst-case tracking error as a safety margin. Reachability-based Trajectory Design (RTD) [60] adapts system decomposition techniques [21] to compute forward reachable sets that provably bound the tracking error. Persistently feasible trajectories, ensuring the existence of a new safe trajectory before a previous one is completed, are generated online based on a discrete finite-set obstacle approximation.

Control barrier functions (CBF) [4, 5, 2] enable complex safety constraints that remain linear in the control input of a control-affine nonlinear system. As mentioned earlier, this leads to an elegant CLF-CBF-QP formulation, successfully utilized in safe-critical applications [4, 15, 3, 99, 77]. CBF techniques are applied to trajectory tracking in [9, 93]. Barry et al. [9] use SOS optimization to construct multiple polynomial barrier functions to ensure collision-free navigation for an unmanned aerial vehicle, with potentially non-smooth transitions among different reference trajectories. The complexity of this CBF construction, however, increases drastically with the number of obstacles because each obstacle requires an additional barrier function. Collision avoidance among multiple agents in the absence of other obstacles is considered in [93]. A CBF is constructed by computing the infimum of a safety function along closed-loop future trajectories, with the help of a pre-designed evading maneuver strategy. Recently, a predictor-corrector collision avoidance algorithm for multiple systems was developed in a MPC framework without explicit communication among the systems [89]. The host agent, executing the algorithm, treats the differences between the predicted and observed control actions of other agents as disturbances for a robust CBF-based quadratic program. Existing CBF techniques, however, have not considered safe trajectory tracking in unknown cluttered environments with arbitrary shape obstacles. Recent work proposes a support vector machine learning approach [94] to classify safe and unsafe regions and synthesize a CBF constraint from online sensor data.

Kinodynamic motion planning is another popular approach that trying to solve the planning and control task at the same time. For example, [83] use linear-quadratic-regulation (LQR) cost in steering function design and optimal control is used in [98] to connecting sampling states. However, as pointed out by [82], locally optimal control, does not necessarily lead to global optimality. Moreover, these methods often use simplified system dynamics to reduce computation requirement. The potential tracking error due to model discrepancy can lead to unsafe behavior.

1.2.3 Dynamics Learning

For safe critical applications, an accurate robot model is always preferred. This motivated data-driven dynamics learning approaches, utilizing machine learning techniques, such as Gaussian process (GP) regression [27, 51, 45] and neural networks [85, 23]. For physical systems, recent works [74, 102, 28] design the model architecture to impose a Lagrangian or Hamiltonian formulation of the dynamics [73, 47], which a black-box model might struggle to infer. For Lagrangian dynamics, Lutter et al. [74] use neural networks to represent the mass and potential energy in the Euler-Lagrange equations of motion. Meanwhile, Zhong et al. [102] use a differentiable neural ODE solver [22] to predict state trajectories of a Hamiltonian dynamics model, encoding Hamilton's equations of motion. A trajectory loss function is back-propagated through the ODE solver to update the Hamiltonian model parameters. Our prior work [28] extends the neural ODE Hamiltonian formulation by imposing $SE(3)$ constraints to capture the kinematic evolution of rigid-body systems, such as ground or aerial robots. A Hamiltonian-based model architecture also allows the design of stable regulation or tracking controllers by energy shaping [102, 29, 28]. Interconnection and damping assignment passivity-based control (IDA-PBC) [97], one of the main approaches for energy shaping, injects additional energy into the system via the control input to achieve a desired total energy, which is minimized at a desired regulation point. Instead of learning robot dynamics in continuous time, Saemundsson et al. [88] design a variational integrator network to learn discrete-time Lagrangian dynamics. Havens and Chowdhary [42] extend it by including control input in the model and use model predictive control for stabilization.

1.2.4 Environmental Adaptive Motion Control and Planning

In the motion planning algorithm mentioned above, safety is considered as hard constraints. In safe tracking and planning works mentioned above, only system dynamics are considered (in motion prediction). To achieve navigation behavior like smart agents, one needs to develop environmental aware safety metric and integrate in planner and control design.

As mentioned in introduction, a good risk metric should jointly consider system motion and surrounding environment. The author of [6] constructs a dynamic safety metric (DSM) [35] using the difference of distance to obstacle and Lyapunov function. Using this DSM, a navigation algorithm is developed using reference-governor techniques for a second order fully-actuated system. In [67], the author develops a reference-governor path tracker using time-varying ellipsoid motion predictions instead of using Lyapunov function induced invariant set. By embedding directional preference and aligning it with given geometric path using state-dependent directional metric (SDDM), the tracker pays less attention to lateral obstacles and achieves better performance in corridor-like narrow passages. However, due to front-back symmetry, ellipsoid shape trajectory bound prevents robot gaining speed when leaving obstacle-dense area. To overcome this, EVA-planner [84] proposes environmental adaptive safety aware (EASA) by evaluating angle difference using inner product of gradient of Euclidean signed distance filed (ESDF) and robot motion direction. However, the volume of free space is neglected which might result in path with insufficient clearance [71]. In this paper, we use the distance between an accurate cone shape motion prediction [50] and inflated obstacle space in costmap to serve as our DSM in governor design. Furthermore, by comparing directional DSM and regular, we construct a SDDM-base gain to speed up robot longitudinal motion. The resulting tracker can go fast in narrow passages and quickly speed up when leaving obstacle dense area.

1.3 Overview

The main contribution of this dissertation is developing a safe and adaptive tracking controller design framework based on reference governor techniques. In specific, at Chapter 2, we develop an adaptive tracking control policy based on reference governor techniques, which bridges the gap between geometric planner and low-level stabilizing controller. In contrast to existing work, by employing parameterized barrier function, our algorithms enable safe output tracking for time-varying constraints that obtained from online sensor measurements. In Chapter 3, we extend our safe tracking framework from feedback-linearizable system to more complex nonlinear system modeled using Hamiltonian Dynamics. To address model uncertainty, we use an ordinary differential equation network to learn the dynamics of a Hamiltonian system from trajectory data. By exploiting the structure of passivity-based controller and associated energy-based Lyapunov function, we extend our reference governor tracking control design Hamiltonian model and analyze the robustness against uncertainty. Our Hamiltonian dynamics learning and tracking control techniques are demonstrated on simulated hexarotor and quadrotor robots navigating in cluttered 3D environments. In Chapter 4, we focus on improving adaptiveness of algorithm and extend our method to dynamical environments. We developed novel state-dependent directional metric for accurate trajectory estimation and reachable set approximation. Together with distance-based costmap as input for path planner, adaptive safety metrics allow the robot to adapt its behavior to sensed local environments, resulting control policy can navigate the robot fast and safely in complex unknown environments. Furthermore, combining CBF techniques, we propose a novel active governor control policy, allowing intermediate goals temporarily deviated from reference signal, so robot is alert to surrounding dynamical obstacles and take necessary action to avoid them. We validate our algorithm extensively at different robot platforms both in simulated and physical worlds. The code is open source at here. In Chapter 5, we summarized our work and pointed out future research directions.

Chapter 2

Safe Output Tracking via Reference Governor

In this chapter, we introduce a reference governor design approach for the safe output tracking problem. We formulate the safe output tracking control problem for a linear time-invariant (LTI) system and cover background on output tracking. Next, we present our safe output regulation method using parametric barrier functions (PBFs). Then, we present a safe output tracking design using reference governor techniques and summarize the main result in a theorem. Finally, we evaluate our algorithm using a simulated omni-directional robot which is a feedback linearizable system.

2.1 Problem Formulation

Consider a linear time-invariant dynamical system:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{y} &= \mathbf{C}\mathbf{x}, \end{aligned} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the control input, $\mathbf{y} \in \mathbb{R}^m$ is the output.

The goal of this chapter is to design a controller such that the output $\mathbf{y}$ of (2.1) tracks a reference path without violating safety constraints or adhering to pre-defined time scaling. To accommodate output constraints, we define an obstacle-free open set $\mathscr{F} \subset \mathbb{R}^m$ and a closed obstacle set $\mathscr{O} := \mathbb{R}^m \setminus \mathscr{F}$. Motivated by applications in autonomous system navigation, we assume that the obstacle set $\mathscr{O}$ is not known a priori. Instead, the system can sense the distance from its output $\mathbf{y}$ to $\mathscr{O}$ only locally with a limited sensing range $\beta > 0$:

$$d_s(\mathbf{y}, \mathscr{O}) = \begin{cases} -\min\{d(\mathbf{y}, \partial\mathscr{O}), \beta\} & \text{if } \mathbf{y} \in \operatorname{int}(\mathscr{O}) \\ \min\{d(\mathbf{y}, \partial\mathscr{O}), \beta\} & \text{if } \mathbf{y} \notin \operatorname{int}(\mathscr{O}) \end{cases} \tag{2.2}$$

where $d(\mathbf{y}, \partial\mathscr{O}) := \min_{\mathbf{a}\in\partial\mathscr{O}} \|\mathbf{y} - \mathbf{a}\|$. We denote the interior of a set $\mathscr{A}$ as $\operatorname{int}(\mathscr{A})$ and its closure as $\operatorname{cl}(\mathscr{A})$.

Problem 1. Let $\mathscr{P} : [0,1] \mapsto \mathscr{F}$ be a continuous function specifying an output reference path for the system in (2.1). Assume that $\mathscr{P}(0) = \mathbf{y}(t_0) \in \mathscr{F}$. Using local distance observations $d_s(\mathbf{y}, \mathscr{O})$ of the obstacle set $\mathscr{O}$, design a control policy for (2.1) so that the output $\mathbf{y}(t)$ of the closed-loop system converges asymptotically to $\mathscr{P}(1)$, while remaining safe, i.e., $\mathbf{y}(t) \in \operatorname{cl}(\mathscr{F})$ for all $t \geq t_0$.

In robot navigation applications, a reference path $\mathscr{P}$ can be generated and updated online by continuously mapping the occupied space using distance observations [80, 48] and replanning a path in free space using a motion planning algorithm [62].

2.2 Output Regulation without Constraints

We first discuss stabilizing the output of (2.1) to an arbitrary fixed point $\mathbf{g} \in \mathbb{R}^m$ without constraints.

Problem 2. Design a sufficiently smooth function $\mathbf{u} = \mathbf{k}(\mathbf{x}, \mathbf{g})$ such that for any constant $\mathbf{g} \in \mathbb{R}^m$, the closed-loop system:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\,\mathbf{k}(\mathbf{x}, \mathbf{g}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \\ \mathbf{y} &= \mathbf{C}\mathbf{x}, \end{aligned} \tag{2.3}$$

admits a $\mathbf{g}$-parameterized equilibrium $\mathbf{x}_\mathbf{g} = \mathbf{X}\mathbf{g}$ for some constant matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ and the output $\mathbf{y}$ converges to $\mathbf{g}$ exponentially.

Theorem 1 ([32])**.** *Let* $\mathbf{K} \in \mathbb{R}^{m\times n}$ *be such that* $(\mathbf{A}+\mathbf{BK})$ *is Hurwitz. If there exist matrices* $\mathbf{X} \in \mathbb{R}^{n\times m}$ *and* $\mathbf{U} \in \mathbb{R}^{m\times n}$ *that satisfy the regulator equations:*

$$\begin{aligned} \mathbf{0} &= \mathbf{AX}+\mathbf{BU}, \\ \mathbf{0} &= \mathbf{CX}-\mathbf{I}, \end{aligned} \tag{2.4}$$

then there exists a static state-feedback controller that solves Problem 2:

$$\mathbf{k}(\mathbf{x},\mathbf{g}) = \mathbf{Kx}+(\mathbf{U}-\mathbf{KX})\mathbf{g}. \tag{2.5}$$

Proof. Let $\bar{\mathbf{x}} := \mathbf{x}-\mathbf{Xg}$. Since $\mathbf{X}$ and $\mathbf{U}$ satisfy (2.4), the closed-loop system in (2.3) with $\mathbf{k}(\mathbf{x},\mathbf{g})$ in (2.5) becomes:

$$\begin{aligned} \dot{\bar{\mathbf{x}}} &= \mathbf{Ax}+\mathbf{B}\left[\mathbf{Kx}+(\mathbf{U}-\mathbf{KX})\mathbf{g}\right] \\ &= (\mathbf{A}+\mathbf{BK})(\bar{\mathbf{x}}+\mathbf{Xg})-(\mathbf{A}+\mathbf{BK})\,\mathbf{Xg} \\ &= (\mathbf{A}+\mathbf{BK})\bar{\mathbf{x}}, \\ \mathbf{y} &= \mathbf{C}\bar{\mathbf{x}}+\mathbf{g}. \end{aligned} \tag{2.6}$$

Since $(\mathbf{A}+\mathbf{BK})$ is Hurwitz by assumption, the closed-loop system in (2.6) is exponentially stable with equilibrium $\mathbf{x_g} = \mathbf{Xg}$ and steady-state output $\mathbf{y} = \mathbf{g}$. □

Assumption 1. The pair $(\mathbf{A},\mathbf{B})$ is stabilizable and

$$\operatorname{rank}\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} = n+m \tag{2.7}$$

Under Assumption 1, the regulator equations (2.4) are guaranteed to have a solution [49, Theorem 1.9] and the static feedback controller in (2.5) solves Problem 2.

2.3 Safe Output Regulation via PBF

Section 2.2 showed how to regulate the output of (2.1) under Assumption 1 to a desired reference point $\mathbf{g}$. Next, we consider Problem 2 in the presence of output constraints. We construct a $\mathbf{g}$-*parameterized barrier function* (PBF) that quantifies the trade-off between safety (distance from $\mathbf{g}$ to the obstacle set $\mathscr{O}$ in (2.2)) and the system energy. For controllable $(\mathbf{A},\mathbf{B})$, following Thm. 1, the energy of (2.3) can be measured by a $\mathbf{g}$-parameterized quadratic Lyapunov function [54]:

$$V(\mathbf{x};\mathbf{g}) = (\mathbf{x}-\mathbf{X}\mathbf{g})^\top \mathbf{P}\,(\mathbf{x}-\mathbf{X}\mathbf{g})\,, \tag{2.8}$$

where $\mathbf{P}$ is the unique solution of the Lyapunov equation $(\mathbf{A}+\mathbf{B}\mathbf{K})^\top\mathbf{P}+\mathbf{P}(\mathbf{A}+\mathbf{B}\mathbf{K}) = -\mathbf{Q}$ for any positive-definite symmetric matrix $\mathbf{Q} \in \mathbb{S}^n_{>0}$. The Lyapunov function can be expressed as $V(\mathbf{x};\mathbf{g}) = \|\mathbf{x}-\mathbf{X}\mathbf{g}\|^2_{\mathbf{P}}$ using a quadratic norm $\|\mathbf{x}\|_{\mathbf{P}} := \sqrt{\mathbf{x}^\top\mathbf{P}\mathbf{x}}$ on $\mathbb{R}^n$ defined by $\mathbf{P} \in \mathbb{S}^n_{>0}$. The system output $\mathbf{y} = \mathbf{C}\mathbf{x}$ is Lipschitz continuous with respect to $\|\mathbf{x}\|_{\mathbf{P}}$, allowing us to relate the distance to the obstacle set $\mathscr{O}$ with the distance to the desired equilibrium $\mathbf{x}_{\mathbf{g}} = \mathbf{X}\mathbf{g}$.

Lemma 1. *For any* $\mathbf{P} \in \mathbb{S}^n_{>0}$ *with Cholesky factorization* $\mathbf{P} = \mathbf{L}\mathbf{L}^\top$*, there exists a global Lipschitz constant* $L = \lambda^{1/2}_{\max}(\mathbf{L}^{-1}\mathbf{C}^\top\mathbf{C}\mathbf{L}^{-\top})$ *such that:*

$$\|\mathbf{C}\mathbf{x}_1 - \mathbf{C}\mathbf{x}_2\| \leq L\|\mathbf{x}_1-\mathbf{x}_2\|_{\mathbf{P}},\ \forall \mathbf{x}_1,\mathbf{x}_2 \in \mathbb{R}^n. \tag{2.9}$$

Proof. The result follows from the generalized Rayleigh quotient. □

Definition 1. Let $V(\mathbf{x};\mathbf{g})$ be a $\mathbf{g}$-parameterized Lyapunov function for output regulation to $\mathbf{g} \in \mathbb{R}^m$. Let $d_s(\mathbf{g},\mathscr{O})$ be the truncated signed distance function to the obstacle set $\mathscr{O}$ in (2.2). A $\mathbf{g}$-*parameterized barrier function* (PBF) is:

$$b(\mathbf{x};\mathbf{g}) := d^2_s(\mathbf{g},\mathscr{O}) - L^2V(\mathbf{x};\mathbf{g}) \tag{2.10}$$

where L is the Lipschitz constant in (2.9).

Proposition 1. *Let* $\mathbf{g} \in \mathscr{F}$*. Consider the closed-loop system in* (2.3) *with controller in* (2.5)*. The set* $\mathscr{S}(\mathbf{g}) := \{\mathbf{x} \mid b(\mathbf{x};\mathbf{g}) \geq 0\}$ *is positively invariant and the output* $\mathbf{y}(t)$ *converges to* $\mathbf{g}$ *asymptotically without violating the output constraints, i.e.,* $\mathbf{y}(t) \in cl(\mathscr{F})$ *for all* $t \geq t_0$*.*

Proof. For fixed $\mathbf{g}$, the time derivative of $b(\mathbf{x};\mathbf{g})$ is strictly positive, $\frac{\partial b}{\partial \mathbf{x}}\dot{\mathbf{x}} = -L^2 \frac{\partial V}{\partial \mathbf{x}}\dot{\mathbf{x}} > 0$, because $V(\mathbf{x};\mathbf{g})$ is a Lyapunov function. Hence, $\mathscr{S}(\mathbf{g})$ is forward invariant. Thm. 1 guarantees that $\mathbf{y}(t) \to \mathbf{g}$ and $\mathbf{x}(t) \to \mathbf{x}_{\mathbf{g}} = \mathbf{X}\mathbf{g}$. Finally, we show that, if $\mathbf{x}_0 \in \mathscr{S}(\mathbf{g})$, then $\mathbf{y}(t) \in \mathrm{cl}(\mathscr{F})$ for all $t \geq t_0$. Since the time derivative of $b(\mathbf{x};\mathbf{g})$ is positive, $b(\mathbf{x}(t);\mathbf{g}) \geq b(\mathbf{x}_0;\mathbf{g}) \geq 0$ for all $t \geq t_0$. From (2.4),(2.9) and (2.10):

$$\begin{aligned} b(\mathbf{x}(t);\mathbf{g}) &\leq d^2(\mathbf{g},\partial\mathscr{O}) - L^2\|\mathbf{x}(t) - \mathbf{x}_{\mathbf{g}}\|_{\mathbf{P}}^2 \\ &\leq d^2(\mathbf{g},\partial\mathscr{O}) - \|\mathbf{C}\mathbf{x}(t) - \mathbf{g}\|^2. \end{aligned} \tag{2.11}$$

Hence, for all $t \geq t_0$, $\|\mathbf{y}(t) - \mathbf{g}\| \leq d(\mathbf{g},\partial\mathscr{O})$, and since $\mathbf{g} \in \mathscr{F}$, $\mathbf{y}(t) \in \mathrm{cl}(\mathscr{F})$ for all $t \geq t_0$. □

Remark 1. Proposition 1 and the remaining results in the paper hold when only a subspace of $\mathbb{R}^m$ is constrained. Assume, without loss of generality, that only the first $m_1 \leq m$ dimensions are constrained so that $\mathscr{O} = \mathscr{O}_1 \times \emptyset^{m-m_1}$, $\mathscr{F}_1 = \mathbb{R}^{m_1} \setminus \mathscr{O}_1$, and $\mathscr{F} = \mathscr{F}_1 \times \mathbb{R}^{m-m_1}$. The results can be extended by defining $d(\mathbf{y},\partial\mathscr{O}) := \min_{\mathbf{b}\in\partial\mathscr{O}_1}\|\mathbf{b} - \mathbf{P}_1\mathbf{y}\|$, where $\mathbf{P}_1 = [\mathbf{I},\mathbf{0}] \in \mathbb{R}^{m_1\times m}$ is a projection matrix. For example, noting that $\|\mathbf{P}_1\mathbf{y} - \mathbf{P}_1\mathbf{g}\| \leq \|\mathbf{y} - \mathbf{g}\|$, $\forall \mathbf{y},\mathbf{g} \in \mathbb{R}^m$, (2.11) implies that

$$\min_{\mathbf{b}\in\partial\mathscr{O}_1}\|\mathbf{b} - \mathbf{P}_1\mathbf{g}\| \geq \|\mathbf{P}_1\mathbf{y}(t) - \mathbf{P}_1\mathbf{g}\|,$$

i.e., $\mathbf{P}_1\mathbf{y}(t) \in \mathrm{cl}(\mathscr{F}_1)$, which means that $\mathbf{y}(t) \in \mathrm{cl}(\mathscr{F})$.

2.4 Safe Output Tracking using a Reference Governor

Section 2.3 discussed output regulation to a static reference point $\mathbf{g}$ using a PBF to quantify safety. In this section, we develop an approach to adaptively change the regulation point $\mathbf{g}(t)$ so that the output of the closed-loop system in (2.3) tracks the desired path $\mathscr{P}$ safely.

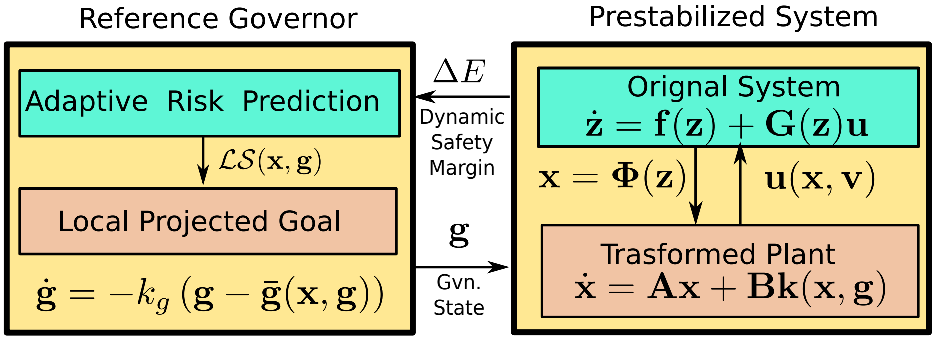

Reference Governor
Adaptive Risk Prediction
$\mathcal{LS}(\mathbf{x},\mathbf{g})$
Local Projected Goal
$\dot{\mathbf{g}} = -k_g\,(\mathbf{g} - \bar{\mathbf{g}}(\mathbf{x},\mathbf{g}))$
ΔE
Dynamic
Safety
Margin
$\mathbf{g}$
Gvn.
State
Prestabilized System
Orignal System
$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}) + \mathbf{G}(\mathbf{z})\mathbf{u}$
$\mathbf{x} = \mathbf{\Phi}(\mathbf{z})$
$\mathbf{u}(\mathbf{x},\mathbf{v})$
Trasformed Plant
$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{k}(\mathbf{x},\mathbf{g})$

Figure 2.1. The structure of the reference-governor controller. A virtual governor system with state $\mathbf{g}$ adaptively tracks the desired reference path $\mathscr{P}$ while ensuring that the output of a prestabilized system can track $\mathbf{g}$.

Our control design consists of two parts: a virtual *reference governor* system whose state $\mathbf{g}(t)$ adaptively moves along the path $\mathscr{P}$ and the closed-loop system in (2.3), tracking the time-varying reference point $\mathbf{X}\mathbf{g}(t)$. The structure of the reference-governor controller is visualized in Fig. 2.1.

Definition 2. A *reference governor* is a linear system:

$$\dot{\mathbf{g}} = -k_g\,(\mathbf{g} - \bar{\mathbf{g}}) \tag{2.12}$$

with gain $k_g > 0$, state $\mathbf{g} \in \mathbb{R}^m$, and input $\bar{\mathbf{g}} \in \mathbb{R}^m$.

We show that the slackness in the PBF safety metric in (2.10) can be used to move the governor state $\mathbf{g}$ along the reference path $\mathscr{P}$ without endangering safety or stability of the closed-loop system in (2.3).

Definition 3 ([78]). A continuous function $\Delta E(\mathbf{x},\mathbf{g}) : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ is a *dynamic safety margin* (DSM) for the closed-loop system in (2.3) if:

1. $\Delta E(\mathbf{x},\mathbf{g}) \geq 0 \implies d_s(\mathbf{C}\mathbf{x},\mathscr{O}) \geq 0$,
2. $\Delta E(\mathbf{x}_0,\mathbf{g}) = 0 \implies \Delta E\,(\mathbf{x}(t),\mathbf{g}) \geq 0, \forall t \geq t_0$,

3. for all $\delta > 0$, there exists $\varepsilon > 0$ such that

   $d_s(\mathbf{g}, \mathscr{O}) \geq \delta \implies \Delta E(\mathbf{x}_{\mathbf{g}}, \mathbf{g}) \geq \varepsilon.$

A DSM is a measure of system safety, i.e., larger ΔE means that the system is safer with respect to the output constraints. The first condition requires that non-negative ΔE implies that the system is safe at the current moment, while the second condition requires certification of safety forward in time for fixed $\mathbf{g}$. The last condition requires that the DSM captures the slackness in the safety constraints.

Lemma 2. *For $\mathbf{g} \in \mathscr{F}$, the $\mathbf{g}$-parameterized barrier function in* (2.10) *is a dynamic safety margin for the closed-loop system in* (2.3)*:* $\Delta E(\mathbf{x}, \mathbf{g}) = b(\mathbf{x}; \mathbf{g})$.

Proof. From (2.10), (2.8), (2.9), and (2.2):

$$\begin{aligned} \Delta E(\mathbf{x}, \mathbf{g}) \geq 0 &\implies d_s^2(\mathbf{g}, \mathscr{O}) \geq L^2 \|\mathbf{x} - \mathbf{X}\mathbf{g}\|_{\mathbf{P}}^2 \\ &\implies d_s(\mathbf{g}, \mathscr{O}) \geq \|\mathbf{C}\mathbf{x} - \mathbf{g}\| \\ &\implies d(\mathbf{g}, \partial \mathscr{O}) \geq \|\mathbf{C}\mathbf{x} - \mathbf{g}\| \end{aligned} \tag{2.13}$$

Since $\mathbf{g} \in \mathscr{F}$ the last inequality implies that $\mathbf{y} = \mathbf{C}\mathbf{x} \in \mathrm{cl}(\mathscr{F})$ and, hence, $d_s(\mathbf{C}\mathbf{x}, \mathscr{O}) \geq 0$. The second requirement of Def. 3 follows from Proposition 1. The last one holds with $\varepsilon = \min\left\{\delta^2, \beta^2\right\}$: $b(\mathbf{x}_{\mathbf{g}}; \mathbf{g}) = d_s^2(\mathbf{g}, \mathscr{O}) - 0 \geq \min\left\{\delta^2, \beta^2\right\} = \varepsilon$. □

In the rest of the paper, we study the case of a moving governor $\mathbf{g}(t)$. We denote the Lyapunov function and PBF by $V(\mathbf{x}, \mathbf{g})$ and $b(\mathbf{x}, \mathbf{g})$, instead of $V(\mathbf{x}; \mathbf{g})$ and $b(\mathbf{x}; \mathbf{g})$, to emphasize the fact that $\mathbf{g}(t)$ is time-varying. To guarantee safe output tracking, the input $\bar{\mathbf{g}}$ of the governor system in (2.12) must be chosen by jointly considering the geometry of the local safe space and the activeness of the prestabilized system. This trade-off is captured by the PBF $b(\mathbf{x}, \mathbf{g})$. We define a set of feasible governor inputs that will not violate the safety or stability for the closed-loop system in (2.3).

Definition 4. A *local safe zone* is a time-varying set, determined by the joint system-governor state $(\mathbf{x},\mathbf{g})$, a dynamic safety margin $\Delta E(\mathbf{x},\mathbf{g})$, and a constant $l > 1$:

$$\mathscr{LS}(\mathbf{x},\mathbf{g}) := \left\{ \mathbf{q} \in \mathbb{R}^m \mid \|\mathbf{q}-\mathbf{g}\|^2 \leq l^{-1}\Delta E(\mathbf{x},\mathbf{g}) \right\}. \tag{2.14}$$

Remark 2. The constant $l > 1$ in Def. 4 is needed to ensure that $\mathscr{LS}(\mathbf{x},\mathbf{g}) \subset \mathscr{F}$. When $\Delta E(\mathbf{x},\mathbf{g}) = 0$, $\mathscr{LS}(\mathbf{x},\mathbf{g}) = \{\mathbf{g}\} \subset \mathscr{F}$. If $\Delta E(\mathbf{x},\mathbf{g}) > 0$, then for any $\mathbf{q} \in \mathscr{LS}(\mathbf{x},\mathbf{g})$, $\|\mathbf{q}-\mathbf{g}\|^2 < \Delta E(\mathbf{x},\mathbf{g}) \leq d_s^2(\mathbf{g},\mathscr{O})$, which implies that $\|\mathbf{q}-\mathbf{g}\| < d_s(\mathbf{g},\mathscr{O})$, i.e., $\mathbf{q} \in \mathscr{F}$.

We show, in the proof of Thm. 2 below, that choosing $\bar{\mathbf{g}} \in \mathscr{LS}(\mathbf{x},\mathbf{g}) \subset \mathscr{F}$ ensures that system safety is guaranteed and the governor trajectory $\mathbf{g}(t)$ lies in $\mathscr{F}$ and always eventually makes $\Delta E(\mathbf{x},\mathbf{g})$ strictly positive until reaching $\mathscr{P}(1)$. To make the governor progress along the reference path $\mathscr{P}$ and lead the closed-loop system, we choose the governor input $\bar{\mathbf{g}}$ as the furthest point along $\mathscr{P}$ that is contained in $\mathscr{LS}(\mathbf{x},\mathbf{g})$.

Definition 5. A *local projected goal* at system-governor state $(\mathbf{x},\mathbf{g})$ is a point $\bar{\mathbf{g}} \in \mathscr{LS}(\mathbf{x},\mathbf{g})$ that is furthest along the reference path $\mathscr{P}$:

$$\bar{\mathbf{g}} = \mathscr{P}(\bar{\sigma}), \;\; \bar{\sigma} = \underset{\sigma\in[0,1]}{\operatorname{argmax}} \{\sigma \mid \mathscr{P}(\sigma) \in \mathscr{LS}(\mathbf{x},\mathbf{g})\}. \tag{2.15}$$

We summarize the closed-loop dynamics for the joint $(\mathbf{x},\mathbf{g})$ system controlled by the output regulator in (2.5) and the reference-governor control law in (2.15):

$$\dot{\mathbf{x}} = (\mathbf{A}+\mathbf{BK})(\mathbf{x}-\mathbf{Xg}), \tag{2.16a}$$

$$\dot{\mathbf{g}} = -k_g\left(\mathbf{g}-\bar{\mathbf{g}}\right), \tag{2.16b}$$

$$\mathbf{y} = \mathbf{Cx} \tag{2.16c}$$

Theorem 2. *Given a reference path $\mathscr{P}$, consider the closed-loop system in* (2.16)*. Suppose that* *the initial state* $(\mathbf{x}_0, \mathbf{g}_0)$ *satisfies:*

$$\Delta E(\mathbf{x}_0, \mathbf{g}_0) > 0, \quad \mathbf{g}_0 = \mathscr{P}(0) = \mathbf{y}(t_0) \in \mathscr{F}, \tag{2.17}$$

where $\Delta E(\mathbf{x}, \mathbf{g})$ *is the dynamic safety margin in* (2.10)*. Then, the joint state* $(\mathbf{x}, \mathbf{g})$ *converges to* $(\mathbf{X}\mathscr{P}(1), \mathscr{P}(1))$ *without violating the output constraints, i.e.,* $\mathbf{y}(t) \in cl(\mathscr{F})$*,* $\forall t \geq t_0$.

Proof. The proof consists of three parts. First, we prove that the dynamics in (2.16) are updated continuously. Second, we show that the output constraints are not violated under (2.16). Last, we prove that the joint system (2.16) has a unique stable equilibrium point at $(\mathbf{X}\mathscr{P}(1), \mathscr{P}(1))$.

First, we show that the DSM $\Delta E(t) := b(\mathbf{x}(t), \mathbf{g}(t))$ is continuous. In Lemma 6, we prove that $\|\dot{\mathbf{g}}(t)\|$ is uniformly bounded by $k_g \beta \sqrt{l^{-1}}$ and therefore $\mathbf{g}(t)$ is continuous. Then, since the truncated signed distance function $d_s(\mathbf{g}(t), \mathscr{O})$ is continuous (Lemma 7) and $V(\mathbf{x}, \mathbf{g})$ is continuous in $(\mathbf{x}, \mathbf{g})$, we show in Lemma 8 that $\Delta E(t)$ is also a continuous function in t. The state $\mathbf{x}$ is regulated by a static feedback controller $\mathbf{k}(\mathbf{x}, \mathbf{g})$ and is also continuous. Hence, the system dynamics in (2.16) are updated continuously.

Second, we prove that safety is ensured, i.e., for all $t \geq t_0$, $\mathbf{y}(t) \in \mathrm{cl}(\mathscr{F})$, when $\mathbf{g}(t)$ is changing according to (2.16b). Lemma 10 shows that the set

$$\mathscr{S} := \{(\mathbf{x}, \mathbf{g}) \in \mathbb{R}^n \times \mathbb{R}^m \mid \Delta E(\mathbf{x}, \mathbf{g}) \geq 0\}$$

is positively invariant for the closed-loop system in (2.16). Hence, $\Delta E(t) \geq 0$ for all $t \geq t_0$ and, by the first property of a dynamic safety margin in Def. 3, $d_s(\mathbf{y}(t), \mathscr{O}) \geq 0$ for all $t \geq t_0$. In detail, initially $\mathbf{g}_0 = \mathbf{y}_0 = \mathscr{P}(0) \in \mathscr{L}\mathscr{S}(\mathbf{x}_0, \mathbf{g}_0)$ and $\Delta E(t_0) > 0$. The local projected goal $\bar{\mathbf{g}}$ in (2.15) is well defined and moves along the reference path $\mathscr{P}$, i.e., $\bar{\sigma}$ in (2.15) increases. As $\mathbf{g}$ tracks $\bar{\mathbf{g}}$ using (2.16b), the system state $\mathbf{x}$ tracks $\mathbf{X}\mathbf{g}$ using the controller in (2.5). During this process, the DSM $\Delta E(t) = b(\mathbf{x}(t), \mathbf{g}(t))$, as the difference of $d_s^2(\mathbf{g}, \mathscr{O})$ and the scaled Lyapunov function $V(t)$, is fluctuating and regulating the rate of change of $\mathbf{g}$ (see Fig. 2.3). Lemma 9 shows that for

$\bar{\mathbf{x}} = \mathbf{x} - \mathbf{X}\mathbf{g}$:

$$\dot{V}(t) \leq -\bar{\mathbf{x}}(t)^\top \mathbf{Q}\bar{\mathbf{x}}(t) + 2\|\mathbf{X}^\top \mathbf{P}\bar{\mathbf{x}}(t)\|\|\dot{\mathbf{g}}(t)\|,$$
$$D_+\Delta E(t) \geq -2k_g M(t)\sqrt{\Delta E(t)/l} + L^2\bar{\mathbf{x}}(t)^\top \mathbf{Q}\bar{\mathbf{x}}(t),$$

where $D_+\Delta E(t)$ is the lower-right Dini derivative and $M(t)$ is bounded pointwise in time. By continuity, $\Delta E(t)$ cannot become negative instantaneously without crossing 0 at some time $t = T_0$. Lemma 9 shows that $\Delta E(T_0 + h)$ will bounce back from 0 to a strictly positive number after any such time T_0.

Finally, we show that the joint state $(\mathbf{x}, \mathbf{g})$ converges to $(\mathbf{X}\mathscr{P}(1), \mathscr{P}(1))$ under the dynamics in (2.16). Note that $\mathbf{g}(t) \in \mathscr{L}\mathscr{S}(\mathbf{x}(t), \mathbf{g}(t))$ and, from Remark 2, $\mathbf{g}(t) \in \mathscr{F}$ for all $t \geq t_0$. If $\mathbf{g} = \mathscr{P}(1)$, then $\mathscr{P}(1) \in \mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g})$, $\bar{\sigma} = 1$, and $\bar{\mathbf{g}} = \mathscr{P}(1)$ in (2.15). Then, $\dot{\mathbf{g}} \equiv 0$ and the output regulator in (2.5) drives $\mathbf{x}$ to $\mathbf{X}\mathscr{P}(1)$. Hence, $(\mathbf{X}\mathscr{P}(1), \mathscr{P}(1))$ is an equilibrium point for (2.16). From Lemma 9, whenever $\Delta E(t) = 0$ at an arbitrary time $t = T_i$, it becomes strictly positive after some time h_i. Then, at $t_i = T_i + h_i$, the joint state $(\mathbf{x}(t_i), \mathbf{g}(t_i))$ satisfies:

$$\Delta E(t_i) > 0, \quad \mathbf{g}(t_i) \in \mathscr{F}, \tag{2.18}$$

and we are back to the case from the beginning. The local projected goal $\bar{\mathbf{g}}$ gets closer to $\mathscr{P}(1)$ and guides the joint system. It is not possible to have another equilibrium point because $\mathbf{g}(t) \in \mathscr{F}$ for all $t \geq t_0$ and, by the third DSM property in Def. 3, $\Delta E(\mathbf{X}\mathbf{g}, \mathbf{g}) \geq \varepsilon$. From Def. 5, $\bar{\mathbf{g}}(t)$ can only stop moving at $\mathscr{P}(1)$ when $\Delta E(\mathbf{X}\mathbf{g}, \mathbf{g}) > 0$. Hence, the joint system in (2.16) has a unique stable equilibrium point at $(\mathbf{X}\mathscr{P}(1), \mathscr{P}(1))$. □

In summary, Thm. 2 shows that the control law:

$$\pi(\mathbf{x}, \mathbf{g}) = \mathbf{k}(\mathbf{x}, \mathbf{g}), \tag{2.19}$$
$$\dot{\mathbf{g}} = -k_g(\mathbf{g} - \bar{\mathbf{g}}(\mathbf{x}, \mathbf{g})), \;\; \mathbf{g}_0 = \mathbf{y}(t_0) = \mathscr{P}(0),$$

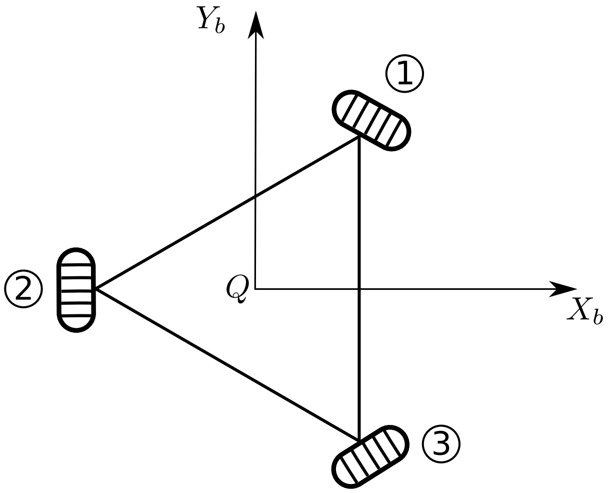


Figure 2.2. A mobile robot with omnidirectional wheels.

combining the controller $\mathbf{k}(\mathbf{x},\mathbf{g})$ in (2.5) and the reference governor in (2.12), (2.15) solves Problem 1 as long as the dynamic safety margin is strictly positive initially, $\Delta E(\mathbf{x}_0,\mathbf{g}_0) = b(\mathbf{x}_0,\mathbf{g}_0) > 0$.

2.5 Evaluation

This section evaluates our safe output-tracking controller on a simulated mobile robot, measuring distances to obstacles in an unknown environment.

System Model. Consider a mobile robot equipped three identical Swedish omnidirectional wheels [25], shown in Fig. 2.2. Let m be the mass and I be the inertia around the Z_b axis (perpendicular to the X_b, Y_b plane in the body frame). The robot's motion is described by the position and orientation, (x,y,θ), of the body frame and the positions of ϕ_1, ϕ_2, ϕ_3 of the three wheels. The robot's dynamics can be obtained using Euler-Lagrange equations subject to pure-rolling non-holonomic constraints for the three wheels [25, 20]. When the non-holonomic constraints are considered, the wheel positions ϕ_i may be eliminated, leading to the following dynamics model:

$$\mathbf{M}_1\mathbf{R}^\top(\theta)\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{\theta}\end{bmatrix} = -\mathbf{J}_1^\top\mathbf{J}_3^{-1}\mu, \tag{2.20}$$

with:

$$\mathbf{R}(\theta) = \begin{bmatrix}\cos(\theta) & -\sin(\theta) & 0\\ \sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix}, \quad \mathbf{J}_1 = \begin{bmatrix}-\frac{\sqrt{3}}{2} & \frac{1}{2} & d\\ 0 & -1 & d\\ \frac{\sqrt{3}}{2} & \frac{1}{2} & d\end{bmatrix},$$

$$\mathbf{M}_1 = \operatorname{diag}(m,m,I), \qquad \mathbf{J}_3 = \operatorname{diag}(r,r,r),$$

where d is the distance from the robot center to the wheels and r is the wheel radius. The input $\mu \in \mathbb{R}^3$ contains the generalized forces and torque.

We consider obstacles $\mathscr{O}_1 \subset \mathbb{R}^2$ with no constraints on orientation, i.e., $\mathscr{O} = \mathscr{O}_1 \times \{\emptyset\}$, as shown in Fig. 2.3 and Fig. 2.4. As mentioned in Remark 1, the distance $d(\mathbf{y}, \partial\mathscr{O})$ is defined as:

$$d(\mathbf{y}, \partial\mathscr{O}) = \min_{\mathbf{b}\in\partial\mathscr{O}_1} \|\mathbf{P}_1\mathbf{y} - \mathbf{b}\|, \tag{2.21}$$

where $\mathbf{P}_1 = [\mathbf{I}, \mathbf{0}]$. An output reference path $\mathscr{P} : [0,1] \mapsto \mathscr{F} \subset \mathbb{R}^3$ is provided as shown in Fig. 2.3 and Fig. 2.4 with desired orientation fixed at 0.

Environment Sensing and Path Generation. In an unknown environment, the obstacle set $\mathscr{O}_1$ is not known. In our simulations, a simulated Lidar sensor provides a set of points $\mathscr{P}(t) := \{\mathbf{p}_i(t)\}_i$ on the surface of the obstacle set $\mathscr{O}_1$, depending on the (position) output $\mathbf{y}(t)$, with a maximum sensing range of $\beta = 25$. The distance from the governor to the obstacle set is approximated as,

$$d(\mathbf{g}(t), \partial\mathscr{O}) \approx \min_{\mathbf{p}\in\mathscr{P}(t)} \|\mathbf{P}_1\mathbf{g}(t) - \mathbf{p}\|. \tag{2.22}$$

Note that, the same Lidar hit points $\mathscr{P}(t)$ from output depth measurement are used in above expression. To obtain a feasible reference path $\mathscr{P}$, an occupancy grid [48] is created and updated

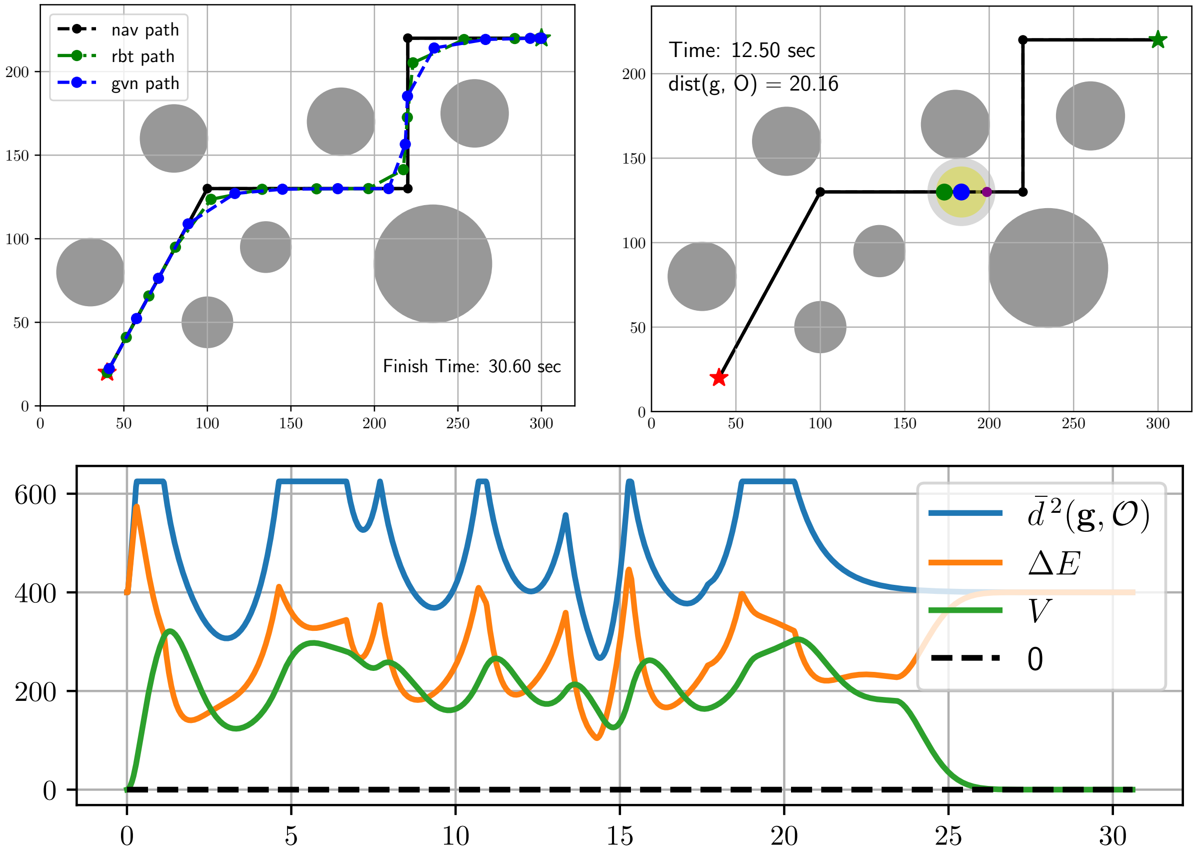


Figure 2.3. Simulation result of output-tracking control of an omnidirectional mobile robot navigating in an unknown environment. Top left shows the projection of the reference path $\mathscr{P}$ and the paths followed by the system and the governor. Top right shows a snapshot at time $t = 15$. The bottom plot shows the summary of safety metrics, indicating that safe navigation is achieved.

using the Lidar measurements (see Fig. 2.4). The grid map is discretized at resolution 0.5 m with a 0.5 m inflation around the obstacles. The unknown obstacle set $\mathscr{O}_1$ is over-approximated by the union $\hat{\mathscr{O}}_1(t)$ of all occupied cells up to time t. Using the latest map, the reference path $\mathscr{P}$ is recomputed periodically using the A^* motion planner [62]. Since $\hat{\mathscr{O}}_1(t)$ is an over-approximation of $\mathscr{O}_1$, the re-planned reference paths lie within the free space $\mathscr{F}$. A snapshot of the occupancy grid map and one of the (re)planned paths (blue curves at top) are shown in Fig. 2.4.

Control Design. Observe that in equation (2.20), $\mathbf{M}_1$, $\mathbf{R}(\theta)$, $\mathbf{J}_1$, $\mathbf{J}_3^{-1}$ are all invertible matrices. Hence, this model is feedback linearizable. Let $\mathbf{x} := [x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}]^\top \in \mathbb{R}^6$ and $\mathbf{y} :=$

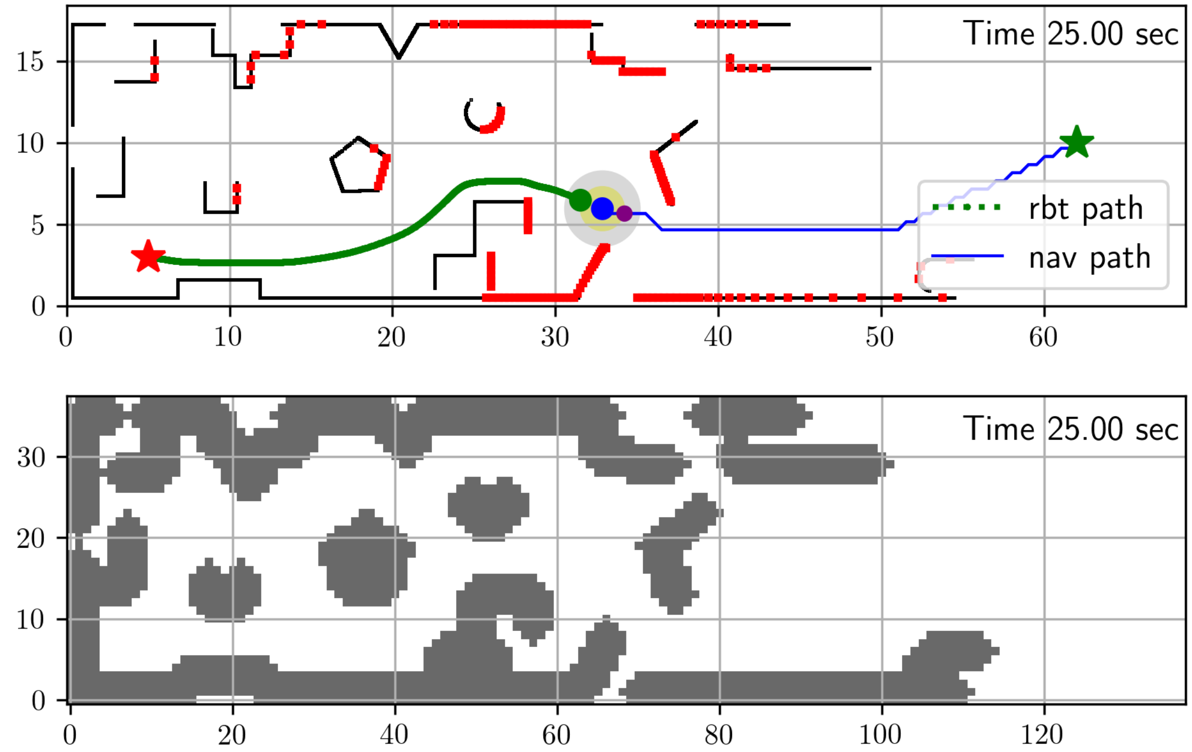


Figure 2.4. Simulation result of output-tracking control of an omnidirectional robot in an unknown environment with complex obstacles, sensed by a simulated Lidar. The bottom plot is the corresponding occupancy grid map at resolution 0.5 m/cell, where gray cells represent inflated obstacles (0.5 m inflation) and white cells represent free space.

$[x, y, \theta]^\top \in \mathbb{R}^3$ be the new state and output vector. Applying control input:

$$\mu = \mathbf{J}_3 \mathbf{J}_1^{-\top} \mathbf{M}_1 \mathbf{R}^\top(\theta) \mathbf{u}, \tag{2.23}$$

transforms system (2.20) into a linear time-invariant form as in (2.1). Specifically, we have block-diagonal matrices $\mathbf{A} = \mathrm{diag}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$, $\mathbf{B} = \mathrm{diag}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$, $\mathbf{C} = \mathrm{diag}(\mathbf{c}_1^\top, \mathbf{c}_2^\top, \mathbf{c}_3^\top)$ with elements $\mathbf{A}_i \in \mathbb{R}^{2\times 2}$, $\mathbf{b}_i \in \mathbb{R}^2$, $\mathbf{c}_i \in \mathbb{R}^2$:

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{b}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \mathbf{c}_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{2.24}$$

for $i = 1, 2, 3$. It can be verified that Assumption 1 is satisfied, and the regulator equation (2.4) is solved by $\mathbf{X} = \mathbf{C}^\top$ and $\mathbf{U} = \mathbf{0}$.

With $\bar{\mathbf{x}} = (\mathbf{x} - \mathbf{C}^\top \mathbf{g})$, the state-feedback controller $\mathbf{k}(\mathbf{x},\mathbf{g}) = \mathbf{K}\bar{\mathbf{x}}$, with

$$\mathbf{K} = -\mathbf{I} \otimes [2.553, 1.9478],$$

can drive the system output to an arbitrary reference $\mathbf{g}$. To define the PBF $\Delta E(\mathbf{x},\mathbf{g}) = b(\mathbf{x},\mathbf{g})$ in (2.10), we choose a quadratic Lyapunov function $V(\mathbf{x},\mathbf{g}) = \bar{\mathbf{x}}^\top \mathbf{P} \bar{\mathbf{x}}$ with:

$$\mathbf{P} = \mathbf{I} \otimes \begin{bmatrix} 1.7508, & 0.7769 \\ 0.7769, & 0.9739 \end{bmatrix} \in \mathbb{S}^6_{>0}, \tag{2.25}$$

Lipschitz constant $L \approx 0.9403$, computed using Lemma 1, and sensing range $\beta = 25$. The governor control law in (2.12), (2.14), (2.15) is defined with $l = 1.001$ and $k_g = 1.0$.

Simulation results. Throughout this chapter, we denotes governor state $\mathbf{g}$ as blue dot, local projected goal $\bar{\mathbf{g}}$ as purple dot, obstacle distance $\bar{d}(\mathbf{g}, \mathscr{O})$ as gray ball, and local safe zone $\mathscr{L}\mathscr{S}$ as yellow ball. Obstacles are denoted by gray circles or squares. The reference path is depicted as solid black line, while governor path and robot trail are plotted as blue and green line respectively.

Fig. 2.3 shows the behavior of the closed-loop joint system in a constrained output tracking simulation. Our control policy (2.19) successfully enforces the locally sensed obstacle avoidance constraints and drives the system to the goal configuration $\mathscr{P}(1)$. During this process, $\Delta E(t)$ (orange line) is determined by the difference in the size of the local free space and the value of the Lyapunov function value $V(t)$ (green line) as shown in the bottom plot. The governor is controlled adaptively based on $\Delta E(t)$, slowing down when $\Delta E(t)$ is small and speeding up when it is large. The value of PBF never crosses the zero line, indicating that safe navigation is achieved.

In Fig. 2.4, we test our controller in a more challenging unknown environment with non-convex obstacles. With the same controller parameters described above, the system reaches the goal without collisions. The Lidar provides distance measurements (red dots) from the robot position (green dot) to the obstacle set $\mathscr{O}_1$ (black surfaces). The reference path (blue curve) is recomputed online from the governor position (blue dot) to a goal location (green star). The local projected goal $\bar{\mathbf{g}}$ (purple dot) is computed based on the obstacle distance (gray ball) and the local safe zone (yellow ball). Our algorithm can successfully drive the robot towards goal in this unknown environment.

2.6 Summary

This chapter developed an output-tracking controller framework that provides formal safety and stability guarantees for feedback-linearizable control-affine nonlinear systems. We showed that reference-governor techniques can be extended to output tracking with distance measurements to an unsafe set. A key component of our design was a governor-parameterized barrier function, which uses the trade-off between the safe distance and the system Lyapunov function to define a local safe set for the system and governor states. The slackness in the safe set allows the governor to track the reference and the system to track the governor without endangering safety or stability. Our approach allows safe autonomous navigation in a priori unknown unstructured environments. The simple structure of the safety conditions in our design offers a promising research avenue for safe control with learned and approximate system dynamics and constraints. In the next chapter, we will extend the safe tracking framework to learned dynamics for rigid-body systems.

Acknowledgements

Chapter 2, in full, is a reprint of “Governor-parameterized barrier function for safe output tracking with locally sensed constraints” by Zhichao Li and Nikolay Atanasov, which appears in Automatica, Elsevier Press, 2023. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Robust and Safe Autonomous Navigation for Systems with Learned SE(3) Hamiltonian Dynamics

3.1 Overview

In this chapter, we consider both dynamics model learning and safe control synthesis for rigid-body systems, whose states include position, orientation, and generalized velocity. We assume that the system has an unknown dynamics model but, as a physical system, it satisfies Hamilton's equations of motion over the $SE(3)$ manifold of positions and orientations. Given state-control trajectories, from past experiments or collected by a human operator, we seek to learn the system dynamics and design a tracking control law that handles safety constraints, e.g., obtained from distance measurements to obstacles in the environment. In our preliminary work [69], we learn a translation-equivariant Hamiltonian model of the system dynamics using a physics-guided neural ODE network [28]. We use the Hamiltonian model to synthesize an energy-shaping geometric tracking controller. The total energy of the system serves as a Lyapunov function and enables us to enforce safety constraints during trajectory tracking using a reference governor to regulate the difference between the system energy and the distance to safety violation. However, our preliminary work [69] uses the learned Hamiltonian model as the ground-truth dynamics and ignores the model estimation error in the control design. In this paper, we capture the estimation error as a bounded disturbance applied to the learned system and develop a robust safe tracking controller that takes the disturbance into account in the design of the reference governor. Our Hamiltonian dynamics learning and tracking control techniques are compared to a GP MPC technique [45] and are demonstrated in a 3D environment using a simulated hexarotor robot to achieve collision-free autonomous navigation.

In summary, the contribution of this chapter is a tracking control design for Hamiltonian systems with learned dynamics, which achieves *robustness* to model estimation errors and *safety* with respect to state constraints.

3.2 Problem Statement

Consider a rigid body with position $\mathbf{p} \in \mathbb{R}^3$, orientation $\mathbf{R} \in SO(3)$, body-frame linear velocity $\mathbf{v} \in \mathbb{R}^3$, and body-frame angular velocity $\omega \in \mathbb{R}^3$. Let $\mathfrak{q} = [\mathbf{p}^\top \;\; \mathbf{r}_1^\top \;\; \mathbf{r}_2^\top \;\; \mathbf{r}_3^\top]^\top \in SE(3)$ denote the body's generalized coordinates, where $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3 \in \mathbb{R}^3$ are the rows of the rotation matrix $\mathbf{R}$. Let $\zeta = [\mathbf{v}^\top \;\; \omega^\top]^\top \in \mathbb{R}^6$ denote the body's generalized velocity. The generalized momentum $\mathfrak{p}$ of the body is defined as:

$$\mathfrak{p} = \mathbf{M}(\mathfrak{q})\zeta \in \mathbb{R}^6, \tag{3.1}$$

where $\mathbf{M}(\mathfrak{q}) \succ 0$ is the positive-definite generalized mass matrix. Let $\mathbf{x} = (\mathfrak{q}, \mathfrak{p}) \in T^*SE(3)$ denote the state of the rigid body system on the cotangent bundle $T^*SE(3)$ of the $SE(3)$ manifold. The Hamiltonian, $\mathscr{H}(\mathfrak{q}, \mathfrak{p})$, captures the total energy of the system as the sum of the kinetic energy $\mathscr{T}(\mathfrak{q}, \mathfrak{p}) = \frac{1}{2}\mathfrak{p}^\top \mathbf{M}^{-1}(\mathfrak{q})\mathfrak{p}$ and the potential energy $\mathscr{U}(\mathfrak{q})$:

$$\mathscr{H}(\mathfrak{q}, \mathfrak{p}) = \mathscr{T}(\mathfrak{q}, \mathfrak{p}) + \mathscr{U}(\mathfrak{q}). \tag{3.2}$$

The evolution of the state $\mathbf{x}$ is governed by Hamilton's equations of motion [66]:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}, \qquad\qquad \mathbf{x}(t_0) = \mathbf{x}_0,\\ &= \begin{bmatrix}\mathbf{0} & \mathfrak{q}^\times \\ -\mathfrak{q}^{\times\top} & \mathfrak{p}^\times\end{bmatrix}\begin{bmatrix}\nabla_{\mathfrak{q}}\mathcal{H}(\mathfrak{q},\mathfrak{p})\\ \nabla_{\mathfrak{p}}\mathcal{H}(\mathfrak{q},\mathfrak{p})\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{B}(\mathfrak{q})\end{bmatrix}\mathbf{u}\end{aligned} \tag{3.3}$$

where $\mathbf{u} \in \mathbb{R}^6$ is the control input, e.g. force and torque or motor speeds for a UAV system, $\mathbf{B}(\mathfrak{q}) \in \mathbb{R}^{6\times 6}$ is an input gain matrix, and the operators $\mathfrak{q}^\times$, $\mathfrak{p}^\times$ are defined as:

$$\mathfrak{q}^\times = \begin{bmatrix}\mathbf{R}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \hat{\mathbf{r}}_1^\top & \hat{\mathbf{r}}_2^\top & \hat{\mathbf{r}}_3^\top\end{bmatrix}^\top, \quad \mathfrak{p}^\times = \begin{bmatrix}\mathfrak{p}_{\mathbf{v}}\\ \mathfrak{p}_{\omega}\end{bmatrix}^\times = \begin{bmatrix}\mathbf{0} & \hat{\mathfrak{p}}_{\mathbf{v}}\\ \hat{\mathfrak{p}}_{\mathbf{v}} & \hat{\mathfrak{p}}_{\omega},\end{bmatrix},$$

where the hat map $\hat{\mathbf{w}}$ for $\mathbf{w} \in \mathbb{R}^3$ is:

$$\hat{\mathbf{w}} = \begin{bmatrix}0 & -w_3 & w_2\\ w_3 & 0 & -w_1\\ -w_2 & w_1 & 0\end{bmatrix}.$$

The Hamiltonian dynamics model in (3.3) can be extended to include energy dissipation in a port-Hamiltonian formulation [97] such as friction or drag forces [103]. However, for clarity of the control design, we leave this for future work.

We consider the case that the parameters of the Hamiltonian dynamics model in (3.3), including the mass $\mathbf{M}(\mathfrak{q})$, the potential energy $\mathcal{U}(\mathfrak{q})$, and the input matrix $\mathbf{B}(\mathfrak{q})$, are unknown. Instead, we are given a trajectory dataset $\mathscr{D} = \{t_{0:N}^{(i)}, \mathfrak{q}_{0:N}^{(i)}, \zeta_{0:N}^{(i)}, \mathbf{u}_{0:N-1}^{(i)}\}_{i=1}^{D}$ consisting of D sequences of generalized coordinates and velocities $(\mathfrak{q}_{0:N}^{(i)}, \zeta_{0:N}^{(i)})$ at times $t_0^{(i)} < t_1^{(i)} < \ldots < t_N^{(i)}$, collected by applying a constant control input $\mathbf{u}_n^{(i)}$ to the system with initial condition $(\mathfrak{q}_n^{(i)}, \zeta_n^{(i)})$ for $t \in [t_n, t_{n+1})$ and $n = 0, \ldots, N-1$. Our objective is to learn the system dynamics from the data set $\mathscr{D}$ and design a control policy $\mathbf{u} = \pi(\mathbf{x})$ such that the system follows a desired reference path without violating safety constraints. Let $\mathscr{O} \subset \mathbb{R}^3$ and $\mathscr{F} := \mathbb{R}^3 \setminus \mathscr{O}$ denote the unsafe (obstacle) set and the safe (obstacle-free) set, respectively. Denote the interior of $\mathscr{F}$ as $\text{Int}(\mathscr{F})$. We assume that $\mathscr{O}$ is not known apriori but the distance $\bar{d}(\mathbf{p}, \mathscr{O})$ from the system's position $\mathbf{p}$ to $\mathscr{O}$ can be sensed with a limited sensing range $d_{max} > 0$:

$$\bar{d}(\mathbf{p}, \mathscr{O}) := \min\{d(\mathbf{p}, \mathscr{O}), d_{max}\}, \tag{3.4}$$

where $d(\mathbf{p}, \mathscr{O}) := \inf_{\mathbf{a} \in \mathscr{O}} \|\mathbf{p} - \mathbf{a}\|$ denotes the Euclidean distance from $\mathbf{p}$ to the set $\mathscr{O}$. The safe tracking control problem considered in this paper is summarized below.

Problem 3. Let $\mathscr{D} = \{t^{(i)}_{0:N}, \mathfrak{q}^{(i)}_{0:N}, \zeta^{(i)}_{0:N}, \mathbf{u}^{(i)}_{0:N-1}\}_{i=1}^{D}$ be a training dataset of state-control trajectories obtained from a rigid-body system with unknown Hamiltonian dynamics in (3.3). Let $\mathbf{r} : [0, 1] \mapsto \text{Int}(\mathscr{F})$ be a continuous function specifying a desired position reference path for the system. Assume that the reference path starts at the initial position at time t_0, i.e., $\mathbf{r}(0) = \mathbf{p}(t_0) \in \text{Int}(\mathscr{F})$. Using local distance observations $\bar{d}(\mathbf{p}(t), \mathscr{O})$ of the unsafe set $\mathscr{O}$, design a control policy $\pi : T^*SE(3) \mapsto \mathbb{R}^6$ so that the position $\mathbf{p}(t)$ of the closed-loop system with control law $\mathbf{u} = \pi(\mathbf{x})$ converges asymptotically to $\mathbf{r}(1)$, while remaining safe, i.e., $\mathbf{p}(t) \in \mathscr{F}, \forall t \geq t_0$.

3.3 Learning $SE(3)$ Hamiltonian Dynamics from Data

In this section, we design a dynamics model that can be learned from a previously collected trajectory dataset, e.g., obtained from manual control, and is sufficiently general to represent different mobile robots, such as cars and drones. We describe how to learn Hamiltonian dynamics from the dataset $\mathscr{D} = \{t^{(i)}_{0:N}, \mathfrak{q}^{(i)}_{0:N}, \zeta^{(i)}_{0:N}, \mathbf{u}^{(i)}_{0:N-1}\}_{i=1}^{D}$, described in Sec. 3.2, using translation-equivariant Hamiltonian-based neural ODE networks [28]. The mass $\mathbf{M}(\mathfrak{q})$, the potential energy $\mathscr{U}(\mathfrak{q})$ and the input gain $\mathbf{B}(\mathfrak{q})$ are approximated by neural networks. We show that the model estimation errors caused by the trained neural networks can be considered as a disturbance applied on the learned system.

3.3.1 Translation-Equivariant $SE(3)$ Hamiltonian Dynamics Learning

Since the system dynamics does not change if we shift the position $\mathbf{p}$ to any points in the world frame, we offset the trajectories in the dataset $\mathscr{D}$ so that they start from the position $\mathbf{0}$ and learn the system dynamics well around the origin. This is sufficient for stabilization task, e.g. using the controller design in Sec. 3.4, because driving the system from state $\mathbf{x}$ with position $\mathbf{p}$ to a desired state $\mathbf{x}^*$ with position $\mathbf{p}^*$ is the same as driving the system from the state $\mathbf{x}$ with position $\mathbf{0}$ to a desired state $\mathbf{x}^*$ with offset position $\mathbf{p}^* - \mathbf{p}$.

Since the momentum $\mathfrak{p}$ is not directly available from the dataset $\mathscr{D}$, we use the time derivative of the generalized velocity, derived from (3.1):

$$\dot{\zeta} = \left(\frac{d}{dt}\mathbf{M}^{-1}(\mathfrak{q})\right)\mathfrak{p} + \mathbf{M}^{-1}(\mathfrak{q})\dot{\mathfrak{p}}. \tag{3.5}$$

Eq. (3.3) and (3.5) describe the Hamiltonian dynamics of the generalized coordinates and velocities with unknown inverse generalized mass matrix $\mathbf{M}^{-1}(\mathfrak{q})$, input matrix $\mathbf{B}(\mathfrak{q})$, and potential energy $\mathscr{U}(\mathfrak{q})$, for which we aim to approximate by three neural networks $\mathbf{M}_\theta^{-1}(\mathfrak{q}), \mathbf{B}_\theta(\mathfrak{q})$ and $\mathscr{U}_\theta(\mathfrak{q})$, respectively, with parameters θ.

To optimize for the parameters θ, we use the Hamiltonian-based neural ODE framework that encodes the Hamiltonian dynamics (3.3) and (3.5) with $\mathbf{M}_\theta(\mathfrak{q}), \mathbf{B}_\theta(\mathfrak{q})$ and $\mathscr{U}_\theta(\mathfrak{q})$ in the network structure (Fig. 3.1). The forward pass rolls out the dynamics $\bar{\mathbf{f}}_\theta$ described by (3.3) and (3.5) with the neural networks $\mathbf{M}_\theta(\mathfrak{q}), \mathbf{B}_\theta(\mathfrak{q})$ and $\mathscr{U}_\theta(\mathfrak{q})$ using a neural ODE solver ([22]) with initial state $(\mathfrak{q}_n^{(i)}, \zeta_n^{(i)})$. We obtain a predicted state $(\bar{\mathfrak{q}}_{n+1}^{(i)}, \bar{\zeta}_{n+1}^{(i)})$ at times $t_{n+1}^{(i)}$ for each $n = 0, \ldots, N-1$ and $i = 1, \ldots, D$ as:

$$(\bar{\mathfrak{q}}_{n+1}^{(i)}, \bar{\zeta}_{n+1}^{(i)}) = \text{ODESolver}\left((\mathfrak{q}_n^{(i)}, \zeta_n^{(i)}), \bar{\mathbf{f}}, t_{n+1}^{(i)} - t_n^{(i)}; \theta\right).$$

The loss function is defined as $\mathscr{L} = \sum_{i=1}^{D}\sum_{n=1}^{N} c(\mathfrak{q}_n^{(i)}, \zeta_n^{(i)}, \bar{\mathfrak{q}}_n^{(i)}, \bar{\zeta}_n^{(i)})$ where the distance metric c

is defined as the sum of position, orientation, and velocity errors on the tangent bundle $TSE(3)$:

$$c\left(\mathfrak{q}, \zeta, \bar{\mathfrak{q}}, \bar{\zeta}\right) = c_{\mathbf{p}}(\mathbf{p}, \bar{\mathbf{p}}) + c_{\mathbf{R}}(\mathbf{R}, \bar{\mathbf{R}}) + c_{\zeta}(\zeta, \bar{\zeta}), \tag{3.6}$$

with the position error $c_{\mathbf{p}}(\mathbf{p}, \bar{\mathbf{p}}) = \|\mathbf{p} - \bar{\mathbf{p}}\|_2^2$, the velocity error $c_{\zeta}(\zeta, \bar{\zeta}) = \|\zeta - \bar{\zeta}\|_2^2$, and the rotation error $c_{\mathbf{R}}(\mathbf{R}, \bar{\mathbf{R}}) = \| \left(\log(\bar{\mathbf{R}}\mathbf{R}^\top)\right)^\vee \|_2^2$. The log-map $\log(\cdot) : SE(3) \mapsto \mathfrak{so}(3)$ returns a skew-symmetric matrix in $\mathfrak{so}(3)$ from a rotation matrix in $SE(3)$, and the $\vee$-map $(\cdot)^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ is the inverse of the hat map $\hat{(\cdot)}$ in Sec. 3.2.

The network parameters θ are optimized using gradient descent by back-propagating the gradient $\nabla_\theta \mathscr{L}$ of the loss through the neural ODE solver efficiently using adjoint method [22]. Specifically, let $\mathbf{a} = \nabla_{\mathfrak{q},\zeta} \mathscr{L}$ be the adjoint state and $\mathbf{s} = ((\mathfrak{q}, \zeta), \mathbf{a}, \nabla_\theta \mathscr{L})$ be the augmented state. The augmented state dynamics are [22]:

$$\dot{\mathbf{s}} = \bar{\mathbf{f}}_{\mathbf{s}} = \left(\bar{\mathbf{f}}_\theta, -\mathbf{a}^\top \nabla_{\mathfrak{q},\zeta} \bar{\mathbf{f}}_\theta, -\mathbf{a}^\top \nabla_\theta \bar{\mathbf{f}}_\theta\right). \tag{3.7}$$

We obtain the gradient $\nabla_\theta \mathscr{L}$ by a single call to a reverse-time ODE solver starting from $\mathbf{s}_{n+1} = \mathbf{s}(t_{n+1})$:

$$\mathbf{s}_0 = (\bar{\mathbf{x}}_0, \mathbf{a}_0, \nabla_\theta \mathscr{L}) = \text{ODESolver}(\mathbf{s}_{n+1}, \bar{\mathbf{f}}_{\mathbf{s}}, t_{n+1} - t_n), \tag{3.8}$$

for $n = 0, \ldots, N-1$, and update the parameters θ using gradient descent. Please refer to [22] for more details.

3.3.2 Model Estimation Error as a Disturbance

Via the training process described in Sec. 3.3.1, we approximate the ground truth mass $\tilde{\mathbf{M}}(\mathfrak{q})$, potential energy $\tilde{\mathscr{U}}(\mathfrak{q})$ and input gain matrix $\tilde{\mathbf{B}}(\mathfrak{q})$ with the learned mass $\mathbf{M}_\theta(\mathfrak{q}) = \tilde{\mathbf{M}}(\mathfrak{q}) + \Delta\mathbf{M}_\theta(\mathfrak{q})$, potential energy $\mathscr{U}(\mathfrak{q}) = \tilde{\mathscr{U}}(\mathfrak{q}) + \Delta\mathscr{U}_\theta(\mathfrak{q})$, and input gain $\mathbf{B}(\mathfrak{q}) = \tilde{\mathbf{B}}(\mathfrak{q}) + \Delta\mathbf{B}_\theta(\mathfrak{q})$ where $\Delta\mathbf{M}_\theta(\mathfrak{q}), \Delta\mathscr{U}_\theta(\mathfrak{q})$, and $\Delta\mathbf{B}_\theta(\mathfrak{q})$ are the estimation errors. We drop the subscript θ to simplify the notations. The generalized coordinates $\mathfrak{q}$ and the ground-truth momentum $\tilde{\mathfrak{p}} := \tilde{\mathbf{M}}(\mathfrak{q})\zeta$, satisfy

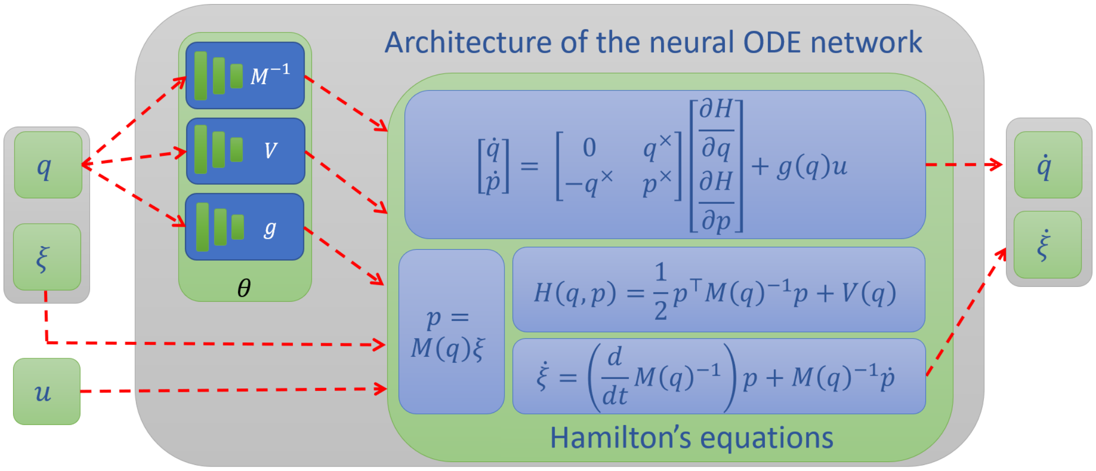

Architecture of the neural ODE network
q
ξ
u
M^{-1}
V
g
θ
$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}0 & q^\times\\ -q^\times & p^\times\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q}\\ \frac{\partial H}{\partial p}\end{bmatrix} + g(q)u$
$p = M(q)\xi$
$H(q,p) = \frac{1}{2}p^\top M(q)^{-1}p + V(q)$
$\dot{\xi} = \left(\frac{d}{dt}M(q)^{-1}\right)p + M(q)^{-1}\dot{p}$
Hamilton's equations
$\dot{q}$
$\dot{\xi}$

Figure 3.1. Architecture of $SE(3)$ Hamiltonian neural ODE network.

the Hamiltonian dynamics (3.3):

$$\begin{aligned}
\dot{\mathfrak{q}} &= \mathfrak{q}^\times \nabla_{\tilde{\mathfrak{p}}} \tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) = \mathfrak{q}^\times \zeta \\
\dot{\tilde{\mathfrak{p}}} &= -\mathfrak{q}^{\times\top} \nabla_{\mathfrak{q}} \tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) + \tilde{\mathfrak{p}}^\times \nabla_{\mathfrak{p}} \tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) + \tilde{\mathbf{B}}(\mathfrak{q})\mathbf{u} \\
&= -\mathfrak{q}^{\times\top} \nabla_{\mathfrak{q}} \tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) + \tilde{\mathfrak{p}}^\times \zeta + \tilde{\mathbf{B}}(\mathfrak{q})\mathbf{u},
\end{aligned} \tag{3.9}$$

with the ground-truth Hamiltonian

$$\tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) = \frac{1}{2}\tilde{\mathfrak{p}}^\top \tilde{\mathbf{M}}^{-1}(\mathfrak{q})\tilde{\mathfrak{p}} + \tilde{\mathscr{U}}(\mathfrak{q}) = \frac{1}{2}\zeta^\top \tilde{\mathbf{M}}(\mathfrak{q})\zeta + \tilde{\mathscr{U}}(\mathfrak{q}). \tag{3.10}$$

Meanwhile, for the generalized coordinates $\mathfrak{q}$ and the momentum $\mathfrak{p} := \mathbf{M}(\mathfrak{q})\zeta$, the Hamiltonian dynamics is learned from data and of the form:

$$\begin{aligned}
\dot{\mathfrak{q}} &= \mathfrak{q}^\times \nabla_{\mathfrak{p}} \mathscr{H}(\mathfrak{q},\mathfrak{p}) = \mathfrak{q}^\times \zeta \\
\dot{\mathfrak{p}} &= -\mathfrak{q}^{\times\top} \nabla_{\mathfrak{q}} \mathscr{H}(\mathfrak{q},\mathfrak{p}) + \mathfrak{p}^\times \nabla_{\mathfrak{p}} \mathscr{H}(\mathfrak{q},\mathfrak{p}) + \mathbf{B}(\mathfrak{q})\mathbf{u} \\
&= -\mathfrak{q}^{\times\top} \nabla_{\mathfrak{q}} \mathscr{H}(\mathfrak{q},\mathfrak{p}) + \mathfrak{p}^\times \zeta + \mathbf{B}(\mathfrak{q})\mathbf{u},
\end{aligned} \tag{3.11}$$

with the learned Hamiltonian

$$\mathscr{H}(\mathfrak{q},\mathfrak{p}) = \frac{1}{2}\zeta^\top \mathbf{M}(\mathfrak{q})\zeta + \mathscr{U}(\mathfrak{q}) = \tilde{\mathscr{H}}(\mathfrak{q},\tilde{\mathfrak{p}}) + \Delta\mathscr{H}(\mathfrak{q},\mathfrak{p}),$$

and its estimation error $\Delta\mathscr{H}(\mathfrak{q},\mathfrak{p}) = \frac{1}{2}\zeta^\top \Delta\mathbf{M}(\mathfrak{q})\zeta + \Delta\mathscr{U}(\mathfrak{q})$.

However, the learned dynamics (3.11) is only an approximation of the actual dynamics for $(\mathfrak{q},\mathfrak{p})$. While the dynamics of $\mathfrak{q}$ does not change, the actual dynamics of the learned momentum, $\mathfrak{p} = \mathbf{M}(\mathfrak{q})\zeta = \tilde{\mathfrak{p}} + \Delta\mathfrak{p}$, where $\Delta\mathfrak{p} = \Delta\mathbf{M}(\mathfrak{q})\zeta$, is derived from (3.9) as follows:

$$\begin{aligned} \dot{\mathfrak{p}} &= \dot{\tilde{\mathfrak{p}}} + \dot{\Delta\mathfrak{p}} \\ &= -\mathfrak{q}^{\times\top}\nabla_{\mathfrak{q}}\mathscr{H}(\mathfrak{q},\mathfrak{p}) + \mathfrak{p}^\times\zeta + \mathbf{B}(\mathfrak{q})\mathbf{u} \\ &\quad + \mathfrak{q}^{\times\top}\nabla_{\mathfrak{q}}(\Delta\mathscr{H}(\mathfrak{q},\mathfrak{p})) - \Delta\mathfrak{p}^\times\zeta - \Delta\mathbf{B}(\mathfrak{q})\mathbf{u} + \dot{\Delta\mathfrak{p}}. \\ &= -\mathfrak{q}^{\times\top}\nabla_{\mathfrak{q}}\mathscr{H}(\mathfrak{q},\mathfrak{p}) + \mathfrak{p}^\times\zeta + \mathbf{B}(\mathfrak{q})\mathbf{u} + \mathbf{d}_1, \end{aligned} \tag{3.12}$$

where the force

$$\mathbf{d}_1 \ := \ \mathfrak{q}^{\times\top}\nabla_{\mathfrak{q}}(\Delta\mathscr{H}(\mathfrak{q},\mathfrak{p})) - \Delta\mathfrak{p}^\times\zeta - \Delta\mathbf{B}(\mathfrak{q})\mathbf{u} + \dot{\Delta\mathfrak{p}}, \tag{3.13}$$

represents the effect of the model errors $\Delta\mathbf{M}(\mathfrak{q}), \Delta\mathscr{U}(\mathfrak{q})$, and $\Delta\mathbf{B}(\mathfrak{q})$ and is considered as a disturbance applied to the learned system (3.11). To improve the error $\mathbf{d}_1$ with respect to the position $\mathbf{p}$, we enforce translation-equivariance in the neural ODE model, as described in Sec. III.A, and learn the model well around the origin. This allows us to offset any position $\mathbf{p}$ to the well-learned region around the origin. To reduce the model error with respect to orientation, we collect a training dataset that covers different regions of roll, pitch, and yaw angles, e.g. by manually driving a UAV to different desired positions and yaw angles. A promising approach to estimate the disturbance magnitude is to employ a Bayesian formulation of the neural ODE network used to learn the dynamics model. A Bayesian model will provide a posterior distribution, rather than point estimates, for the model parameters (i.e. $\mathbf{M}^{-1}(\mathfrak{q})$, $\mathbf{B}(\mathfrak{q})$, and $\mathscr{U}(\mathfrak{q})$), whose variance can be used to obtain parameter error bounds and, in turn, a disturbance bound. Bayesian neural network models that can be used for dynamics learning include Bayesian neural ODE networks [24, 100], neural stochastic differential equation (SDE) networks [72], or Gaussian-process ODEs [43]. This motivates analyzing the robustness of our control design in next section to the disturbance $\mathbf{d}_1$ caused by the model errors.

3.4 Stabilization of Hamiltonian Dynamics with Matched Disturbances

As discussed in Sec.3.3.2, due to estimation errors in the dynamics learning process, the learned system model satisfies Hamilton's equations of motion in (3.3) subject to a matched disturbance signal $\mathbf{d}_1 : \mathbb{R} \mapsto \mathbb{R}^6$:

$$\begin{bmatrix} \dot{\mathfrak{q}} \\ \dot{\mathfrak{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathfrak{q}^{\times} \\ -\mathfrak{q}^{\times\top} & \mathfrak{p}^{\times} \end{bmatrix} \begin{bmatrix} \nabla_{\mathfrak{q}}\mathscr{H}(\mathfrak{q},\mathfrak{p}) \\ \nabla_{\mathfrak{p}}\mathscr{H}(\mathfrak{q},\mathfrak{p}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}(\mathfrak{q}) \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_1 \end{bmatrix}. \tag{3.14}$$

We consider a passivity-based stabilizing controller for (3.14), and analyze its robustness with respect to the disturbance signal $\mathbf{d}_1$ and its safety with respect to the obstacle set $\mathscr{O}$.

3.4.1 Passivity-based Control

Consider a desired regulation point $\mathbf{x}^* = (\mathfrak{q}^*, \mathfrak{p}^*)$ for the system in (3.14) with generalized coordinates $\mathfrak{q}^* = (\mathbf{p}^*, \mathbf{R}^*)$ and momentum $\mathfrak{p}^* = 0$. Since the Hamiltonian $\mathscr{H}(\mathbf{x})$ may not have a minimum at $\mathbf{x}^*$, the control signal $\mathbf{u}$ in (3.14) should be designed to inject additional energy $\mathscr{H}_a(\mathbf{x}, \mathbf{x}^*)$ into system and achieve a desired Hamiltonian $\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) = \mathscr{H}(\mathbf{x}) + \mathscr{H}_a(\mathbf{x}, \mathbf{x}^*)$, which is minimized at $\mathbf{x}^*$. This is the approach followed by interconnection and damping assignment passivity-based control (IDA-PBC) [81]. Let $\mathbf{x}_e = (\mathfrak{q}_e, \mathfrak{p}_e)$ denote the error in generalized coordinates and momentum:

$$\begin{aligned} \mathbf{R}_e &= \mathbf{R}^{*\top}\mathbf{R} = \begin{bmatrix} \mathbf{r}_{e1} & \mathbf{r}_{e2} & \mathbf{r}_{e3} \end{bmatrix}^\top & \mathbf{p}_e &= \mathbf{p} - \mathbf{p}^* \\ \mathfrak{q}_e &= \begin{bmatrix} \mathbf{p}_e^\top & \mathbf{r}_{e1}^\top & \mathbf{r}_{e2}^\top & \mathbf{r}_{e3}^\top \end{bmatrix}^\top & \mathfrak{p}_e &= \mathfrak{p} - \mathfrak{p}^*. \end{aligned} \tag{3.15}$$

A possible choice of $\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*)$, minimized at $\mathbf{x} = \mathbf{x}^*$, is:

$$\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) = \mathscr{T}(\mathfrak{q}_e, \mathfrak{p}_e) + \mathscr{U}_d(\mathfrak{q}_e) = \frac{1}{2}\mathfrak{p}_e^\top \mathbf{M}^{-1}(\mathfrak{q}_e)\mathfrak{p}_e + \frac{k_\mathbf{p}}{2}\|\mathbf{p}_e\|^2 + \frac{k_\mathbf{R}}{2}\operatorname{tr}(\mathbf{I} - \mathbf{R}_e) \tag{3.16}$$

where $k_\mathbf{p}$ and $k_\mathbf{R}$ are positive scalars.

The IDA-PBC method [28, 81] designs a controller $\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*)$ such that the closed-loop dynamics of the system in (3.14) are governed by the desired Hamiltonian in (3.16) as:

$$\begin{bmatrix} \dot{\mathfrak{q}}_e \\ \dot{\mathfrak{p}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{J}(\mathbf{x}, \mathbf{x}^*) \\ -\mathbf{J}(\mathbf{x}, \mathbf{x}^*)^\top & -\mathbf{K}_d \end{bmatrix} \begin{bmatrix} \nabla_{\mathfrak{q}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) \\ \nabla_{\mathfrak{p}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{3.17}$$

where the terms $\mathbf{J}(\mathbf{x}, \mathbf{x}^*)$, $\mathbf{K}_d$, and $\mathbf{d}$ in the transformed dynamics depend on the control design. To obtain the controller, one uses the relationship between $\mathbf{x}$ and $\mathbf{x}_e$ in (3.15) to equate the dynamics in (3.14) and (3.17), leading to:

$$\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*) = \mathbf{B}^\dagger(\mathfrak{q})\mathbf{b}(\mathbf{x}, \mathbf{x}^*), \tag{3.18}$$

where $\mathbf{B}^\dagger(\mathfrak{q}) = (\mathbf{B}^\top(\mathfrak{q})\mathbf{B}(\mathfrak{q}))^{-1}\mathbf{B}^\top(\mathfrak{q})$ is the pseudo-inverse of the input gain $\mathbf{B}(\mathfrak{q})$ and:

$$\mathbf{b}(\mathbf{x}, \mathbf{x}^*) = \Big(\mathfrak{q}^{\times\top}\nabla_\mathfrak{q}\mathscr{H}(\mathbf{x}) - \mathfrak{p}^\times\nabla_\mathfrak{p}\mathscr{H}(\mathbf{x}) - \mathbf{J}(\mathbf{x}, \mathbf{x}^*)^\top\nabla_{\mathfrak{q}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) - \mathbf{K}_d\nabla_{\mathfrak{p}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*)\Big) \tag{3.19}$$

with $\mathbf{J}(\mathbf{x},\mathbf{x}^*) := \begin{bmatrix} \mathbf{R}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{r}}_{e1}^\top & \hat{\mathbf{r}}_{e2}^\top & \hat{\mathbf{r}}_{e3}^\top \end{bmatrix}^\top$. If the IDA-PBC matching equations [13],

$$\mathbf{B}^\perp(\mathfrak{q})\mathbf{b}(\mathbf{x},\mathbf{x}^*) = 0, \tag{3.20}$$

are satisfied, where $\mathbf{B}^\perp(\mathfrak{q})$ is a maximal-rank left annihilator of $\mathbf{B}(\mathfrak{q})$, i.e., $\mathbf{B}^\perp(\mathfrak{q})\mathbf{B}(\mathfrak{q}) = \mathbf{0}$, then the controller in (3.18) achieves the desired closed-loop dynamics in (3.17) with $\mathbf{d} = \mathbf{d}_1$, i.e., without introducing any extra disturbance.

If the matching equations (3.20) cannot be satisfied globally, i.e., the IDA-PBC controller does not solve the system $\mathbf{B}(\mathfrak{q})\mathbf{u} = \mathbf{b}(\mathbf{x},\mathbf{x}^*)$ exactly, then $\pi(\mathbf{x}^*,\mathbf{x}) = \mathbf{B}^\dagger(\mathfrak{q})\mathbf{b}(\mathbf{x},\mathbf{x}^*)$ is a least-squares solution. In this case, the residual,

$$\mathbf{d}_2 := \left(\mathbf{B}(\mathfrak{q})\mathbf{B}^\dagger(\mathfrak{q}) - \mathbf{I}\right)\mathbf{b}(\mathbf{x},\mathbf{x}^*), \tag{3.21}$$

is introduced as an additional matched disturbance:

$$\mathbf{d} = \mathbf{d}_1 + \mathbf{d}_2 \tag{3.22}$$

in the closed-loop dynamics in (3.17). Since the magnitude of $\mathbf{d}_2$ is proportional to that of $\mathbf{b}(\mathbf{x},\mathbf{x}^*)$, it depends on the desired regulation point $\mathbf{x}^*$. An underactuated quadrotor system example is provided in Sec.3.7.4.

In general, the matching equations (3.20) are nonlinear PDEs and can be solved explicitly only for certain cases [13]. If $\mathbf{B}(\mathfrak{q})$ is invertible, i.e., the system in (3.14) is fully-actuated, then the solution in (3.18) exists and is unique. For systems with underactuation degree 1, the matching equations may be reduced to ODEs with closed-form solution [38] or solved with certain desired kinetic energy [1]. Yuksel et al. [101] solve the matching equations specifically for stabilizing a quadrotor system, using Euler angles instead of a rotation matrix. A survey on this topic is available in [13].

3.4.2 Robustness Analysis

In this section, we analyze the stability and robustness with respect to the disturbance signal $\mathbf{d}$ in (3.22) of the IDA-PBC controller in (3.18). Although the techniques we developed for dynamics learning in Sec. 3.3 and control synthesis in Sec. 3.4.1 did not make any assumptions about the Hamiltonian system in (3.14), our robustness and safety analysis that follows is developed under two assumptions.

Assumption 2. The disturbance signal (3.22) is uniformly bounded, i.e., $\|\mathbf{d}\| \leq \delta_{\mathbf{d}}$ for some constant $\delta_{\mathbf{d}} > 0$.

Assumption 3. The generalized mass matrix is constant, i.e., $\mathbf{M}(\mathfrak{q}) \equiv \mathbf{M}$.

Without Assumption 2, it is not possible to provide any performance guarantees for the control design because the disturbance $\mathbf{d}$ can have an arbitrary effect on the evolution of the closed-loop system dynamics. The disturbance magnitude bound $\delta_{\mathbf{d}}$ exists if we assume bounded estimation errors, bounded velocity and acceleration, bounded $\nabla_{\mathfrak{q}} (\Delta\mathscr{H}(\mathfrak{q},\mathfrak{p}))$, and bounded control input $\mathbf{u}$ from the controller (3.18).

Our robustness analysis in Thm. 3 below constructs an ISS-Lyapunov function [92] to handle the disturbance $\mathbf{d}$. Assumption 3 simplifies the proof that we have a valid ISS-Lyapunov function. Extending the analysis to handle a state-dependent mass $\mathbf{M}(\mathfrak{q})$ is left for future work. We simplify the error dynamics (3.17) by noting that:

$$\mathbf{e}(\mathbf{x},\mathbf{x}^*) := \mathbf{J}(\mathbf{x},\mathbf{x}^*)^\top \nabla_{\mathfrak{q}_e} \mathscr{H}_d(\mathbf{x},\mathbf{x}^*) = \begin{bmatrix} k_{\mathbf{p}} \mathbf{R}^\top \mathbf{p}_e \\ \frac{1}{2} k_{\mathbf{R}} \left(\mathbf{R}_e - \mathbf{R}_e^\top\right)^\vee \end{bmatrix},$$

which leads to:

$$\begin{aligned} \dot{\mathfrak{q}}_e &= \mathbf{J}(\mathbf{x},\mathbf{x}^*)\mathbf{M}^{-1}\mathfrak{p}_e, \\ \dot{\mathfrak{p}}_e &= -\mathbf{e}(\mathbf{x},\mathbf{x}^*) - \mathbf{K}_d \mathbf{M}^{-1}\mathfrak{p}_e + \mathbf{d}. \end{aligned} \tag{3.23}$$

Theorem 3. *Consider the Hamiltonian system in* (3.14) *with desired regulation point* $\mathbf{x}^* = (\mathfrak{q}^*, 0)$ *and control law specified in* (3.18) *with parameters* $k_{\mathbf{p}}$*,* $k_{\mathbf{R}}$*,* $\mathbf{K_d}$*. Assume that the initial state* $\mathbf{x}(t_0)$ *lies in the domain* $\mathscr{A} = \left\{\mathbf{x} \mid \mathrm{tr}(\mathbf{I} - \mathbf{R}^{*\top}\mathbf{R}) \leq \alpha < 4, \|\mathfrak{p}\| \leq \beta\right\}$ *for some positive constants* α *and* β*. Then, the function:*

$$\mathscr{V}(\mathbf{x},\mathbf{x}^*) = \mathscr{H}_d(\mathbf{x},\mathbf{x}^*) + \rho\frac{d}{dt}\mathscr{U}_d(\mathfrak{q}_e) \tag{3.24}$$

is an ISS-Lyapunov function [92] with respect to $\mathbf{d}$ *in* (3.22) *and satisfies:*

$$\begin{aligned} k_1\|\mathbf{z}\|^2 \leq \mathscr{V}(\mathbf{x},\mathbf{x}^*) &\leq k_2\|\mathbf{z}\|^2, \\ \dot{\mathscr{V}}(\mathbf{x},\mathbf{x}^*) &\leq -k_3\|\mathbf{z}\|^2 + k_\gamma\delta_{\mathbf{d}}^2, \end{aligned} \tag{3.25}$$

where $\mathbf{z} := [\|\mathbf{e}(\mathbf{x},\mathbf{x}^*)\| \ \|\mathfrak{p}_e\|]^\top \in \mathbb{R}^2$*,* $k_\gamma = \frac{1}{2\lambda_{\min}(\mathbf{K_d})} + \frac{\rho\lambda_2^2}{2\lambda_1}$*,* $\lambda_1 := \lambda_{\min}(\mathbf{M}^{-1})$*,* $\lambda_2 := \lambda_{\max}(\mathbf{M}^{-1})$*,* $k_1 = \frac{1}{2}\lambda_{\min}(\mathbf{Q}_1)$*,* $k_2 = \frac{1}{2}\lambda_{\max}(\mathbf{Q}_2)$*,* $k_3 = \frac{1}{2}\lambda_{\min}(\mathbf{Q}_3)$*, and the associated matrices* $\mathbf{Q}_1$*,* $\mathbf{Q}_2$*,* $\mathbf{Q}_3$ *are defined as:*

$$\mathbf{Q}_1 = \begin{bmatrix} \min\left\{k_{\mathbf{p}}^{-1}, k_{\mathbf{R}}^{-1}\right\} - \rho\lambda_2 \\ -\rho\lambda_2\lambda_1 \end{bmatrix} \quad \mathbf{Q}_2 = \begin{bmatrix} \max\left\{k_{\mathbf{p}}^{-1}, \frac{4k_{\mathbf{R}}^{-1}}{4-\alpha}\right\}\rho\lambda_2 \\ \rho\lambda_2\lambda_2 \end{bmatrix} \quad \mathbf{Q}_3 = \begin{bmatrix} q_1q_2 \\ q_2q_3 \end{bmatrix}, \tag{3.26}$$

where the elements of $\mathbf{Q}_3$ *are:*

$$\begin{aligned} q_1 &= \rho\lambda_1, \\ q_2 &= -\rho\left[\lambda_{\max}(\mathbf{M}^{-1}\mathbf{K_d}\mathbf{M}^{-1}) + \beta\lambda_2^2\right], \\ q_3 &= \lambda_{\min}(\mathbf{K_d})\lambda_1^2 - 2\rho\lambda_2^2\max\left\{k_{\mathbf{p}}, k_{\mathbf{R}}\right\}. \end{aligned} \tag{3.27}$$

Denote the sub-level set of $\mathscr{V}(\mathbf{x},\mathbf{x}^*)$ *with respect to positive scalar c as:* $\mathscr{S}_c := \{\mathbf{x} \mid \mathscr{V}(\mathbf{x},\mathbf{x}^*) \leq c\}$*.*

Given constants c_1, c_2 defined as:

$$c_1 := \frac{k_2 k_\gamma}{k_3}\delta_{\mathbf{d}}^2,\; c_2 := k_1 \min\left\{k_{\mathbf{R}}^2 \alpha(4-\alpha)/4, \beta^2\right\}, \tag{3.28}$$

$\mathscr{S}_{c_2} \subseteq \mathscr{A}$ is an estimate of the region of attraction of the control law in (3.18). *Any state* $\mathbf{x}$ *starting within $\mathscr{S}_{c_2}$ will converge exponentially to $\mathscr{S}_{c_1}$ and remain within it. The position error trajectory $\mathbf{p}_e(t)$ is uniformly ultimately bounded as:*

$$\lim_{t\to\infty} \|\mathbf{p}_e(t)\|^2 \le \frac{c_1}{k_1 k_{\mathbf{p}}^2} = \frac{k_2 k_\gamma}{k_1 k_3 k_{\mathbf{p}}^2}\delta_{\mathbf{d}}^2. \tag{3.29}$$

To ensure that $c_1 < c_2$, the disturbance bound $\delta_{\mathbf{d}}$ should satisfy $\delta_{\mathbf{d}} < \sqrt{\frac{c_2 k_3}{k_2 k_\gamma}}$.

Proof. See Appendix B.1. □

The estimates of the region of attraction and the uniform ultimate bound on the position error are provided by Thm. 3 for the IDA-PBC controller are conservative because our analysis considers the mass and inertia jointly as a generalized mass $\mathbf{M}$ and does not differentiate the force and torque disturbances. Besides considering separate disturbance bounds for the force and torque inputs, less conservative bounds can be achieved by introducing disturbance compensation as shown in [65].

3.4.3 Safety Analysis

Section 3.4.2 analyzed the stability and robustness properties of the IDA-PBC controller for a given regulation point $\mathbf{x}^*$. Next, we use the Lyapunov function $\mathscr{V}(\mathbf{x}, \mathbf{x}^*)$ in (3.24) to derive conditions under which the trajectory of the closed-loop system remains outside the unsafe set $\mathscr{O}$. We introduce a barrier function, which takes the region of attraction $\mathscr{S}_{c_2}$ of the controller and the invariant set $\mathscr{S}_{c_1}$ associated with the ultimate bound in Thm. 3 as well as the distance

$\bar{d}(\mathbf{p}^*, \mathscr{O})$ to $\mathscr{O}$ into account to quantify the margin to safety violation:

$$\Delta E(\mathbf{x}, \mathbf{x}^*) := \min\left\{c_2, k_1 k_{\mathbf{p}}^2 \bar{d}^2\left(\mathbf{p}^*, \mathscr{O}\right)\right\} - \mathscr{V}(\mathbf{x}, \mathbf{x}^*) + \max\left\{c_1 - \mathscr{V}(\mathbf{x}, \mathbf{x}^*), 0\right\}, \tag{3.30}$$

where k_1, $k_{\mathbf{p}}$, c_1, c_2 are the constants specified in Thm. 3. If, for a given regulation point $\mathbf{x}^*$, the safety margin $\Delta E(\mathbf{x}, \mathbf{x}^*)$ is positive initially, then any trajectory of the closed-loop system remains safe as it converges to the invariant set $\mathscr{S}_{c_1}$.

Proposition 2. *Consider the system in* (3.14) *with regulation point* $\mathbf{x}^* = (\mathrm{q}^*, 0)$ *and control law in* (3.18). *Suppose that the desired position* $\mathbf{p}^*$ *has sufficient clearance from the unsafe set* $\mathscr{O}$ *and the disturbance* $\mathbf{d}$ *is bounded as follows:*

$$\bar{d}^2(\mathbf{p}^*, \mathscr{O}) \geq \frac{k_2 k_\gamma}{k_1 k_3 k_{\mathbf{p}}^2} \delta_{\mathbf{d}}^2, \qquad \|\mathbf{d}\|^2 \leq \delta_{\mathbf{d}}^2 < \frac{c_2 k_3}{k_2 k_\gamma}. \tag{3.31}$$

If the initial state $\mathbf{x}(t_0) = \mathbf{x}_0$ *satisfies:*

$$\Delta E(\mathbf{x}_0, \mathbf{x}^*) \geq 0, \tag{3.32}$$

then the position error trajectory is uniformly ultimately bounded as in (3.29) *and the system remains safe, i.e.,* $d(\mathbf{p}(t), \mathscr{O}) \geq 0$ *for all* $t \geq t_0$.

Proof. By the definition in (3.30), $\Delta E(\mathbf{x}, \mathbf{x}^*) \geq 0$ implies that the Lyapunov function $\mathscr{V}(\mathbf{x}, \mathbf{x}^*)$ satisfies one of three cases:

1. $c_1 < \mathscr{V}$, $\mathscr{V} \leq \min\left\{c_2, k_1 k_{\mathbf{p}}^2 \bar{d}^2\left(\mathbf{p}^*, \mathscr{O}\right)\right\}$,
2. $c_1 \geq \mathscr{V}$, $\mathscr{V} \leq \min\left\{c_2, k_1 k_{\mathbf{p}}^2 \bar{d}^2\left(\mathbf{p}^*, \mathscr{O}\right)\right\}$,
3. $c_1 \geq \mathscr{V}$, $\mathscr{V} > \min\left\{c_2, k_1 k_{\mathbf{p}}^2 \bar{d}^2\left(\mathbf{p}^*, \mathscr{O}\right)\right\}$.

Case 3) can never happen because (3.31) implies that $c_1 \leq k_1 k_{\mathbf{p}}^2 \bar{d}^2\left(\mathbf{p}^*, \mathscr{O}\right)$ and $c_1 < c_2$.

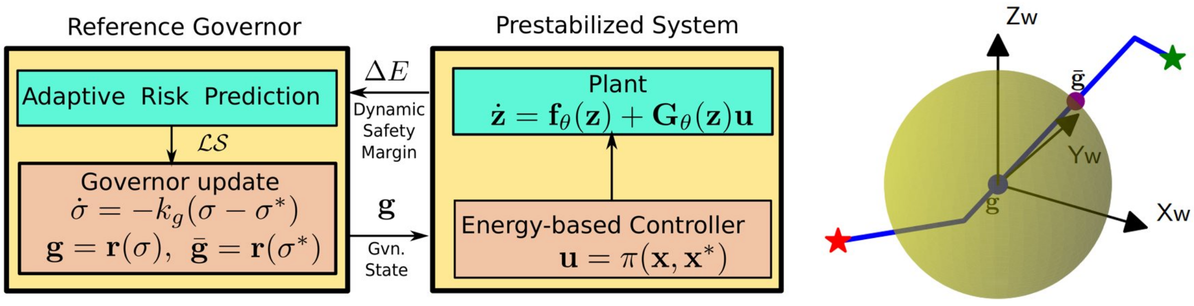

Reference Governor
Adaptive Risk Prediction
$\mathcal{LS}$
Governor update
$\dot{\sigma} = -k_g(\sigma - \sigma^*)$
$\mathbf{g} = \mathbf{r}(\sigma),\ \bar{\mathbf{g}} = \mathbf{r}(\sigma^*)$
ΔE
Dynamic Safety Margin
$\mathbf{g}$
Gvn. State
Prestabilized System
Plant
$\dot{\mathbf{z}} = \mathbf{f}_\theta(\mathbf{z}) + \mathbf{G}_\theta(\mathbf{z})\mathbf{u}$
Energy-based Controller
$\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*)$
Zw
$\bar{\mathbf{g}}$
Yw
$\mathbf{g}$
Xw

Figure 3.2. Structure of the reference-governor tracking controller (left). A governor with state $\mathbf{g}$ adaptively tracks desired path $\mathbf{r}$ and generates a reference point $\mathbf{x}^* = \ell(\mathbf{g})$ for the closed-loop Hamiltonian system (right). A local projected goal $\bar{\mathbf{g}}$ (purple dot) is generated as the farthest intersection between the local safe set $\mathscr{LS}(\mathbf{x},\mathbf{g})$ (yellow sphere) and the path $\mathbf{r}$ (blue curve) to guide the governor motion.

For Case 1), when $c_1 < \mathscr{V} \leq c_2$, we know from Thm. 3 that $\dot{\mathscr{V}} < 0$ and every trajectory starting in $\mathscr{S}_{c_2}$ converges exponentially to $\mathscr{S}_{c_1}$. In this case, from (3.25):

$$k_1 k_{\mathbf{p}}^2 \bar{d}^2(\mathbf{p}^*,\mathscr{O}) \geq \mathscr{V}(\mathbf{x}(t_0),\mathbf{x}^*) > \mathscr{V}(\mathbf{x}(t),\mathbf{x}^*) \geq k_1\|\mathbf{z}(t)\|^2 \geq k_1 k_{\mathbf{p}}^2 \|\mathbf{p}(t)-\mathbf{p}^*\|^2. \tag{3.33}$$

Therefore, $\|\mathbf{p}(t)-\mathbf{p}^*\|^2 \leq \bar{d}^2(\mathbf{p}^*,\mathscr{O}) \leq d^2(\mathbf{p}^*,\mathscr{O})$ and $d(\mathbf{p}(t),\mathscr{O}) \geq 0$ for all $t \in [t_0,t_1]$, where t_1 is the time when the trajectory enters $\mathscr{S}_{c_1}$, corresponding to Case 2) above.

For Case 2), we have $\mathscr{V}(\mathbf{x},\mathbf{x}^*) \leq c_1$ since (3.31) implies that $c_1 < c_2$. From Thm. 3, $\mathscr{S}_{c_1}$ is forward invariant and:

$$\|\mathbf{p}(t)-\mathbf{p}^*\|^2 \leq \frac{\mathscr{V}(\mathbf{x}(t),\mathbf{x}^*)}{k_1 k_{\mathbf{p}}^2} \leq \frac{c_1}{k_1 k_{\mathbf{p}}^2} = \frac{k_2 k_\gamma}{k_1 k_3 k_{\mathbf{p}}^2}\delta_{\mathbf{d}}^2. \tag{3.34}$$

Hence, (3.31) implies that $d(\mathbf{p}(t),\mathscr{O}) \geq 0$. □

3.5 Safe and Stable Tracking using a Reference Governor

In this section, we develop a safe tracking controller by introducing a reference governor [10] to guide the reference point $\mathbf{x}^*$ for the stabilizing control law $\pi(\mathbf{x},\mathbf{x}^*)$ in (3.18) along the desired reference path $\mathbf{r}$ introduced in Problem 3.

A reference governor is a virtual dynamical system whose state $\mathbf{g}(t)$ moves along $\mathbf{r}(\sigma)$ for $\sigma \in [0,1]$. In this paper, the governor state $\mathbf{g}(t) \in \mathbb{R}^3$ specifies a desired position $\mathbf{p}^*(t)$ for the Hamiltonian system. We introduce a lifting function $\mathbf{x}^*(t) = \ell(\mathbf{g}(t))$ to provide a desired orientation $\mathbf{R}^*(t)$ and specify a reference state $\mathbf{x}^*(t)$ for the Hamiltonian system.

Given $\mathbf{x}^*(t)$, we compute the safety margin $\Delta E(\mathbf{x}(t), \mathbf{x}^*(t))$ in (3.30) and use the leeway amount by which the margin exceeds 0 to move the governor state $\mathbf{g}(t)$ along $\mathbf{r}(\sigma)$. Intuitively, the reference point $\mathbf{x}^*(t) = \ell(\mathbf{g}(t))$ speeds up when $\Delta E(\mathbf{x}(t), \mathbf{x}^*(t))$ increases, e.g., the distance to obstacles increases or the system energy function decreases, and vice versa.

Given a point $\mathbf{g} = \mathbf{r}(\sigma)$ on the reference path for some $\sigma \in [0,1]$, we generate a reference state $\mathbf{x}^* = (\mathfrak{q}^*, \mathfrak{p}^*)$ where $\mathfrak{q}^* = (\mathbf{p}^*, \mathbf{R}^*) = (\mathbf{g}, \mathbf{I})$ and $\mathfrak{p}^* = 0$. The governor state $\mathbf{g}$ represents the desired position $\mathbf{p}^*$ on the path. For simplicity, we set the desired rotation matrix $\mathbf{R}^* = \mathbf{I}$. If, in addition to $\mathbf{r}$, a desired yaw angle reference is provided, one can generate $\mathbf{R}^*$ using the method described in [64] to achieve better orientation tracking. We define a lifting function $\ell : \mathbb{R}^3 \mapsto T^*SE(3)$ that generates a reference state $\mathbf{x}^* = \ell(\mathbf{g})$ from the governor state $\mathbf{g}$:

$$\ell(\mathbf{g}) := \begin{bmatrix} \mathbf{g}^\top & \mathbf{e}_1^\top & \mathbf{e}_2^\top & \mathbf{e}_3^\top & \mathbf{0}^\top & \mathbf{0}^\top \end{bmatrix}^\top, \tag{3.35}$$

where $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ are the rows of the identity matrix. Given the reference state $\mathbf{x}^* = \ell(\mathbf{g})$, we compute the safety margin $\Delta E(\mathbf{x}, \mathbf{x}^*)$ in (3.30) and describe how to update the governor state to ensure that safety is preserved.

We update the governor state $\mathbf{g}(t) = \mathbf{r}(\sigma(t))$ along the path by regulating the path parameter σ:

$$\mathbf{g}(t) = \mathbf{r}(\sigma(t)), \quad \dot{\sigma}(t) = -k_g(\sigma(t) - \sigma^*(t)), \tag{3.36}$$

where $k_g > 0$ is a control gain and $\sigma^*(t) \in [0,1]$ is a desired time-varying parameter, which we construct using the safety margin $\Delta E(\mathbf{x}, \mathbf{x}^*)$. We require $\sigma^*(t)$ to satisfy two conditions: 1) always stay ahead of the current $\sigma(t)$: $\sigma^*(t) \geq \sigma(t)$, $\forall t \geq t_0$, and 2) have distance $\|\sigma^*(t) - \sigma(t)\|$

proportional to $\Delta E(\mathbf{x}(t), \mathbf{x}^*(t))$. The first condition guarantees that the governor state $\mathbf{g}(t)$ moves forward along the path towards the goal $\mathbf{r}(1)$. The second condition allows the safety margin ΔE to adaptively regulate the governor state $\mathbf{g}(t)$ in order to ensure safety for the Hamiltonian system. To construct the desired path parameter σ^*, we define a *local safe zone* as a ball around the governor state $\mathbf{g}$ with radius $\Delta E(\mathbf{x}, \mathbf{x}^*)$ based on the state $\mathbf{x}$ and the reference state $\mathbf{x}^* = \ell(\mathbf{g})$.

Definition 6. A *local safe zone* is a subset of $\mathbb{R}^3$ that depends on the system state $\mathbf{x}$ and the governor state $\mathbf{g}$:

$$\mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g}) := \left\{ \mathbf{q} \in \mathbb{R}^3 \mid \|\mathbf{q} - \mathbf{g}\|^2 \leq \Delta E(\mathbf{x}, \ell(\mathbf{g})) \right\}, \tag{3.37}$$

where ℓ is the lifting function in (3.35) and ΔE is the safety margin in (3.30).

We determine σ^* as the farthest intersection between the local safe zone $\mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g})$ and the path $\mathbf{r}$ by solving the scalar optimization problem in Def. 7.

Definition 7. A *local projected goal* for system-governor state $(\mathbf{x}, \mathbf{g})$ is a point $\bar{\mathbf{g}} \in \mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g})$ that is farthest along the path $\mathbf{r}$:

$$\bar{\mathbf{g}} = \mathbf{r}(\sigma^*), \;\; \sigma^* = \operatorname*{argmax}_{\sigma \in [0,1]} \left\{ \sigma \mid \mathbf{r}(\sigma) \in \mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g}) \right\}. \tag{3.38}$$

The construction of the local projected goal $\bar{\mathbf{g}}$ is shown in Fig. 3.2 (right), showing a reference path $\mathbf{r}$, the local safe zone $\mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g})$ and the local projected goal $\bar{\mathbf{g}}$. A local projected goal $\bar{\mathbf{g}}$ (purple dot) is generated as the farthest intersection between the local safe set $\mathscr{L}\mathscr{S}(\mathbf{x}, \mathbf{g})$ (yellow sphere) and the path $\mathbf{r}$ (blue curve) to guide the governor motion. This constructing of σ^* and $\bar{\mathbf{g}}$ completes the governor update law (3.36).

Our safe tracking controller consists of the reference governor system in (3.36), adaptively updating the reference point $\mathbf{x}^* = \ell(\mathbf{g})$ via the lifting function in (3.35), and the passivity-based controller in (3.18) that drives the Hamiltonian system towards $\mathbf{x}^*$. The stability, safety, and robustness of the proposed tracking controller are analyzed in Thm. 4.

Theorem 4. *Suppose that the desired path $\mathbf{r}(\sigma)$ has sufficient clearance from the unsafe set $\mathscr{O}$ and the disturbance $\mathbf{d}$ is bounded as:*

$$\min_{\sigma\in[0,1]} \bar{d}^2(\mathbf{r}(\sigma),\mathscr{O}) \geq \frac{k_2 k_\gamma}{k_1 k_3 k_{\mathbf{p}}^2}\delta_{\mathbf{d}}^2, \quad \|\mathbf{d}\|^2 \leq \delta_{\mathbf{d}}^2 < \frac{c_2 k_3}{k_2 k_\gamma}.$$

Consider the Hamiltonian system in (3.14)*, the governor system in* (3.36) *with σ^* constructed in Def. 7 and the control law $\mathbf{u} = \pi(\mathbf{x},\ell(\mathbf{g}))$ in* (3.18)*. Suppose that the initial state $(\mathbf{x}_0,\mathbf{g}_0)$ satisfies:*

$$\Delta E\left(\mathbf{x}_0,\ell(\mathbf{g}_0)\right) > 0, \quad \mathbf{g}_0 = \mathbf{r}(0) = \mathbf{p}(t_0), \tag{3.39}$$

where $\Delta E(\mathbf{x},\mathbf{x}^*)$ is the safety margin in (3.30)*. The position $\mathbf{p}(t)$ of* (3.14) *converges to a ball of radius $\sqrt{\frac{k_2 k_\gamma}{k_1 k_3 k_{\mathbf{p}}^2}}\delta_{\mathbf{d}}$ around $\mathbf{r}(1)$ and remains safe, i.e. $\mathbf{p}(t) \in \mathscr{F}$, for all $t \geq t_0$*

Proof. To simplify notation, let $\Delta E(t) = \Delta E\left(\mathbf{x}(t),\ell(\mathbf{g}(t))\right)$. Initially, $\mathbf{g}_0 = \mathbf{p}(t_0) = \mathbf{r}(0) \in \mathscr{LS}(\mathbf{x}_0,\mathbf{g}_0)$ and $\Delta E(t_0) > 0$. The local projected goal $\bar{\mathbf{g}}$ and the associated σ^* are well defined in Def. 7. By the governor update law (3.36), the path parameter σ increases, the governor state $\mathbf{g}(\sigma)$ moves along $\mathbf{r}$ towards the goal $\mathbf{r}(1)$. The desired state $\mathbf{x}^* = \ell(\mathbf{g})$ is updated via the lifting function (3.35). As $\mathbf{g}$ tracks $\bar{\mathbf{g}}$ on the path $\mathbf{r}$ via the path parameter update in (3.36), the system state $\mathbf{x}$ tracks $\mathbf{x}^* = \ell(\mathbf{g})$. During this process, the safety margin $\Delta E(t)$ fluctuates and regulates the rate of change of σ.

Since $\sigma^*(t)$ is bounded in (3.38), $\sigma(t)$ is updated continuously [30] in (3.36), leading to a continuous governor state $\mathbf{g}(t)$. By construction, the lifting function $\ell(\mathbf{g})$ is continuous in $\mathbf{g}$. Therefore, the reference point $\mathbf{x}^*(t) = \ell(\mathbf{g}(t))$ is continuous in time, leading to a continuous Lyapunov function $\mathscr{V}(\mathbf{x},\mathbf{x}^*)$ and a continuous safety margin $\Delta E(t)$. As a result, the safety margin $\Delta E(t)$ cannot become negative without crossing 0 from above at some time T_0. As $\Delta E(t) \downarrow 0$, the local safe zone shrinks to a point, i.e., $\mathscr{LS}(\mathbf{x},\mathbf{g}) \downarrow \{\mathbf{g}\}$. This immediately stops the the governor because $\bar{\mathbf{g}} = \mathbf{g}(T_0) = \mathbf{r}(\sigma(T_0))$ and $\dot{\sigma}(T_0) = 0$.

As a result, Proposition 2 states that $\mathbf{x}(t)$ stays within the invariant set $\mathscr{S}_{c_2}(\mathbf{x}^*(T_0))$ for

$t \geq T_0$ and converges to $\mathcal{S}_{c_1}(\mathbf{x}^*(T_0))$ without leaving $\mathcal{F}$. Eq. (3.30) shows that $\Delta E(t) = 0$ implies $c_1 \leq \mathcal{V}(t) \leq c_2$. By Thm. 3, as $\mathbf{x}(t)$ approaches $\mathbf{x}^*(T_0)$, we have $\dot{\mathcal{V}}(T_0) < 0$, i.e., the Lyapunov function $\mathcal{V}$ is decreasing. There exists $h > 0$ such that $\Delta E(T_0 + h)$ becomes strictly positive. Hence, the governor is able to move again towards a new $\bar{\mathbf{g}}$ generated by the positive $\Delta E(T_0 + h)$. This process continues until the governor state $\mathbf{g}(t)$ converges to $\mathbf{r}(1)$, the closed-loop system converges to the region $\mathcal{S}_{c_1}(\ell(\mathbf{r}(1)))$ and the position $\mathbf{p}(t)$ satisfies the uniform ultimate bound in (3.29) around $\mathbf{r}(1)$. □

Note that while our control design does not account for state estimation errors, e.g. from an odometry algorithm with a sensor setup (e.g. stereo camera, LiDAR, or visual-inertial), we can conservatively handle the errors by reducing the obstacle distance $\bar{d}$ in the safety margin specification in (3.31).

3.6 Application to Hamiltonian Dynamics in $\mathbb{R}^n$

In this section, we show that our control design can be easily modified and applied to a Hamiltonian system with configuration $\mathfrak{q}$ in $\mathbb{R}^n$ and dynamics:

$$\begin{bmatrix} \dot{\mathfrak{q}} \\ \dot{\mathfrak{p}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{0} \end{bmatrix} \begin{bmatrix} \nabla_{\mathfrak{q}} \mathcal{H}(\mathfrak{q}, \mathfrak{p}) \\ \nabla_{\mathfrak{p}} \mathcal{H}(\mathfrak{q}, \mathfrak{p}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}(\mathfrak{q}) \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_1 \end{bmatrix} \tag{3.40}$$

where the Hamiltonian $\mathcal{H}(\mathfrak{q}, \mathfrak{p})$ is defined as:

$$\mathcal{H}(\mathfrak{q}, \mathfrak{p}) = \frac{1}{2} \mathfrak{p}^\top \mathbf{M}^{-1}(\mathfrak{q}) \mathfrak{p} + \mathcal{U}(\mathfrak{q}). \tag{3.41}$$

Given a desired regulation point $\mathbf{x}^* = (\mathfrak{q}^*, \mathfrak{p}^*)$ with momentum $\mathfrak{p}^* = 0$, define the error state $\mathbf{x}_e = (\mathfrak{q}_e, \mathfrak{p}_e)$ as:

$$\mathfrak{q}_e = \mathfrak{q} - \mathfrak{q}^*, \qquad \mathfrak{p}_e = \mathfrak{p} - \mathfrak{p}^*. \tag{3.42}$$

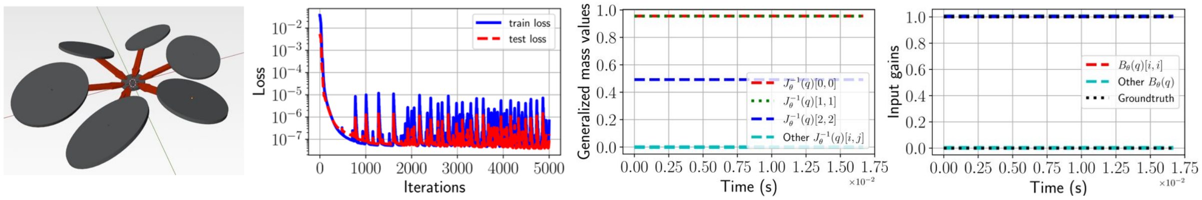


Figure 3.3. $SE(3)$ Hamiltonian neural ODE network (left to right): (a) simulated hexarotor for evaluation, (b) training loss, (c) learned inverse inertia $\mathbf{J}_\theta(\mathfrak{q})^{-1}$, and (d) learned input matrix $\mathbf{B}_\theta(\mathfrak{q})$ along a test trajectory, evaluated on the simulated hexarotor.

A desired Hamiltonian, minimized at $\mathbf{x} = \mathbf{x}^*$, is:

$$\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) = \frac{1}{2}\mathfrak{p}_e^\top \mathbf{M}^{-1}(\mathfrak{q}_e)\mathfrak{p}_e + \frac{k_\mathbf{p}}{2}\|\mathfrak{q}_e\|^2. \tag{3.43}$$

The IDA-PBC controller:

$$\mathbf{u} = \pi(\mathbf{x}, \mathbf{x}^*) = \mathbf{B}^\dagger(\mathfrak{q})\mathbf{b}(\mathbf{x}, \mathbf{x}^*) \tag{3.44}$$

with $\mathbf{b}(\mathbf{x}, \mathbf{x}^*) = \nabla_\mathfrak{q}\mathscr{H}(\mathbf{x}) - \nabla_{\mathfrak{q}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) - \mathbf{K_d}\nabla_{\mathfrak{p}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*)$ achieves the closed-loop dynamics:

$$\begin{bmatrix} \dot{\mathfrak{q}}_e \\ \dot{\mathfrak{p}}_e \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ -\mathbf{I}_n & -\mathbf{K}_d \end{bmatrix} \begin{bmatrix} \nabla_{\mathfrak{q}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) \\ \nabla_{\mathfrak{p}_e}\mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{3.45}$$

where $\mathbf{d} = \mathbf{d}_1 + \mathbf{d}_2$ as in (3.22) and $\mathbf{d}_2$ is as in (3.21).

Theorem 5. *Consider the Hamiltonian system in* (3.40) *with desired regulation point* $\mathbf{x}^* = (\mathfrak{q}^*, \mathbf{0})$ *and control law in* (3.44) *with parameters* $k_\mathbf{p}$, $\mathbf{K_d}$. *Under Assumptions 2 & 3, the function:*

$$\mathscr{V}(\mathbf{x}, \mathbf{x}^*) = \mathscr{H}_d(\mathbf{x}, \mathbf{x}^*) + \rho\frac{d}{dt}\mathscr{U}_d(\mathfrak{q}_e) \tag{3.46}$$

with $\mathscr{U}_d(\mathfrak{q}_e) = \frac{k_\mathbf{p}}{2}\|\mathfrak{q}_e\|^2$ *is an ISS-Lyapunov function [92] with respect to* $\mathbf{d}$ *and satisfies:*

$$\begin{aligned} k_1\|\mathbf{z}\|^2 \leq \mathscr{V}(\mathbf{x}, \mathbf{x}^*) \leq k_2\|\mathbf{z}\|^2, \\ \dot{\mathscr{V}}(\mathbf{x}, \mathbf{x}^*) \leq -k_3\|\mathbf{z}\|^2 + k_\gamma\delta_\mathbf{d}^2, \end{aligned} \tag{3.47}$$

where $\mathbf{z} := [k_\mathbf{p}\|\mathfrak{q}_e\| \;\|\mathfrak{p}_e\|]^\top \in \mathbb{R}^2$, $k_\gamma = \frac{1}{2\lambda_{\min}(\mathbf{K_d})} + \frac{\rho\lambda_2^2}{2\lambda_1}$, $\lambda_1 := \lambda_{\min}(\mathbf{M}^{-1})$, $\lambda_2 := \lambda_{\max}(\mathbf{M}^{-1})$, $k_1 = \frac{1}{2}\lambda_{\min}(\mathbf{Q}_1)$, $k_2 = \frac{1}{2}\lambda_{\max}(\mathbf{Q}_2)$, $k_3 = \frac{1}{2}\lambda_{\min}(\mathbf{Q}_3)$, *and the associated matrices* $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$ *are defined as:*

$$\begin{aligned} \mathbf{Q}_1 &= \begin{bmatrix} k_\mathbf{p}^{-1} & -\rho\lambda_2 \\ -\rho\lambda_2 & \lambda_1 \end{bmatrix}, \quad \mathbf{Q}_2 = \begin{bmatrix} k_\mathbf{p}^{-1} & \rho\lambda_2 \\ \rho\lambda_2 & \lambda_2 \end{bmatrix}, \\ \mathbf{Q}_3 &= \begin{bmatrix} \rho\lambda_1 & -\rho\gamma_d\lambda_2^2 \\ -\rho\gamma_d\lambda_2^2 & \gamma_d\lambda_1^2 - 2\rho\lambda_2^2 k_\mathbf{p} \end{bmatrix}. \end{aligned} \tag{3.48}$$

Any initial state $\mathbf{x}$ *converges exponentially to* $\mathscr{S}_{c_1} = \{\mathbf{x} | \mathscr{V}(\mathbf{x},\mathbf{x}^*) \leq c_1\}$ *with* $c_1 := \frac{k_2 k_\gamma}{k_3}\delta_\mathbf{d}^2$ *and remains within. The error trajectory* $\mathfrak{q}_e(t)$ *is uniformly ultimately bounded:*

$$\lim_{t\to\infty} \|\mathfrak{q}_e(t)\|^2 \leq \frac{c_1}{k_1 k_\mathbf{p}^2} = \frac{k_2 k_\gamma}{k_1 k_3 k_\mathbf{p}^2}\delta_\mathbf{d}^2. \tag{3.49}$$

The proof of Thm. 5 follows the same steps as that of Thm. 3, and is omitted due to space limitations. In contrast to Thm. 3, the result in Thm. 5 for $\mathbb{R}^n$ holds globally, i.e., the region of attraction is $\mathscr{A} = \mathbb{R}^n \times \mathbb{R}^n$. Thus, the disturbance magnitude bound $\delta_\mathbf{d}$ can be arbitrarily large.

The safety analysis in Sec. 3.4.3 can be modified with a new safety margin:

$$\Delta E(\mathbf{x},\mathbf{x}^*) := k_1 k_\mathbf{p}^2 \bar{d}^2(\mathfrak{q}^*, \mathscr{O}) - \mathscr{V}(\mathbf{x},\mathbf{x}^*) + \max\{c_1 - \mathscr{V}(\mathbf{x},\mathbf{x}^*), 0\}, \tag{3.50}$$

as Thm. 5 holds globally. The reference governor lifting function can be chosen as $\ell(\mathbf{g}) = [\mathbf{g}^\top \; \mathbf{0}^\top]^\top$. The governor state update remains the same as in (3.36). The robustness analysis extends the safe tracking results in [68] and [6].

3.7 Evaluation

We evaluate our robust and safe tracking controller using simulated hexarotor and quadrotor robots in 2D and 3D environments with ground-truth mass $m = 6.77$ kg, and inertia

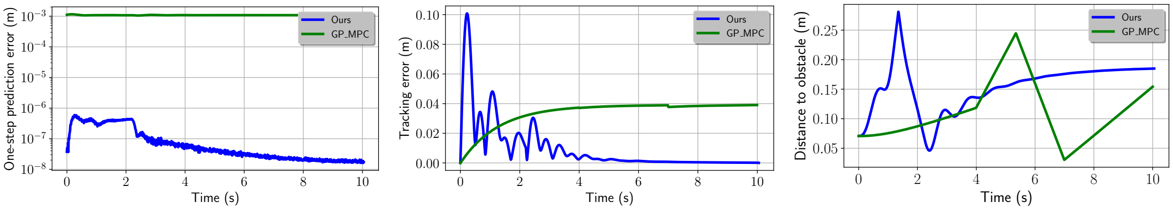


Figure 3.4. Comparison of position prediction errors (left) between the learned Hamiltonian dynamics and GP model, tracking errors (middle) and distance to obstacles (right) between our safe tracking controller and GP-MPC.

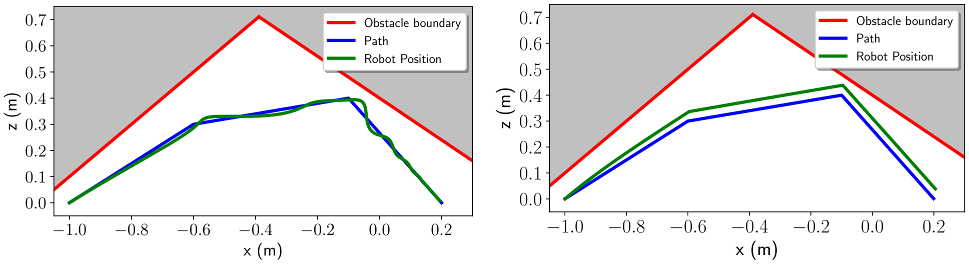


Figure 3.5. Path tracking with our approach (left) and GP-MPC [45] (right).

matrix $\mathbf{J} = \text{diag}([1.05, 1.05, 2.05])$ kg $\cdot$ m^2 , inspired by the solar-powered UAV in [55]. The robot's ground-truth dynamics satisfy Hamilton's equations (3.3) with generalized mass $\mathbf{M}(\mathfrak{q}) = \text{diag}(m\mathbf{I}, \mathbf{J})$, potential energy $\mathscr{U}(\mathfrak{q}) = mg \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{p}$, where $\mathbf{p}$ is the position and $g \approx 9.8ms^{-2}$ is the gravitational acceleration. The input matrices for the hexarotor and the quadrotor are $\mathbf{B}(\mathfrak{q}) = \mathbf{I}$ and $\mathbf{B}(\mathfrak{q}) = \begin{bmatrix} \mathbf{0}_{4\times 2} & \mathbf{I}_{4\times 4} \end{bmatrix}^\top$, respectively. The control input $\mathbf{u}$ of the hexarotor includes a 3D force and a 3D torque while that of the quadrotor includes a scalar force and a 3D torque.

For all experiments, the following control gains are used for our controller in Sec. 3.4.1: $k_\mathbf{p} = 20$, $k_\mathbf{R} = 50$, $\mathbf{K}_d = 15\mathbf{I}$ in (3.16). The parameters shown in Thm. 3 are: $\alpha = 2$, $\beta = 20$, $c_1 = 2.2050$, $c_2 = 8.8200$, $\rho = 3.5822 \times 10^{-5}$. The control gain for the governor in (3.36) is $k_g = 0.5$. The control loop frequency for all experiments is at 120 Hz.

While, our evaluation focuses on rotorcraft aerial robots, the methodology for system identification and control synthesis proposed in this paper is general. The exact same approach is applied to hexarotor, quadrotor and other ground and marine vehicles. This is in contrast with other system identification and control synthesis methods, which require knowledge of the dynamics structure, careful experiment design, and domain expertise for the particular system.

3.7.1 Evaluation of $SE(3)$ Hamiltonian Dynamics Learning

We consider a simulated hexarotor unmanned aerial vehicle (UAV) (Fig. 3.3) with fixed-tilt rotors pointing in different directions [86] and a simulated quadrotor UAV. Since the mass m of the UAVs can be easily measured, we assume the mass m is known, leading to a known potential energy $\mathscr{U}(\mathfrak{q}) = mg[001]\mathbf{p}$. We approximate the inverse generalized mass matrix by $\mathbf{M}_{\theta}^{-1}(\mathfrak{q}) = \text{diag}(m^{-1}\mathbf{I}, \mathbf{J}_{\theta}^{-1}(\mathfrak{q}))$ and learn $\mathbf{J}_{\theta}(\mathfrak{q})^{-1}$ and $\mathbf{B}_{\theta}(\mathfrak{q})$ from data.

We mimic manual flights in an area free of obstacles using a PID controller and drive the UAVs from a random initial pose to 18 desired poses, generating 18 1-second trajectories. We shift the trajectories to start from the origin and create a dataset $\mathscr{D} = \{t_{0:N}^{(i)}, \mathfrak{q}_{0:N}^{(i)}, \zeta_{0:N}^{(i)}, \mathbf{u}_{0:N-1}^{(i)})\}_{i=1}^{D}$ with $N = 24$ and $D = 18$. The Hamiltonian-based neural ODE network is trained with the dataset $\mathscr{D}$, as described in Sec. 3.3, for 5000 iterations and learning rate 10^{-4}. For the hexarotor, Fig. 3.3(c) shows the loss function during training. Note that if we scale $\mathbf{M}_{\theta}(\mathfrak{q})$ and the input matrix $\mathbf{B}(\mathfrak{q})$ by a constant γ, the dynamics of $(\mathfrak{q}, \zeta)$ in (3.3) and (3.5) does not change. Fig. 3.3(d) and 3.3(e) plot the scaled version of the learned inverse mass $\mathbf{J}_{\theta}(\mathfrak{q})^{-1}$ and the input matrix $\mathbf{B}_{\theta}(\mathfrak{q})$, converging to the constant ground truth values. We achieve similar results for the quadrotor using the same training process.

3.7.2 Evaluation of Robust Safe Tracking Control of a Learned 2D Hexarotor Hamiltonian Model

Next, we compare our approach with a GP-MPC technique [45] using a simulated 2D fully-actuated hexarotor UAV, moving on the xz-plane with position $\mathbf{p} = \begin{bmatrix} x, 0, z \end{bmatrix}$ and orientation $\mathbf{R} = \mathbf{R}_{\psi}$ determined by the pitch angle ψ. The control input is a 3D wrench, including a 2D force and a 1D torque. As we only consider the pitch angle ψ, we are interested in the inertia value J^{yy} and ignore J^{xx} and J^{zz}. We assume that the generalized mass m and J^{yy} are unknown for the 2D hexarotor and approximated by m_{θ} and J_{θ}^{yy}, respectively. The input gain $\mathbf{B}(\mathfrak{q})$ is assumed known.

Let $m_0 = 1.5m$ and $J_0^{yy} = 1.5J^{yy}$ be initial guesses of the mass m and the inertia J^{yy}. We model the approximated mass inverse m_θ^{-1} and inertia inverse J_θ^{yy-1} as:

$$m_\theta^{-1} = \left(\sqrt{m_0^{-1}} + L_1(\mathsf{q};\theta)\right)^2, J_\theta^{yy-1} = \left(\sqrt{J_0^{yy-1}} + L_2(\mathsf{q};\theta)\right)^2,$$

where $L_1(\mathsf{q};\theta)$ and $L_2(\mathsf{q};\theta)$ are two neural networks, representing the residual mass inverse and inertia inverse to be learned. In GP-MPC [45], the dynamics (3.3) are split into a prior nominal model with the prior mass m_0 and inertia J_0^{yy}, and residual dynamics, modeled by a GP regression model.

To collect training data, we place the simulated hexarotor at an initial location $(x,z) = (-1,0)$ and apply random control inputs to obtain $\mathscr{D} = \{t_{0:1}^{(i)}, \mathsf{q}_{0:1}^{(i)}, \zeta_{0:1}^{(i)}, \mathbf{u}_0^{(i)}\}_{i=1}^{150}$. Our Hamiltonian neural ODE network is trained with the dataset $\mathscr{D}$, as described in Sec. 3.3. For GP-MPC, the same dataset $\mathscr{D}$ is used to train a GP regression model of the residual dynamics as described in [45] and implemented in [18].

We assume there are two walls in the environments, generating two safety constraints on the robot position: $-x+z < 1.1$, $0.8x+z < 0.4$. The task is to track a predefined piecewise linear path $\mathbf{r}$, shown in Fig. 3.5, while safely avoiding collision with the walls. We adapt the GP-MPC implementation by [18] for the 2D hexarotor and enforce the safety constraints probabilistically with 95% confidence interval using the GP model uncertainty. To propagate the model uncertainty through a horizon of 10 time steps, we linearize the dynamics model around the hovering state and propagate the state mean and covariance using the mean equivalence technique [45, 18] with a time step of $1/120$ s. Meanwhile, our learned Hamiltonian neural ODE model is used with the safe tracking controller described in Sec. 3.5 to perform the task and enforce safety constraints.

Fig. 3.4 compares the prediction errors of our learned neural ODE network and the GP model. We collect the robot states and control inputs, generated by our controller while tracking the path, and predict the next state. Fig. 3.4 (left) plots the prediction error over time, showing

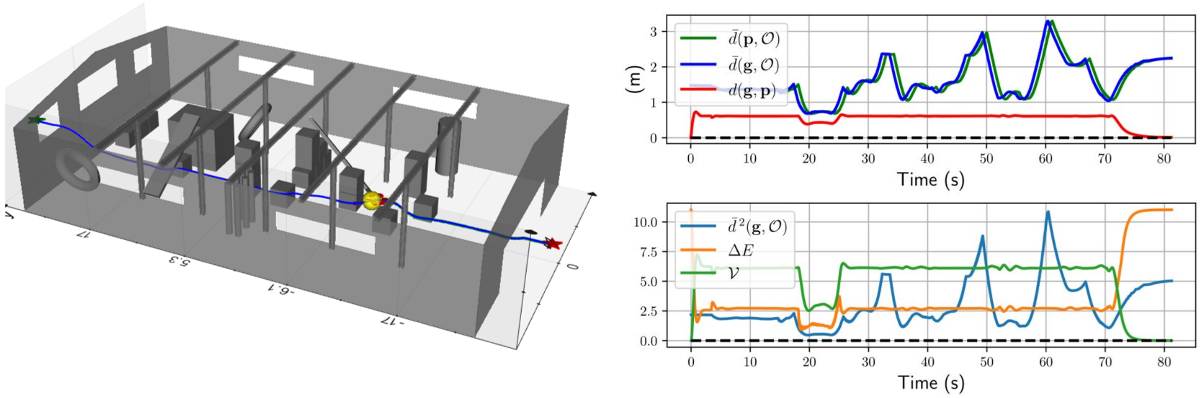


Figure 3.6. Safe navigation of a hexarotor system using learned model in a warehouse (left). The hexarotor (red body) navigates from a start (red star) to a goal location (green star) while avoiding obstacles. The obstacles are sensed by a simulated LiDAR sensor. The reference path, the robot path are shown in blue and green, respectively. Local safe set is shown in yellow sphere. The right plots show the dynamic safety margin ΔE, the Lyapunov function $\mathscr{V}$, and the distance to the obstacles $\bar{d}(\mathbf{p}(t),\mathscr{O})$, indicating that the safety constraints are never violated.

that we achieve better prediction than the trained GP model. This reflects the difference between our model, which encodes the Hamiltonian structure and translation equivariance in the network architecture, and the GP model, which incurs higher model uncertainty in locations far from the data points.

Fig. 3.4 and 3.5 show tracking performance of our approach and GP-MPC. We compare the tracking error of both methods, calculated as the distance from the robot position to the reference point, specified by the governor in our approach and by time parameterization of the path in GP-MPC: $\mathbf{p}^*(t) = \mathbf{r}(\min(t,10)/10)$, i.e., the GP-MPC method finishes the task in about 10 seconds, similar to the tracking time of our approach. Our controller is able to track the path more accurately than GP-MPC, illustrated qualitatively in Fig. 3.5 and quantitatively in Fig. 3.4 (middle). This can be explained by the higher predictions errors shown in Fig. 3.4 (left), which grow quickly after multiple time steps due to uncertainty propagation. Both safe tracking controller with learned Hamiltonian dynamics and the GP-MPC controller keep the hexarotor in the safe region, i.e., the distance to the obstacles is always positive in Fig. 3.4 (right).

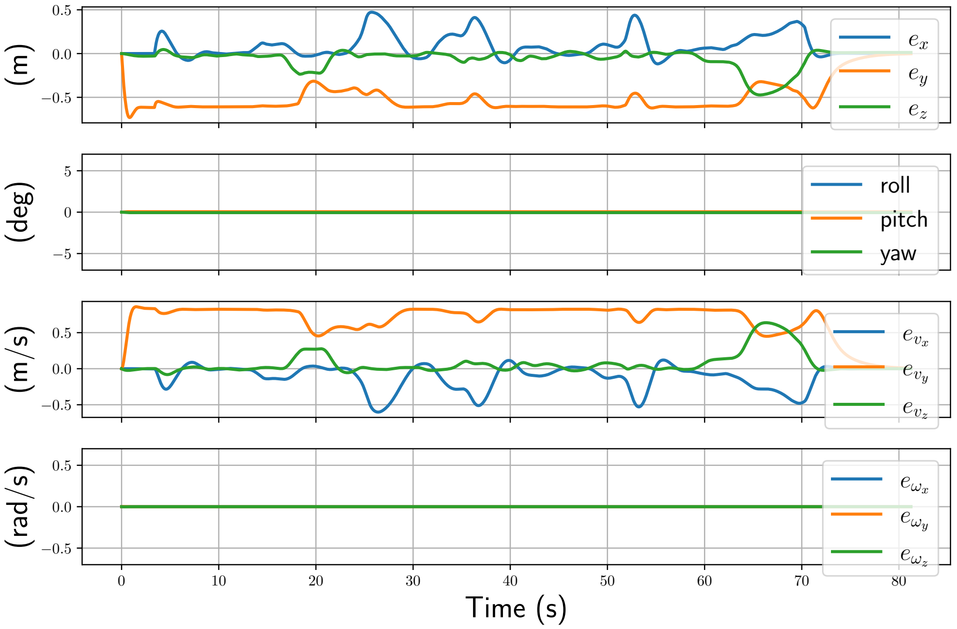


Figure 3.7. Tracking error of a hexarotor system (top to bottom): position, velocity, angle and angular velocity errors.

3.7.3 Evaluation of Robust Safe Tracking Control of a Learned 3D Fully-actuated Hexarotor Hamiltonian Model

This section evaluates our Hamiltonian dynamics learning and safe tracking control techniques using a simulated hexarotor UAV in a 3D environment. The task is to navigate from a start position to a goal in a cluttered warehouse environment without colliding with the obstacles $\mathscr{O}$. The same control gains are used for this 3D navigation task as in the previous section. A simulated LiDAR scanner provides point cloud measurements $\mathscr{P}(t)$ of the surface of the unsafe set $\mathscr{O}$, depending on the system pose at time t, with a maximum sensing range of $d_{max} = 30$ m. The distance from the governor $\mathbf{g}(t)$ to the unsafe set $\mathscr{O}$ is approximated via $\bar{d}(\mathbf{g}(t);\mathscr{O}) \approx \min_{\mathbf{y}\in\mathscr{P}(t)} \|\mathbf{g}(t) - \mathbf{y}\|$. The reference path $\mathbf{r}$ is pre-computed using an A* planner and tracked in about 80 seconds.

Fig. 3.6 shows the behavior of the closed-loop hexarotor system in the warehouse environment. The safety margin $\Delta E(\mathbf{x},\mathbf{x}^*)$ fluctuates during the tracking process but, as can be seen in Fig. 3.6, it never becomes negative. The augmented system $(\mathbf{x},\mathbf{g})$ is controlled adaptively,

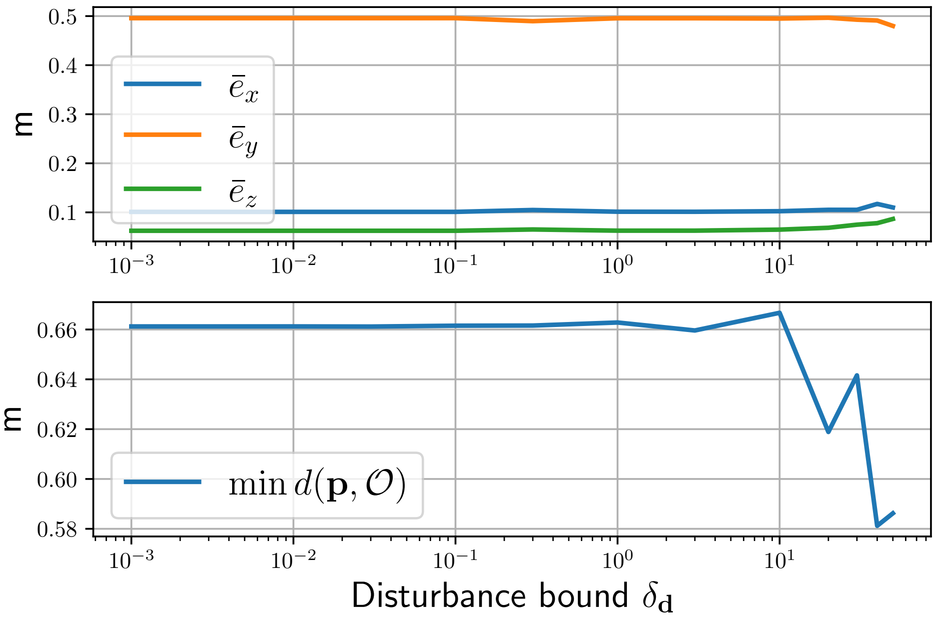


Figure 3.8. Tracking controller performance for hexarotor in warehouse simulation with the ground truth model subject to a disturbance $\mathbf{d}$ with different magnitudes: the average position tracking error (top) and the minimum distance to obstacle (bottom).

slowing down when the dynamic safety margin decreases (e.g., when the hexarotor is close to an obstacle or has large Lyapunov value $\mathscr{V}$) and speeding up otherwise (e.g., when the robot is far away from the obstacles or has small total energy $\mathscr{V}$). The simulations show that our control policy successfully drives the system from the start to the end of the reference path while avoiding sensed obstacles online, i.e., $d(\mathbf{p}, \mathscr{O})$ remains positive throughout the tracking task. Fig. 3.7 plots the tracking errors between the robot state $\mathbf{x}$ and the reference state $\mathbf{x}^*$ generated by the governor, showing that our controller tracks the path well. The tracking errors for the Euler angles and angular velocity, are close to 0. The position and linear velocity errors in the x and z directions are close to zero as well while the errors in y direction fluctuates around -0.5 m and 0.8 m/s, respectively, and converges to 0 at the end. This is expected as the robot stays behind the reference point, mostly in y direction, and converges to the end of the path.

To evaluate the robustness of our controller, we repeat the warehouse experiment using the ground-truth dynamics, subject to a artificially generated disturbances $\mathbf{d} \in \mathbb{R}^6$ with different upper bounds $\delta_{\mathbf{d}}$. Each component of the disturbance $\mathbf{d} \in \mathbb{R}^6$ is uniformly generated in $[-0.5\delta_{\mathbf{d}}, 0.5\delta_{\mathbf{d}}]$. If $\|\mathbf{d}\| > \delta_{\mathbf{d}}$, we normalize the disturbance as $\delta_{\mathbf{d}}\mathbf{d}/\|\mathbf{d}\|$. Our robust tracking controller successfully finishes the tracking task across a wide range of $\delta_{\mathbf{d}}$: $[0.001, 0.01, 0.1, 1, 10, 20, 30]$. Larger $\delta_{\mathbf{d}}$ are not reported due to violation of the positiveness requirement on ΔE. Fig. 3.8 shows the average position errors and the minimum distance to obstacle during the tracking task versus the disturbance upper bound $\delta_{\mathbf{d}}$. The average position tracking errors remain similar against $\delta_{\mathbf{d}}$. The minimum distance to obstacle $d(\mathbf{p}, \mathcal{O})$ is always positive, illustrating the safety guarantees of our controller. This number starts decreasing when $\delta_{\mathbf{d}} > 1$ as larger disturbances can suddenly move the robot towards the obstacles.

3.7.4 Evaluation of Robust Safe Tracking Control of a Learned 3D Quadrotor Hamiltonian Model

In this section, we repeat the task of safely navigating from a start position to a goal in the same cluttered warehouse environment in Sec. 3.7.3 with a quadrotor, whose model is learned from data as described in Sec. 3.7.3.7.1. As mentioned in Sec. 3.4, the control input in (3.18) would not introduce additional disturbance $\mathbf{d}_2$ when the matching condition (3.20) is satisfied. For quadrotor, a maximal-rank left annihilator of the ground-truth $\mathbf{B}(\mathfrak{q}) = \begin{bmatrix} \mathbf{0}_{4\times 2} & \mathbf{I}_{4\times 4} \end{bmatrix}^\top$ is $\mathbf{B}^\dagger(\mathfrak{q}) = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 4} \end{bmatrix}$. The matching condition (3.20) is satisfied if and only if the first two elements of $\mathbf{b}(\mathbf{x}, \mathbf{x}^*) = \begin{bmatrix} \mathbf{b}_{\mathbf{v}}^\top & \mathbf{b}_{\omega}^\top \end{bmatrix}^\top$, $\mathbf{b}_{\mathbf{v}} \in \mathbb{R}^3, \mathbf{b}_{\omega} \in \mathbb{R}^3$ in (3.19) equal to 0, i.e. the force component $\mathbf{b}_{\mathbf{v}}$ coincides with the z-axis of the body frame. As guaranteeing this condition is hard, we instead use the force component in the world frame $\mathbf{R}\mathbf{b}_{\mathbf{v}}$ and a desired yaw angle ψ^* to determine the desired rotation matrix, similar to [64]. The vector $\mathbf{R}\mathbf{b}_{\mathbf{v}}$ is set as the z-axis of the desired frame, i.e., the third column $\mathbf{b}_3^*$ of the rotation matrix $\mathbf{R}^*$, to minimize the disturbance $\mathbf{d}_2$ in (3.21) from the matching condition. We calculate the second column $\mathbf{b}_2^*$ by projecting the second column of the yaw's rotation matrix $\mathbf{b}_2^{\psi} = [-\cos\psi, \sin\psi, 0]$ onto the plane perpendicular

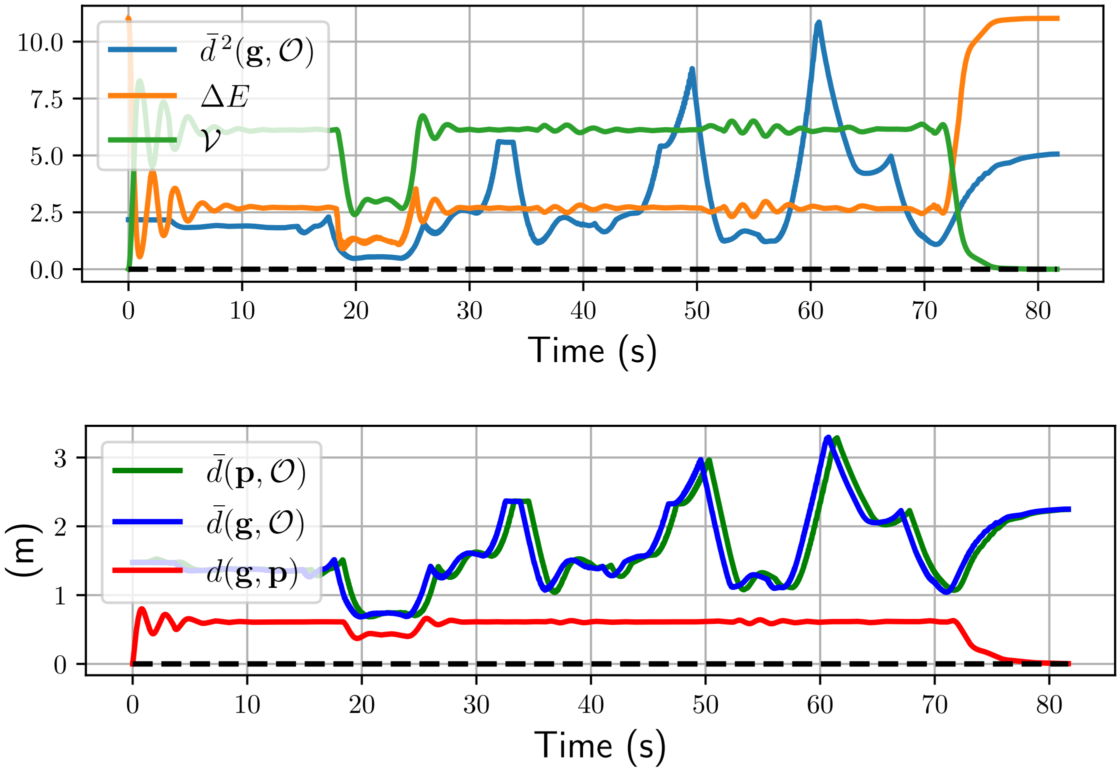


Figure 3.9. Safe navigation of quadrotor system (learned model) in a warehouse: the dynamic safety margin ΔE, the Lyapunov function $\mathscr{V}$ (top) and the distance to the obstacles $\bar{d}(\mathbf{p}(t),\mathscr{O})$ (bottom), indicating that the safety constraints are never violated.

to $\mathbf{b}_3^*$. We use the controller (3.18) with $\mathbf{R}^* = [\mathbf{b}_1^* \quad \mathbf{b}_2^* \quad \mathbf{b}_3^*]$ where:

$$\mathbf{b}_3^* = \frac{\mathbf{R}\mathbf{b}_\mathbf{v}}{\|\mathbf{R}\mathbf{b}_\mathbf{v}\|}, \mathbf{b}_1^* = \frac{\mathbf{b}_2^\psi \times \mathbf{b}_3^*}{\|\mathbf{b}_2^\psi \times \mathbf{b}_3^*\|}, \mathbf{b}_2^* = \mathbf{b}_3^* \times \mathbf{b}_1^*, \tag{3.51}$$

and $\hat{\omega}^* = \mathbf{R}^{*\top}\dot{\mathbf{R}}^*$ for our tracking task.

We successfully finish the task with the quadrotor while remaining safe for the entire experiment, as shown in Fig. 3.9, with similar behavior of the closed-loop quadrotor system in terms of the safety margin, Lyapunov function and distance to obstacle compared to Sec. 3.7.3. However, the orientation tracking error of quadrotor (Fig. 3.10) is larger than that of hexarotor, as expected since the quadrotor is underactuated.

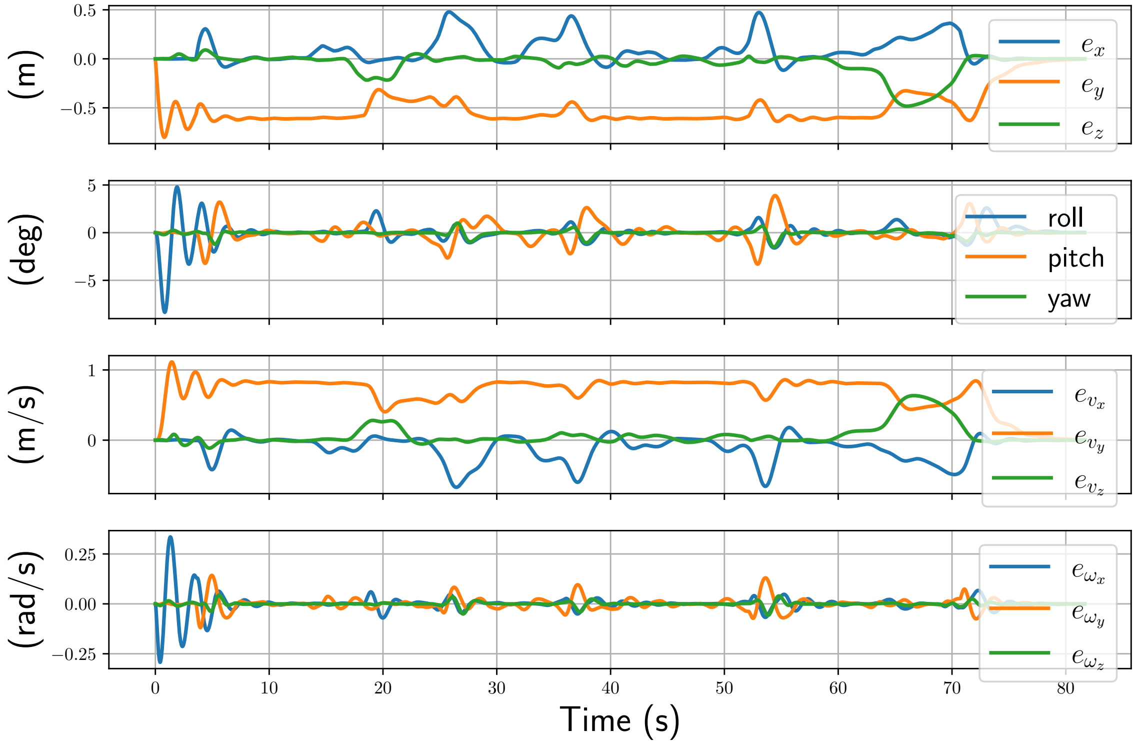


Figure 3.10. Tracking error of a quadrotor system (top to bottom): position, velocity, angle and angular velocity errors.

3.7.5 Evaluation of Our Approach against Unmodeled Noise

In this section, we verify the robustness of our controller against unmodeled noise on a simulated hexarotor by injecting high frequency noise (e.g., propeller vibration) into control inputs and simulating state estimation errors. In particular, a 4.8 kHz 6D sinusoidal signal with amplitude 5 is generated for high frequency noise. Meanwhile, state estimation errors in positions, Euler angles, linear and angular velocity are randomly generated with zero mean and standard deviation, chosen from [46] (position: 0.01 m, Euler angle: 0.01 degree, linear velocity: 0.02 m/s and angular velocity: 0.14 degree/s). We consider the task of stabilizing to a static governor, i.e. the governor is not moving, with the learned dynamics model: without any unmodeled noise (base), with high-frequency noise, and with state estimation error. Fig. 3.11 plots the Lyapunov function $\mathscr{V}$ and the safety margin ΔE over time. Our controller is not affected significantly from the high-frequency noise, potentially because the noise's effect is

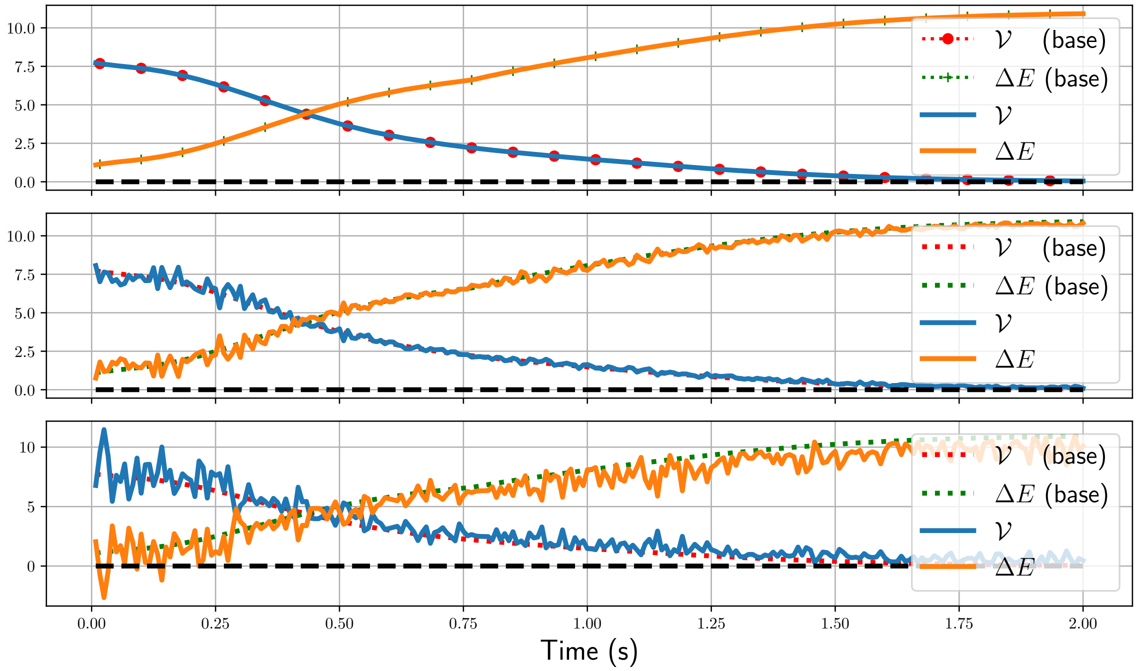


Figure 3.11. The Lyapunov function $\mathscr{V}$ and safety margin ΔE in the presence of: high-frequency noise (top), state estimation error with standard deviation from [46] (middle) and 3x larger (bottom), respectively.

canceled out due to its zero mean. Our controller is safe against the state estimation errors from [46], i.e. $\Delta E > 0$ over time, but fails to remains safe, i.e. $\Delta E < 0$ at some times, if we triple the noise deviation.

3.8 Summary

This chapter extends our tracking controller to Hamiltonian systems with learned dynamics. We employed a neural ODE network to learn translation-invariant Hamiltonian dynamics on the $SE(3)$ manifold from trajectory data. The Hamiltonian of the learned system was used to synthesize an energy-shaping controller and quantify its robustness to modeling errors. A reference governor was employed to guide the system along a desired reference path using the trade-off between system energy, disturbance bounds, and distance to obstacles to guarantee safe tracking. Our results demonstrate that encoding $SE(3)$ kinematics and Hamiltonian dynamics in the model learning process achieves more accurate prediction than Gaussian Process regression.

Utilizing the system energy in the control design offers a general approach for guaranteeing robustness and safety for physical systems and generalizes well to desired trajectories which are significantly different from the training data. Future work will focus on disturbance compensation and real experiments.

Acknowledgements

Chapter 3, is based on material from the following papers. First, “Safe autonomous navigation for systems with learned SE(3) Hamiltonian dynamics” authored by Zhichao Li*, Thai Duong* and Nikolay Atanasov, which appears in Learning for Dynamics and Control Conference (L4DC), 2022. The dissertation author was the co-primary investigator and author of this paper, (*equal contribution). Second, “Robust and Safe Autonomous Navigation for Systems with Learned SE(3) Hamiltonian Dynamics” by Zhichao Li, Thai Duong and Nikolay Atanasov, which appears in Open Journal of Control Systems, IEEE, 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Environment Aware Safe Tracking using Planning and Control Co-Design

4.1 Overview

In this chapter, we propose a new method that systematically puts planning, motion prediction and safety metric design together to achieve environment adaptive and safe navigation. This approach balances optimality in travel distance and safety with respect to passing clearance. A robot employing the proposed approach adapts its speed according to the sensed environment, being fast in wide open areas and slow in narrow passages. In our method, a directional distance measure, directional reachable set prediction and a clearance-based costmap are integrated to evaluate system risk accurately with respect to the local geometry of the surrounding environments. Using such risk estimation, a reference governor technique and a control barrier function are utilized to enable adaptive and safe path tracking in dynamic environments.

We consider a similar safe navigation problem as in Sec. 2.1. First, we show how a directional distance metric can be used to improve tracking performance for linear systems by exploiting the geometric features of the local environment. Then, we describe how to extend the use of a directional metric for a nonlinear system using ice-cream-shaped reachable set approximation. Furthermore, we extend our tracking control design to dynamic environments by injecting CBF-based time-varying constraints to handle moving obstacles. We validate our algorithm extensively both in simulation and challenging real-world environments. In summary, the focus of this chapter is an integrated planning and tracking control design, which achieves safe and *adaptive* navigation in *dynamic environments*.

4.2 Directional Metric and Motion Prediction

As mentioned in the introduction, for the safe tracking task, it is crucial to develop environment-aware and directional safety metrics. Measuring safety using a static Euclidean distance can also lead to poor performance due to the equal consideration of all directions, regardless of the direction of motion. In this section, we develop state-dependent directional metrics. In the beginning, we present a quadratic norm-based distance metric followed by the definition of a directional matrix which incorporates the robot's motion direction. Using a directional matrix, we develop two types of safety metrics based on output trajectory prediction and reachable set approximation. To facilitate further discussion, we introduce some notation.

Let $\mathbb{S}^n_{>0}$ and $\mathbb{S}^n_{\geq 0}$ denote the set of $n \times n$ symmetric positive definite and positive semi-definite matrices. Let $\succ$ and $\succeq$ denote the generalized inequalities associated with $\mathbb{S}^n_{>0}$ and $\mathbb{S}^n_{\geq 0}$. Denote the Euclidean (ℓ^2) norm by $\|\mathbf{x}\|$ and the quadratic norm induced by $\mathbf{Q} \in \mathbb{S}^n_{>0}$ by $\|\mathbf{x}\|_{\mathbf{Q}} := \sqrt{\mathbf{x}^T\mathbf{Q}\mathbf{x}}$. Let $\lambda_{\max}(\mathbf{Q})$ and $\lambda_{\min}(\mathbf{Q})$ be the maximum and minimum eigenvalues of $\mathbf{Q}$. Let $d_{\mathbf{Q}}(\mathbf{x}, \mathscr{A}) := \inf_{\mathbf{a}\in\mathscr{A}} \|\mathbf{x}-\mathbf{a}\|_{\mathbf{Q}}$ denote the quadratic norm distance from a point $\mathbf{x}$ to a set $\mathscr{A}$. Given $\mathbf{Q} \in \mathbb{S}^n_{>0}$ and scaling $\eta \geq 0$, denote the associated ellipsoid centered at $\mathbf{q} \in \mathbb{R}^n$ by $\mathscr{E}_{\mathbf{Q}}(\mathbf{q}, \eta) := \left\{\mathbf{x} \in \mathbb{R}^n \mid (\mathbf{x}-\mathbf{q})^T\mathbf{Q}(\mathbf{x}-\mathbf{q}) \leq \eta\right\}$.

4.2.1 State-Dependent Directional Metric

As mentioned in the introduction, measuring safety using a static Euclidean norm may lead to system performance suffering from the *corridor effect*. We propose a quadratic distance measure $\|\cdot\|_{\mathbf{Q}}$ that assigns priority to obstacles depending on the robot's moving direction. The level sets of $\|\cdot\|_{\mathbf{Q}}$ are ellipsoids $\mathscr{E}_{\mathbf{Q}}(\mathbf{0}, \eta)$ whose shape and orientation are determined by the matrix $\mathbf{Q}$. Our idea is to encode a desired directional preference in the distance metric via an appropriate choice of $\mathbf{Q}$. Consider the example in Fig. 4.1. A quadratic norm, well-aligned

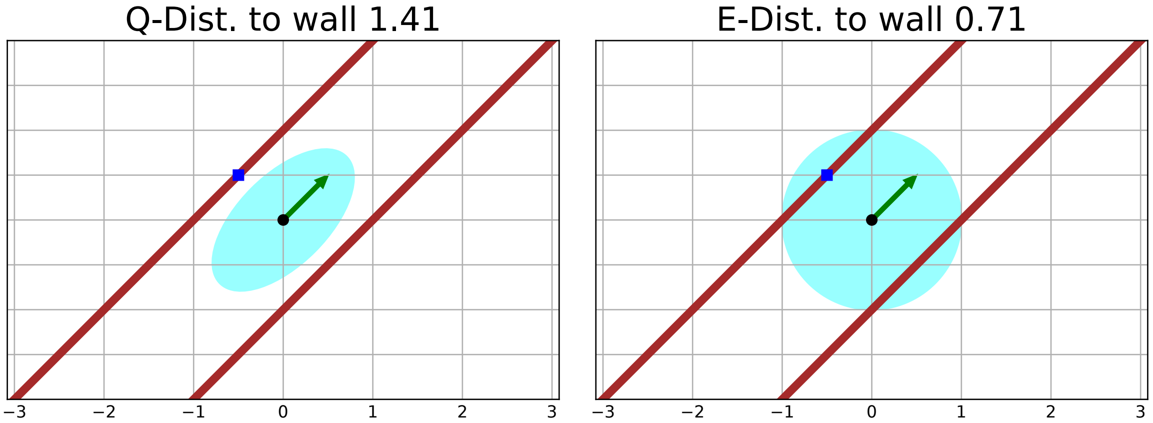


Figure 4.1. Robot (black dot) moving in direction $\mathbf{v} := \left[\sqrt{2}/2, \sqrt{2}/2\right]$ (green arrow) along a corridor. The distances, measured by a quadratic norm $\|\cdot\|_{\mathbf{Q}}$ (left) and Euclidean norm $\|\cdot\|$ (right), from the robot to the closest point (small blue square) on the wall (red line) are 1.41 and 0.71. The matrix $\mathbf{Q} = [[2.5 - 1.5], [-1.5, 2]]$ is defined as a *directional matrix* $\mathbf{Q}[\mathbf{v}]$.

with the local environment geometry, may provide a more accurate evaluation of safety than a static Euclidean norm. Based on this observation, we propose a general construction of a directional matrix $\mathbf{Q}[\mathbf{v}]$, in the direction of vector $\mathbf{v}$, that defines a state-dependent directional metric (SDDM).

Definition 8. A *directional matrix* associated with vector $\mathbf{v}$ and scalars $c_2 > c_1 > 0$ is defined as

$$\mathbf{Q}[\mathbf{v}] = \begin{cases} c_2\mathbf{I} + (c_1 - c_2)\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2}, & \text{if } \mathbf{v} \neq 0, \\ c_1\mathbf{I}, & \text{otherwise.} \end{cases} \tag{4.1}$$

The unit ellipsoid $\mathscr{E}_{\mathbf{Q}[\mathbf{v}]}(\mathbf{p}, 1)$ centered at $\mathbf{p}$ generated by a directional matrix $\mathbf{Q}[\mathbf{v}]$ is elongated in the direction of $\mathbf{v}$.

Lemma 3. *For any vector* $\mathbf{v}$*, the directional matrix* $\mathbf{Q}[\mathbf{v}]$ *is symmetric positive definite.*

Proof. Since $\mathbf{v}\mathbf{v}^T$ is symmetric, $\mathbf{Q}[\mathbf{v}]^T = \mathbf{Q}[\mathbf{v}]$. If $\mathbf{v} = \mathbf{0}$, $\mathbf{Q}[\mathbf{v}] = c_1\mathbf{I}$ is positive definite. If $\mathbf{v} \neq \mathbf{0}$

and $\mathbf{q}$ is arbitrary:

$$\begin{aligned}\mathbf{q}^T\mathbf{Q}[\mathbf{v}]\,\mathbf{q} &= c_2\mathbf{q}^T\mathbf{q} + (c_1 - c_2)\frac{(\mathbf{q}^T\mathbf{v})^2}{\|\mathbf{v}\|^2} \\ &\geq c_2\mathbf{q}^T\mathbf{q} + (c_1 - c_2)\frac{\|\mathbf{q}\|^2\|\mathbf{v}\|^2}{\|\mathbf{v}\|^2} = c_1\|\mathbf{q}\|^2,\end{aligned}$$

which follows from $c_2 > c_1$ and the Cauchy-Schwarz inequality. The proof is completed by noting that $c_1 > 0$. □

4.2.2 Trajectory Bounds Estimation for LTI System

Using a directional matrix, one can define an SDDM to adaptively evaluate the risk of surrounding obstacles. We will show how to use an SDDM to obtain bounds on the closed-loop trajectory of constrained output of LTI systems. Assume the system is stabilized by some state feedback control law. Without loss of generality, the closed-loop dynamics can be written as:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{A}}\bar{\mathbf{x}} \qquad \mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.2}$$

where $\bar{\mathbf{A}} := (\mathbf{A} - \mathbf{B}\mathbf{K})$ is Hurwitz. Any initial state $\bar{\mathbf{x}}_0 := \bar{\mathbf{x}}(t_0)$ will converge exponentially to the equilibrium point at the origin. The output $\mathbf{y}$ represents the constrained output of the system. For example, the position $\mathbf{p}$ of a mobile robot. We are interested in measuring the maximum deviation of $\mathbf{p}(t)$ for $t \geq 0$ from the origin using a directional measure determined by the orientation of initial state $\mathbf{p}_0 := \mathbf{p}(t_0)$ with respect to $\mathbf{0}$. Define an SDDM using the following directional matrix:

$$\mathbf{Q} := \mathbf{Q}[\mathbf{0} - \mathbf{p}_0] \in \mathbb{S}^n_{>0} \tag{4.3}$$

and choose output $\mathbf{y}(t) = \mathbf{Q}^{\frac{1}{2}}\mathbf{p}(t)$ so that $\mathbf{C} := \mathbf{Q}^{\frac{1}{2}}\mathbf{P}$, where $\mathbf{P} := [\mathbf{I}, \mathbf{0}]$ is the projection matrix from $\mathbf{x}$ to $\mathbf{p}$. Note that $\mathbf{y}(t)^T\mathbf{y}(t) = \mathbf{p}(t)^T\mathbf{Q}\mathbf{p}(t) = \|\mathbf{p}(t)\|^2_{\mathbf{Q}}$. Thus, measuring the maximum deviation of $\mathbf{p}(t)$ in the SDDM is equivalent to finding the output peak along the robot trajectory.

$$\eta(t_0) := \max_{t \geq t_0} \|\mathbf{p}(t)\|_{\mathbf{Q}}^2 = \max_{t \geq t_0} \|\mathbf{y}(t)\|^2 \tag{4.4}$$

We outline two approaches to solve this problem.

Exact solution

The output peak $\eta(t_0)$ can be computed exactly by comparing the values of $\|\mathbf{y}(t)\|^2$ at the boundary point $t = t_0$ and all critical points $\left\{t > t_0 \mid \frac{d}{dt}\|\mathbf{y}(t)\|^2 = 0\right\}$. Since the closed-loop system in (4.2) is linear time-invariant, $\mathbf{x}(t)$ can be obtained in closed form. Let $\bar{\mathbf{A}} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$ be the Jordan decomposition of $\bar{\mathbf{A}}$, where $\mathbf{J}$ is block diagonal. The critical points satisfy:

$$\begin{aligned} 0 &= \frac{d}{dt}\mathbf{y}(t)^T\mathbf{y}(t) = 2\mathbf{y}(t)^T\dot{\mathbf{y}}(t) \\ &= 2\left(\mathbf{P}\mathbf{V}e^{\mathbf{J}(t-t_0)}\mathbf{V}^{-1}\mathbf{x}_0\right)^T\mathbf{Q}\left(\mathbf{P}\mathbf{V}\mathbf{J}e^{\mathbf{J}(t-t_0)}\mathbf{V}^{-1}\mathbf{x}_0\right). \end{aligned} \tag{4.5}$$

In general, an exact solution may be hard to compute due to the complicated expression of $e^{\mathbf{J}t}$.

Approximate solution

When an exact solution to (4.4) is hard to obtain, we may instead compute a tight upper bound on $\eta(t_0)$. Given a $\mathbf{U} \in \mathbb{S}_{>0}^n$, let

$$\mathscr{E}_{inv} := \left\{\mathbf{w} \in \mathbb{R}^{n_s} \mid \mathbf{w}^T\mathbf{U}\mathbf{w} \leq 1\right\} \tag{4.6}$$

be an invariant ellipsoid for the robot dynamics (4.2), i.e., $\mathbf{x}(t) \in \mathscr{E}_{inv}$ for all $t \geq t_0$. Instead of finding the peak value of $\|\mathbf{y}(t)\|^2$ along the state trajectory, we can compute it over the invariant ellipsoid $\mathscr{E}_{inv}$. Since $\mathscr{E}_{inv}$ contains the system trajectory, we have for all $t \geq t_0$:

$$\|\mathbf{y}(t)\|^2 \leq \eta(t_0) \leq \max_{\mathbf{w} \in \mathscr{E}_{inv}} \mathbf{w}^T\mathbf{C}^T\mathbf{C}\mathbf{w}. \tag{4.7}$$

Obtaining the upper bound above is equivalent to solving the following SDP [16, Ch.6]:

$$\begin{aligned} \underset{\mathbf{U},\delta}{\text{minimize}} \quad & \delta \\ \text{subject to} \quad & \bar{\mathbf{A}}^T\mathbf{U}+\mathbf{U}\bar{\mathbf{A}} \preceq \mathbf{0},\ \ \bar{\mathbf{x}}_0^T\mathbf{U}\bar{\mathbf{x}}_0 \leq 1 \\ & \begin{bmatrix} \mathbf{U} & \mathbf{C}^T \\ \mathbf{C} & \delta\mathbf{I} \end{bmatrix} \succeq \mathbf{0}, \qquad \mathbf{U} \succ \mathbf{0}, \end{aligned} \tag{4.8}$$

where the constraints ensure that the ellipsoid is invariant.

Lemma 4. *For any initial condition* $\mathbf{x}_0$ *and associated constant directional matrix* $\mathbf{Q}$ *in* (4.3)*, the trajectory* $\mathbf{p}(t)$ *under system dynamics* (4.2) *admits a tight ellipsoid bound,* $\mathbf{p}(t) \in \mathscr{E}_{\mathbf{Q}}(\mathbf{0},\eta(t_0)) \subseteq \mathscr{E}_{\mathbf{Q}}(\mathbf{0},\delta(t_0))$*, for all* $t \geq t_0$*, where* $\eta(t_0)$ *is the solution to* (4.4) *and* $\delta(t_0)$ *is the solution to* (4.8)*.*

Proof. By definition, $\mathbf{p}(t) \in \mathscr{E}_{\mathbf{Q}}(\mathbf{0},\eta(t_0))$ is equivalent to $d^2_{\mathbf{Q}}(\mathbf{0},\mathbf{p}(t)) \leq \eta(t_0)$. Since $\delta(t_0) = \max_{\mathbf{w}\in\mathscr{E}_{inv}} \mathbf{w}^T\mathbf{C}^T\mathbf{C}\mathbf{w}$, inequality (4.7) yields $\delta(t_0) \geq \eta(t_0) \geq \|\mathbf{y}(t)\|^2 = \|\mathbf{p}(t)\|^2_{\mathbf{Q}} = d^2_{\mathbf{Q}}(\mathbf{0},\mathbf{p}(t))$. Hence, $\mathbf{p}(t)\in\mathscr{E}_{\mathbf{Q}}(\mathbf{0},\eta(t_0))\subseteq\mathscr{E}_{\mathbf{Q}}(\mathbf{0},\delta(t_0))$. □

We demonstrate that predicting the robot trajectory using our directional metric has some desirable properties for enforcing safety constraints. Fig. 4.2 compares trajectory bounds obtained from Lemma 4 for a closed-loop double integrator system using a Euclidean metric and an SDDM. It is clear that the ellipsoid bounds on the system trajectory are less conservative (smaller area/volume) than the spherical bounds at beginning.

Remark 3. Unlike a Lyapunov function, the ellipse $\mathscr{E}_{\mathbf{Q}(t)}(\mathbf{0},\eta(t))$ bounding the robot trajectory is not forward invariant. It can be shown that requiring invariance of directional ellipsoids ($\mathscr{E}_{\mathbf{Q}_{(t_1)}} \subset \mathscr{E}_{\mathbf{Q}_{(t_2)}}\ \forall t_2 \geq t_1$) would need infinite damping unless $\mathbf{Q} = k\mathbf{I}$ for some $k > 0$, causing the metric to lose directionality.

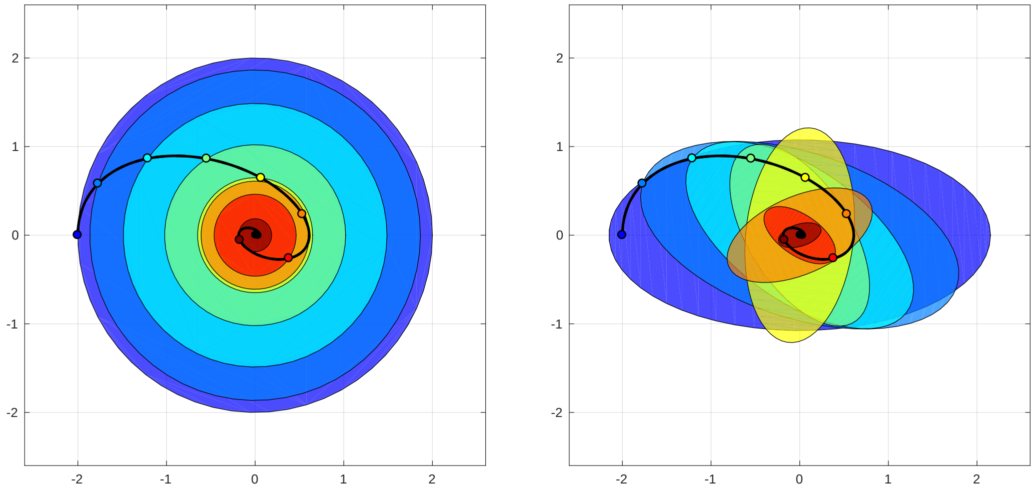


Figure 4.2. Trajectory bounds comparison between a Euclidean metric (left) and an SDDM (right). The governor is fixed at the origin while the robot's initial conditions are $\mathbf{p}_0 = (-2, 0)$ and $\dot{\mathbf{p}}_0 = (0, 2)$. The change of the trajectory bounds over time is illustrated via ellipsoids with different colors, starting from cold blue and converging towards warm red.

4.2.3 Trajectory Bounds Estimation for Unicycle-like Nonlinear Systems

In the last subsection, we discussed an accurate trajectory bound estimation for linear systems. This method can be extended to feedback-linearizable systems. However, for general nonlinear systems, obtaining trajectory bounds is quite challenging as mentioned in introductions. In this section, we focus on nonlinear systems with unicycle-like dynamics as shown below.

The state of the robot $\mathbf{x}$ consists of its position $\mathbf{p} = (x, y)$ and orientation θ, while its input is the linear velocity v and the angular velocity ω. Assuming there is no wheel slip, the kinematic model [26] describing the robot's motion is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega. \tag{4.9}$$

To obtain motion prediction, [50] proposes several outer approximation sets of the robot trajectory. We adopt the ice-cream cone set approximation that outperforms circular reachable set estimation by Lyapunov function. More importantly, this convex cone shape set can be computed efficiently, which makes it a great fit for our task.

When considering moving obstacles, allowing backward motion (rather than rotate-and-avoid) can take a robot to safer place sooner in certain cases. Hence, we extend the forward unicycle controller used in [50] to be bi-directional and use this controller in our experiments with moving obstacles.

Bi-directional Motion Control

Suppose the current robot state is $\mathbf{x} = (\mathbf{p}, \theta)$. Given any goal pose $\mathbf{x}^* = (\mathbf{p}^*, \theta^*)$, the control law $\mathbf{u}_{fm}(\mathbf{x}, \mathbf{x}^*) = (v(\mathbf{x}, \mathbf{x}^*), \omega(\mathbf{x}, \mathbf{x}^*))$ is defined as follows,

$$v = k_v \, e_v \qquad \omega = k_\omega \operatorname{atan}\left(e_v^\perp / e_v\right). \tag{4.10}$$

where $k_v > 0$ and $k_\omega > 0$ are control gains for linear and angular velocities and e_v and $e_v^\perp$ are defined as follows[1]:

$$e_v = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}^\top (\mathbf{p}^* - \mathbf{p}), \quad e_v^\perp = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}^\top (\mathbf{p}^* - \mathbf{p}). \tag{4.11}$$

One can show that the unicycle model (4.9) controlled by (4.10) is globally asymptotically stable [50, Lemma 1] in position, i.e., for any $\mathbf{p}^*$, the closed-loop unicycle trajectory $\mathbf{p}(t)$ satisfies $\lim_{t\to\infty} \mathbf{p}(t) = \mathbf{p}^*$.

Ice-cream Cone Shape Reachable Set Prediction

Given the robot's position and goal location, we can construct an ice-cream-cone shape outer approximation of its reachable set enclosing the future robot trajectory. This leads to a much tighter approximation of the reachable set compared to a circular or ellipsoidal motion

[1] To resolve the indeterminacy, set $\omega = 0$ when $\mathbf{p} = \mathbf{p}^*$.

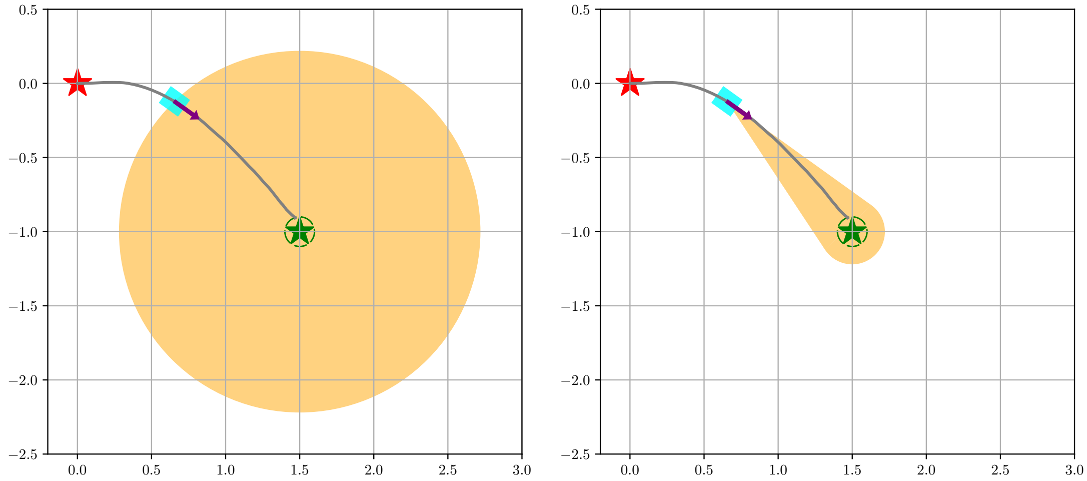


Figure 4.3. Cone shape reachable set prediction. The start and goal are depicted as red and green stars respectively with green circle being the goal region. The robot is scaled and shown as cyan rectangle with purple arrow showing its instantaneous moving direction. The reachable set approximations are shown in orange.

prediction from a quadratic Lyapunov function. An illustrative example given in Fig. 4.3, where the ball and cone approximations are depicted on left and right, respectively.

Proposition 3. *For any goal pose* $\mathbf{x}^* = (\mathbf{p}^*, \theta^*)$ *and any initial condition* $\mathbf{x}_0 = (\mathbf{p}_0, \theta_0)$*, the closed-loop trajectory* $\mathbf{x}(t) = (\mathbf{p}(t), \theta(t))$ *of the unicycle model* (4.9) *under control law* (4.10) *is positively invariant in the ice-cream-cone-shape motion prediction set* $\mathscr{M}(\mathbf{x}, \mathbf{p}^*)$ *defined as:*

$$\mathscr{M}(\mathbf{x}, \mathbf{p}^*) := \mathscr{C}\left(\mathbf{p}, \mathbf{p}^*, \|e_v^{\perp}\|\right) \tag{4.12}$$

where the bounded ice-cream cone $\mathscr{C}(\mathbf{p}, \mathbf{p}^*, \rho)$ *is defined as:*

$$\mathscr{C}(\mathbf{a}, \mathbf{b}, r) := \{\mathbf{a} + \alpha(\mathbf{z} - \mathbf{a}) \mid \alpha \in [0, 1], \mathbf{z} \in \mathscr{B}(\mathbf{b}, r)\}, \tag{4.13}$$

and $\mathscr{B}(\mathbf{c}, r) := \left\{\mathbf{z} \in \mathbb{R}^2 \mid \|\mathbf{z} - \mathbf{c}\| \leq r\right\}$ *is the closed ball centered at* $\mathbf{c} \in \mathbb{R}^2$ *with radius* $r \geq 0$.

Proof. The proof follows from Proposition 4 of [50]. □

To help later discussion, we name the control law in (4.10) a *cone controller*. Note that this controller only guarantees position convergence but orientation alignment is not enforced.

4.3 Dynamic Safety Margin using SDDM

We will show how to use an SDDM as adaptive risk assessment for collision avoidance during navigation. In Chapter 2, we defined dynamic safety margin (DSM) in Def. 3 that jointly considers the robot and governor status. We showed how to use a Parametric Barrier Function (PBF) as a DSM to regulate tracking speed via a reference governor. In this chapter, we will show how to construct a DSM from state-dependent direction metrics developed in the last section.

4.3.1 DSM from Ellipsoid Trajectory Bounds

Without loss of generality, we use a double integrator as running example for using SDDM as dynamic safety metric. It can be applied to any linear system satisfying regulator equations (2.4). Consider an acceleration-controlled robot, stabilized by a proportional-derivative (PD) controller:

$$\ddot{\mathbf{p}} = \mathbf{u} := -2k\mathbf{p} - \zeta\dot{\mathbf{p}}. \tag{4.14}$$

The closed-loop robot-governor system is:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -2k(\mathbf{p}-\mathbf{g}) - \zeta\mathbf{v} \\ -k_g(\mathbf{g}-\bar{\mathbf{g}}) \end{bmatrix}, \tag{4.15}$$

where the governor control policy is the same as in Section. 2.12.

Proposition 4. *Let* $(\mathbf{p}_0, \mathbf{y}_0, \mathbf{g}_0)$ *be any initial state for the robot-governor system in* (4.15) *with* $\mathbf{p}_0, \mathbf{g}_0 \in \mathscr{F}$*. Suppose that governor control law* $\dot{\mathbf{g}} \equiv \mathbf{0}$ *so that* $\mathbf{g}(t) \equiv \mathbf{g}_0$*. Let* $\mathbf{Q} := \mathbf{Q}[\mathbf{g}_0 - \mathbf{p}_0] \in \mathbb{S}^n_{>0}$ *be a constant directional matrix and suppose that the following safety condition is satisfied:*

$$\delta(t_0) \leq d^2_{\mathbf{Q}}(\mathbf{g}_0, \mathscr{O}), \tag{4.16}$$

where $\delta(t_0)$ is an upper bound for $\|\mathbf{p}(t) - \mathbf{g}_0\|^2_{\mathbf{Q}}$ obtained according to Lemma 4. Then, the robot-governor system is globally exponentially stable with respect to the equilibrium $(\mathbf{g}_0, \mathbf{0}, \mathbf{g}_0)$ and, moreover, the robot trajectory is collision free, i.e., $\mathbf{p}(t) \in \mathscr{F}$, for all $t \geq t_0$.

Proof. Since $(\mathbf{g}_0, \mathbf{0})$ is an equilibrium for $\dot{\bar{\mathbf{x}}} = \bar{\mathbf{A}}\bar{\mathbf{x}}$, by Lemma 4, $\mathbf{p}(t) \in \mathscr{E}_{\mathbf{Q}}(\mathbf{g}_0, \delta(t_0))$ for all $t \geq t_0$. From the safety condition in (4.16), $\mathbf{p}(t) \in \mathscr{E}_{\mathbf{Q}}(\mathbf{g}_0, \delta(t_0)) \subseteq \mathscr{E}_{\mathbf{Q}}(\mathbf{g}_0, d^2_{\mathbf{Q}}(\mathbf{g}_0, \mathscr{O})) \subseteq \mathscr{F}$ for all $t \geq t_0$. Stability and output regulation is ensured by Servo-regulator Theorem 1. □

Corollary 1. *The difference between the directional distance from the governor to the obstacle space and the ellipsoid bound is a dynamic safety metric, i.e., $\Delta E(\mathbf{x}, \mathbf{g}) = d^2_{\mathbf{Q}}(\mathbf{g}, \mathscr{O}) - \delta(\mathbf{x}, \mathbf{g})$ is a DSM.*

4.3.2 DSM from Reachable Set Approximation

By jointly considering bi-directional feed-forward information (from a costmap) and feedback information (from motion prediction), we define a safety metric using the distance between the motion prediction set and obstacle set defined by a costmap:

$$d(\mathscr{M}(\mathbf{x}, \mathbf{p}^*), \mathscr{O}), \tag{4.17}$$

where $\mathscr{M}$ is the ice-cream cone reachable set approximation shown in (4.12) and $\mathscr{O}$ is the obstacle space. In practice, one can design different cost curves and compute corresponding inflated obstacle space $\mathscr{O}^+$ to replace $\mathscr{O}$. As shown in [50] and [67], the distance between the reachable set (projection on the working space) and the obstacle space is a suitable DSM candidate.

Proposition 5. *For $\mathbf{g} \in Int(\mathscr{F})$, the $\mathbf{g}$-parameterized function*

$$\Delta E(\mathbf{x}, \mathbf{g}) = d(\mathscr{M}(\mathbf{x}, \mathbf{g}), \mathscr{O}), \tag{4.18}$$

is a dynamic safety margin for the closed-loop unicycle model with bi-directional control law (4.10).

Proof. The proof follows from [50, Proposition 9]. □

4.3.3 DSM for Adaptive Control Law

To achieve proper directional risk assessment, we embed the robot's state and surrounding environment into a risk metric design via SDDM. Furthermore, we develop an adaptive gain for longitudinal motion by computing the ratio of the directional safety metric and the regular one. With this gain, the robot with desired heading can go fast under straight motion in long corridor-like passages. First, let us define distance from a set $\mathscr{A}$ to another set $\mathscr{B}$ using quadratic norm as $d_{\mathbf{Q}}(\mathscr{A},\mathscr{B}) := \inf_{\mathbf{a}\in\mathscr{A},\mathbf{b}\in\mathscr{B}} \|\mathbf{a}-\mathbf{b}\|_{\mathbf{Q}}$. Then, we use a directional matrix $\mathbf{Q}[\mathbf{v}]$ to define a Q-norm and set the robot heading θ in the elongated direction, i.e., let $\mathbf{v} = [\cos\theta, \sin\theta]^\top$. Then, the SDDM-based boost gain can be defined as:

$$\beta := \begin{cases} \frac{d_{\mathbf{Q}[\mathbf{v}]}(\mathscr{M}(\mathbf{x},\mathbf{g}),\mathscr{O})}{d(\mathscr{M}(\mathbf{x},\mathbf{g}),\mathscr{O})} & \text{if } d(\mathscr{M}(\mathbf{x},\mathbf{g}),\mathscr{O}) > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{4.19}$$

to speed up the longitudinal motion (4.10), i.e., $v = \beta\, k_v\, e_v$.

Remark 4. The SDDM-based boost gain β is uniformly bounded. Following the proof of [50], one can verify that the shape of cone-shape motion prediction $\mathscr{M}(\mathbf{x},\mathbf{g})$ does not change if boost gain $\beta > 0$.

Combining the unicycle controller (4.10) with the adaptive boost gain β in (4.19), we have the closed-loop dynamics for unicycle model as:

$$\dot{x} = \beta\, v \cos\theta \tag{4.20a}$$

$$\dot{y} = \beta\, v \sin\theta \tag{4.20b}$$

$$\dot{\theta} = \omega, \tag{4.20c}$$

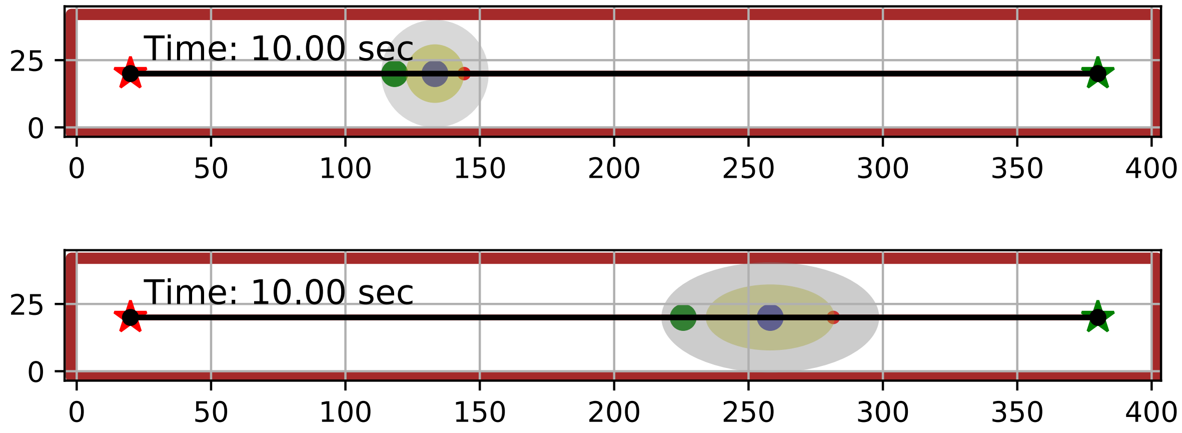


Figure 4.4. Comparison of baseline controller (top) and proposed controller (bottom) in a corridor simulation. A snapshot is shown at the same time instant for both controllers. The local energy zone (yellow) resulting from the proposed SDDM trajectory bounds fits the corridor environment well, leading to fast, yet safe, movement.

where v and ω are the same as shown in bi-directional motion control law (4.10).

4.3.4 Evaluation of Dynamic Safety Margin

We have introduce two new dynamic safety margins for linear and unicycle models. To demonstrate their effectiveness, we construct three different simulated environments with various complexity. We test a DSM backed by ellipsoid trajectory prediction in three cases: 1) *corridor environment* 2) *sparse environment with circular obstacles* 3) *unknown environment with cluttered arbitrary obstacles*. For DSM developed from reachable set prediction, we use a *C-shape environment* to show the advantage of the boost gain discussed in Sec. 4.3.3. Other experiments are delegated to the evaluation section (Sec. 4.6). For all simulations, the start and end of the path are indicated by a red and green star, respectively.

Corridor environment

We show that utilizing a directional metric in the control design alleviates the corridor effect discussed in introduction. We setup a simulation requiring a robot to navigate through a corridor (Fig. 4.4). We use the controller developed in [6] as our baseline controller, in which a

Lyapunov function,

$$V(\mathbf{x},\mathbf{g}) = k\|\mathbf{p}-\mathbf{g}\|^2 + \frac{1}{2}\|\mathbf{v}\|^2 \tag{4.21}$$

is used to estimate spherical reachable set. The results show that baseline controller, using a Lyapunov function with spherical level sets, suffers from the corridor effect while the proposed controller, making directional predictions about the system trajectory, does not.

Sparse environment with circular obstacles

This experiment compares the two controllers in a longer path-following task in an environment with circular obstacles. Snapshots illustrating how the two controllers judge distances to obstacles and define a local energy zone are shown in Fig. 4.6. It can be seen that the controller equipped with a directional sensing ability has a better understanding of the local environment geometry, leading to a larger, elongated local safe zone set. As a result, our controller does not need to slow down for low-risk lateral obstacles, leading to smoother and faster navigation. The directional bounds on the robot trajectory obtained analytically, according to eq. (4.5), and from the SDP in eq. (4.8) are compared in Fig. 4.5.

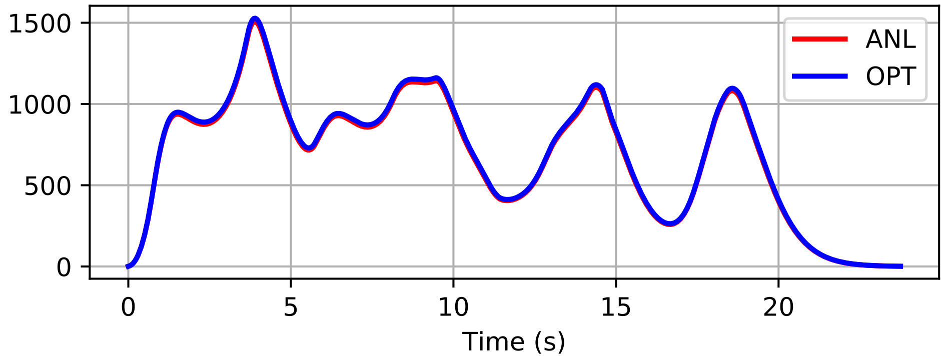


Figure 4.5. Output peak $\|\mathbf{p}(t)\|^2_{\mathbf{Q}(t)}$ from the trajectory followed in Fig. 4.6. The red curve is $\eta(t,\mathbf{Q}(t))$ obtained analytically from eq. (4.5). The blue curve is $\delta(t)$ computed from the SDP optimization in eq. (4.8). It is clear that $\delta(t)$ is an upper bound for $\eta(t,\mathbf{Q}(t))$, and the bound is tight at certain moments. Analytical bounds are used in simulation, while optimization bounds are computed for comparison purpose.

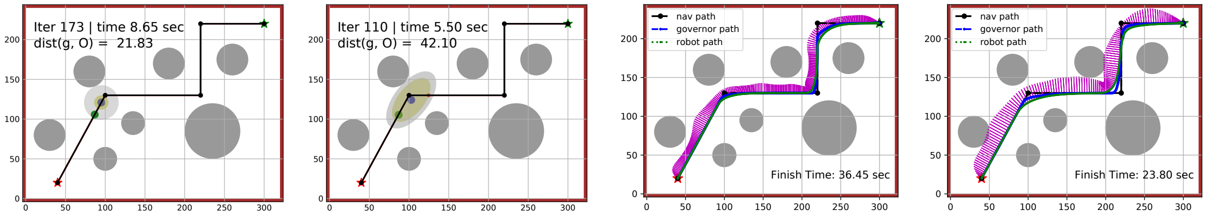


Figure 4.6. Simulation of the robot-governor system tracking a piecewise-linear path (black) in an environment with circular obstacles (dark gray circles). The two plots on the left show that the robot at around the same location behaves differently due to different distance measures. The controller using Euclidean distance is overly cautious with respect to lateral obstacles resulting in conservative motion. The system employing SDDM trajectory bounds has a larger local safe zone, which helps the robot turn fast and smoothly. The two plots on the right show the trajectories followed by the systems employing the two controllers. The velocity profiles are shown as magenta arrows perpendicular to the robot path. The controller based on SDDM trajectory bounds (rightmost) results in higher velocities compared to the controller using Euclidean ball invariant sets. Note that the path followed by the robot (green line) is also smoother, especially when turning, for the directional controller despite the higher velocity.

Unknown cluttered environment with arbitrary obstacles

This experiment demonstrates that our controller can work in a complex unknown cluttered environment relying only on local onboard measurements. The directional distance $d^2_{\mathbf{Q}(t)}(\mathbf{g}(t), \mathscr{O})$ from the governor to the obstacles is computed from the latest lidar scans. The path $\mathscr{P}$ is re-planned from the current governor position to the goal using an occupancy grid map constructed from the lidar scans over time, as illustrated in Fig. 4.7.

C-shape environment

To demonstrate the effectiveness of the directional boost gain (4.19), we create a C-shaped simulated environment using Gazebo as shown in Fig. 4.8. Two controllers are tested: left (boost gain disabled, i.e., $\beta \equiv 1$); right (boost gain enabled). All other configurations remain the same. The boost gain value can be read from the color bar on the side. From the plots, we see that the robot with SDDM boost speeds up faster in straight lines and remains at low speeds at turns, while the other one is overly cautious. As a result, the controller with boost gain finishes the task in $2/3$ of the time of the other one.

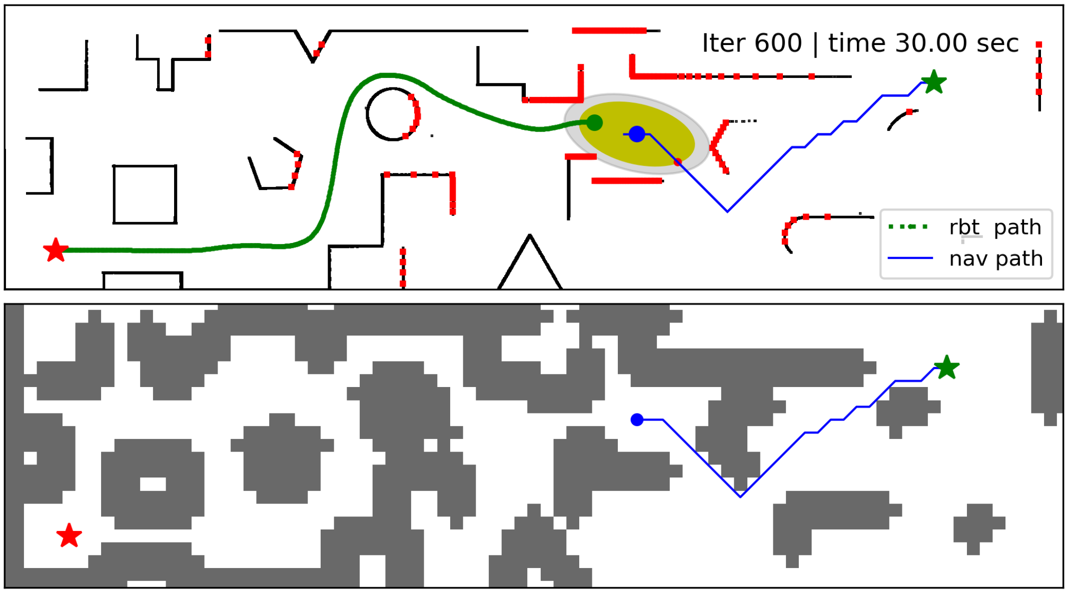


Figure 4.7. Snapshot of the robot-governor system navigating a cluttered maze. Streaming lidar scan measurements (red dots) are used to update an occupancy grid map of the unknown environment. An acceleration-controlled robot (green dot) follows a virtual governor (blue dot) whose motion is modulated based on the local energy zone (yellow ellipse) and the directional distance to obstacles (gray ellipse). Navigation path (blue line) is periodically replanned using an A* planner over the inflated occupancy grid map (bottom plot).

4.4 Safety Aware Planning

In previous sections, we introduced a new directional metric and developed dynamic safety margins. This lays a foundations for safe tracking tasks but a high-quality planner is indispensable for the reference-governor tracking methods to work properly. In previous chapters, we assumed a path is either given or can be computed easily from off-the-shelf planners. As discussed in the Introduction chapter, a simple A* algorithm may not be sufficient in safety-critical applications. In a complex environment, part of the shortest path usually stays close to the obstacle boundaries. Due to unmodelled dynamics and measurement noise, following shortest paths strictly might lead to collisions. On the other hand, aiming to find the maximum clearance path can lead to infeasibility or undesired long-distance paths. To resolve this, we design and construct a distance-field-based costmap as input for a search-based planning algorithm. Instead

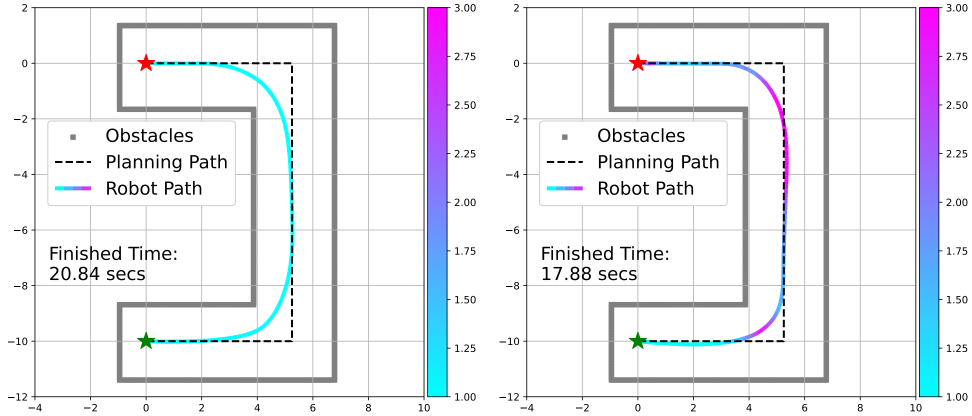


Figure 4.8. Simulation of robot-governor path following controller in C-shape environment. The adaptive boost gain is disable at left and enabled at right.

of searching for an optimal path (in terms of travel distance) over a binary occupancy grid map, using a costmap as input for the planner, we can tune the planner to generate optimal paths favoring safety while maintaining good performance in travel distance. It can generate paths from max-clearance path to shortest travel distance paths, leaving the practitioner the freedom to balance safety and efficiency. We will use a rectangularshaped *Jackal* robot as a running example for this section. As shown in Fig. 4.9, a Jackal robot is a differential-drive robot with four equivalent wheels, two on each side running at the same speed. The external dimensions are 508 × 430 × 250 mm. It weighs about 17 kg and can run at top speed 2.0 m/sec. The planning module consists of two parts, a costmap computation block and a standard geometric planner. We chose a standard A* planning algorithm as our 2D geometric planner. We start with the costmap design for the Jackal UGV.

4.4.1 Costmap Design

A costmap is a representation for the planning search space around a vehicle, it allows planned path to be pushed away from obstacle space [79]. Inspired by ROS costmap_2d package, we design our own costmap curve for the Jackal robot. The footprint of Jackal UGV is approximated as a rectangle with inscribed radius $r_{ins} = 0.215$ m and circumscribed radius $r_{cir} = 0.333$ m as shown in Fig. 4.10 (b). We set cost for real obstacle and inscribed obstacle (distance to robot

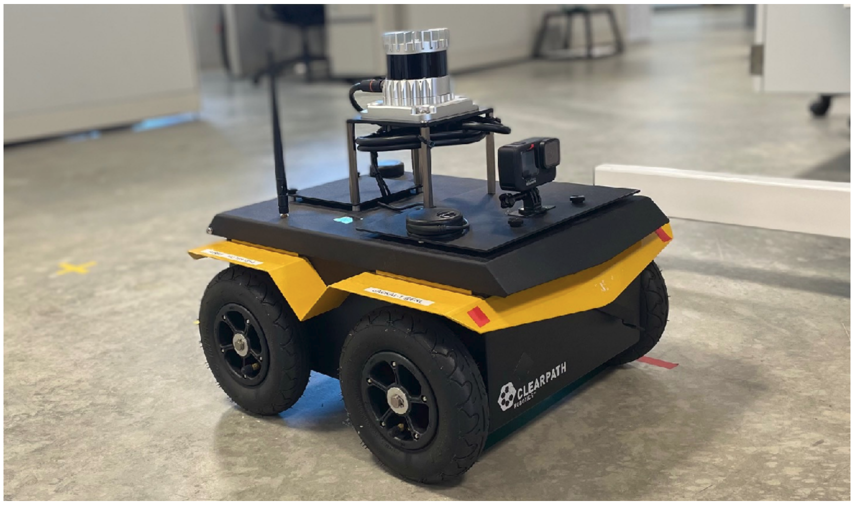


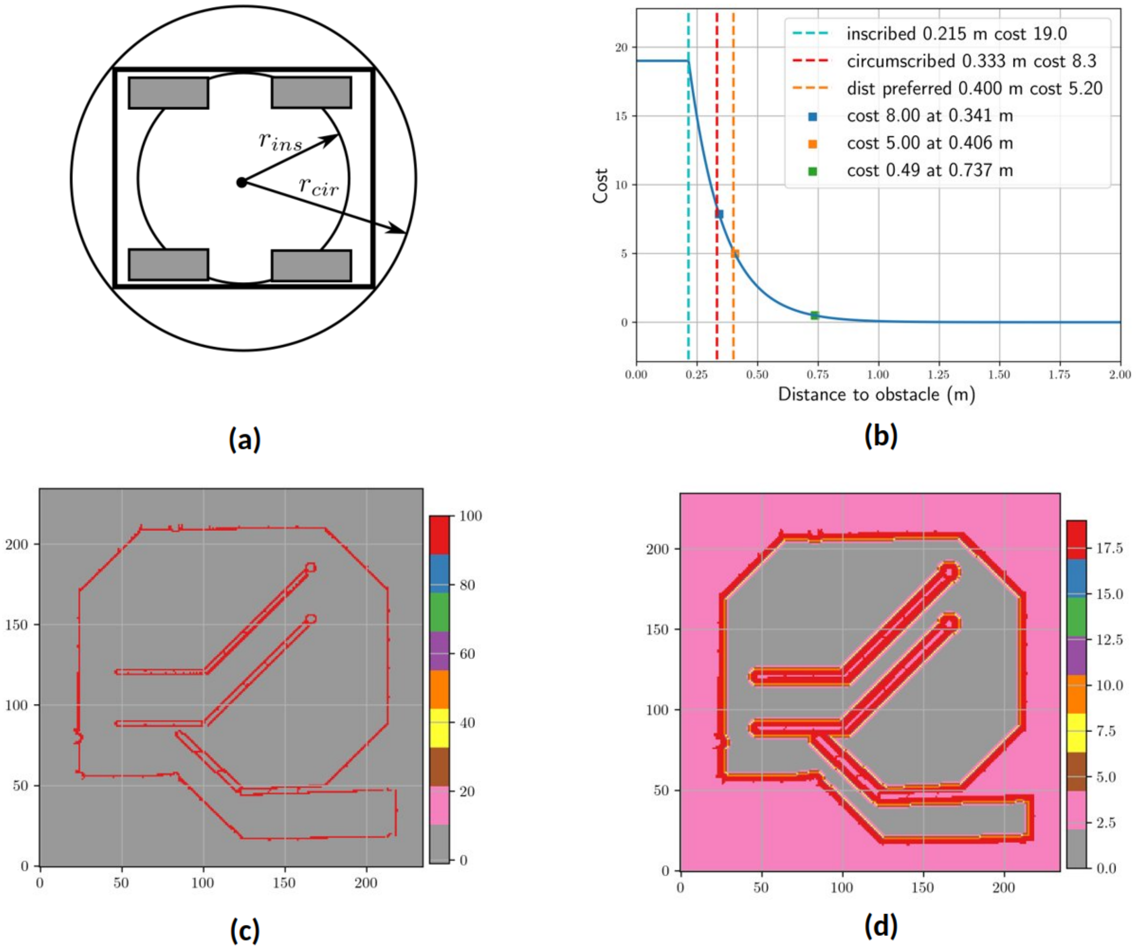
Figure 4.9. Jackal unmanned ground vehicle.

center less than inscribed radius) at 19 and the exponential decay rate chosen to be 7.0. To allow robot planning in unknown space yet still prefer known space, we set unknown cell cost at 3. The resulting cost curve is depicted in Fig. 4.10 (c). Using the cost curve, we can generate costmap from occupancy grid map message obtained from Hector SLAM. The hector map occupancy grid is first processed as binary image and then pass to distanceTransform function provided by OpenCV image processing library. Next, we multiply the distance map (in pixel) by grid map resolution, we get distance map over Euclidean distance metric. Finally, we assign cell cost according to its type and distance and get the costmap. The occupancy grid map and costmap are shown in Fig. 4.10 (c) and (d).

4.4.2 Planner Edge Cost Design

We choose Manhattan distance as heuristic and the cost between neighborhood cells on search graph is computed as follows:

$$\mathrm{cost}(\mathbf{p}_1,\mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\| + \mathrm{costmap}(\mathbf{p}_2), \tag{4.22}$$



(a)

(b)

(c)

(d)

Figure 4.10. The robot footprint is approximated as a rectangle with inscribed radius $r_{ins} = 0.215$ m and circumscribed radius $r_{cir} = 0.333$ m as depicted in (a). Cost curve design for Jackal robot is shown in (b) where lethal obstacle and inscribed cost is set to 19 with decay rate 7.0. The occupancy grid map of Jackal Race simulated world is shown in (c) with cell value being the probability (in percent) being occupied by obstacles. A corresponding costmap is shown in (d), in which cell value stands for traversing cost.

where $\mathbf{p}_1$ and $\mathbf{p}_2$ are the cell coordinates for parent and child cells and $\text{costmap}(\mathbf{p}_2)$ denotes the costmap value at $\mathbf{p}_2$.

Note that we can set different critical values when generating costmap from occupancy grid map. For example, considering circumscribed radius ($r_{cir} = 0.333$ m) and map resolution (0.1 m/cell), we can set the obstacle cost lower bound as 5 to ensure resulting path from planner can achieve at least 0.4 m clearance. This critical value 5 can be determined from the cost curve directly, see vertical orange line in Fig. 4.10 (b). A comprehensive example can be found in Figure 4.11 in next subsection, in which we use different costmap designs to generate various paths: from path with shortest travel distance, path with medium travel distance and clearance and path with maximum clearance.

4.4.3 Evaluation of Costmap Designs

From the previous discussion, we know that the costmap can affect the planner's performance and safety metric estimation. Using different cost curves, we can control the path clearance. In this section, we want to demonstrate how our algorithm can be tailored for different preferences.

Given map resolution and robot circumscribed radius, we choose 0.4 m as the minimum path clearance for all cost curve designs. A planning cutoff value and obstacle lower bound can be computed from the cost curve. For example, for a medium-clearance design whose cost curve is depicted in Fig. 4.10 (b), we set the planning cutoff value as 5 (corresponding to 0.406 m clearance) and the obstacle lower bound as 8 (corresponding to 0.333 m clearance). Similarly, we designed another two cost curves for a minimum clearance path (shortest path with desired 0.4 m clearance) and a maximum clearance path. The parameter values of these three costmap designs are listed in Table 4.1.

Using these different costmap designs, three corresponding paths are computed in Jackal race world and depicted as black dash line as shown in Fig. 4.11. For all cases, our algorithm can safely drive the robot towards goal. The resulting robot paths are plotted using colored

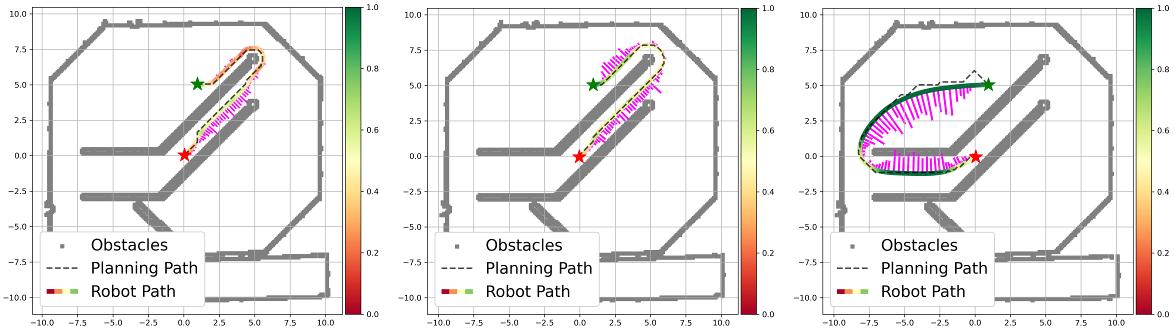


Figure 4.11. Jackal race world ROS simulation with three different cost curve designs.

line where robot to obstacle distance is embedded by gradient color. The value of color bar is saturated at 1 for visibility. Associated quantitative results are summarized at Table. 4.2.

Table 4.1. Three different costmap design parameters.

Costmap Design	Decay Rate	Planning Cutoff	Obstacle Lower Bound
minimum clearance	15.0	1	3
medium clearance	7.0	5	8
maximum clearance	1.0	15	17

Table 4.2. Simulation quantitative result of three costmap designs.

Costmap Design	Plan Path Length	Robot Traj. Length	Avg. Clearance	Min. Clearance
minimum clearance	15.10 m	15.66 m	0.45 m	0.13 m
medium clearance	16.19 m	15.63 m	0.63 m	0.34 m
maximum clearance	22.28 m	20.32 m	1.31 m	0.38 m

4.5 Safe Tracking via Reference Governor

In this section, we show how to use adaptive risk developed in Sec. 4.3 to construct two reference governor trackers for path following. We will first introduce the a reference governor path tracker design with necessary backgrounds. This tracker can help a robot navigate fast and efficiently in static environments. To handle moving obstacles, we change the governor dynamics allowing it to follow a safer goal around the given path but not necessarily on it. This extension can help the robot navigate safely in challenging situations with multiple obstacles. Controlled by our algorithm, the robot can progress to the goal efficiently while actively avoiding incoming obstacles in a natural way.

4.5.1 Governor Control Policy Design

Recall that a reference governor is a virtual first-order system defined as follows:

$$\dot{\mathbf{g}} = -k_g(\mathbf{g} - \mathbf{g}^*), \tag{4.23}$$

where $\mathbf{g} \in \mathbb{R}^2$ is the internal governor states and $\mathbf{g}^* \in \mathbb{R}^2$ is input signal to be determined. The governor state $\mathbf{g}$ is used to guide robot dynamics while following input signal $\mathbf{g}^*(t)$ served as intermediate goals along the reference (path $\mathscr{P}$). To utilize a reference governor for path following, a dynamic safety margin serves as an indicator of system safeness, wherein a larger value of it signifies increased safety. In the following of the paper, we will use DSM based on reachable set approximation as described in Sec. 4.3.2 for Jackal robot control. Considering robot footprint, we have the following DSM,

$$\Delta E(\mathbf{x}, \mathbf{g}) := d(\mathscr{M}(\mathbf{x}, \mathbf{g}), \mathscr{O}^+). \tag{4.24}$$

where $\mathscr{O}^+$ is the inflated obstacle space determined by robot footprint and costmap.

Next step is to find local goals used in governor dynamics, i.e., $\mathbf{g}^*$ in (4.23). Using safety margin ΔE, we can define a *local safe zone*, from which local goals will be computed. Then we can compute *local projected goal* as input signal $\bar{\mathbf{g}}$. The definitions of local safe zone and local projected goal are presented below with an illustrative plot in Fig. 4.12.

Definition 9. A *local safe zone* is a time-varying set, determined by the joint system-governor state $(\mathbf{x}, \mathbf{g})$, a dynamic safety margin $\Delta E(\mathbf{x}, \mathbf{g})$,

$$\mathscr{LS}(\mathbf{x}, \mathbf{g}) := \left\{\mathbf{q} \in \mathbb{R}^2 \mid \|\mathbf{q} - \mathbf{g}\|^2 \leq \Delta E(\mathbf{x}, \mathbf{g})\right\}. \tag{4.25}$$

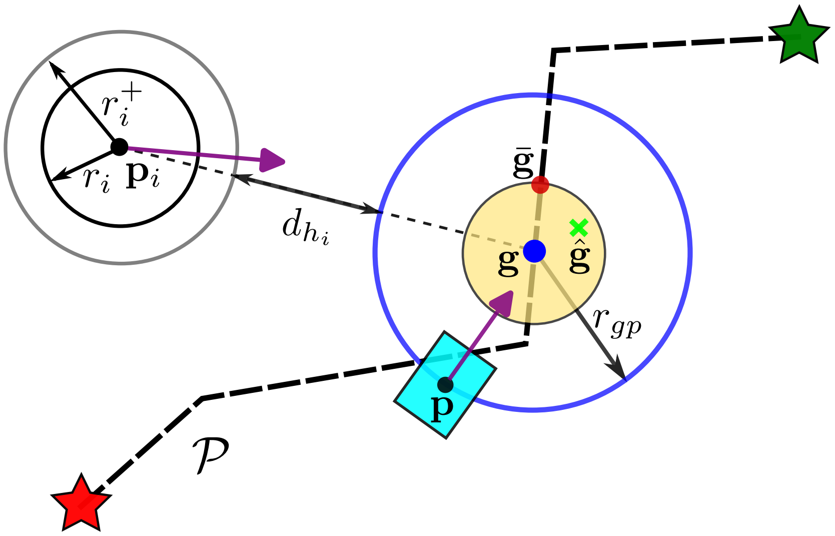


Figure 4.12. Geometric relationship of local safe zone $\mathscr{LS}$ (yellow ball), local projected goal $\bar{\mathbf{g}}$ (red dot) and optimized local goal $\hat{\mathbf{g}}$ (green dot).

Definition 10. A *local projected goal* at system-governor state $(\mathbf{x},\mathbf{g})$ is a point $\bar{\mathbf{g}} \in \mathscr{LS}(\mathbf{x},\mathbf{g})$ that is furthest along the reference path $\mathscr{P}$:

$$\bar{\mathbf{g}} = \mathscr{P}(\bar{\sigma}), \;\; \bar{\sigma} = \underset{\sigma\in[0,1]}{\operatorname{argmax}} \left\{\sigma \mid \mathscr{P}(\sigma) \in \mathscr{LS}(\mathbf{x},\mathbf{g})\right\}. \tag{4.26}$$

Until now we have completed the first reference-governor-based path tracker. Combining a unicycle controller (4.10) and the governor dynamics (4.23) with a time-varying input signal $\mathbf{g}^*(t) = \hat{\mathbf{g}}$ in the governor dynamics, we summarize the robot-governor dynamics as follows:

$$\dot{x} = \beta\, v \cos\theta \tag{4.27a}$$

$$\dot{y} = \beta\, v \sin\theta \tag{4.27b}$$

$$\dot{\theta} = \omega \tag{4.27c}$$

$$\dot{\mathbf{g}} = -k_g(\mathbf{g} - \bar{\mathbf{g}}), \tag{4.27d}$$

where β is the SDDM boost gain in (4.19) and v and ω as shown in (4.10).

Theorem 6. *Given a reference path $\mathscr{P}$, consider the closed-loop system in* (4.27)*. Suppose that* *the initial state* $(\mathbf{x}_0, \mathbf{g}_0)$ *satisfies:*

$$\Delta E(\mathbf{x}_0, \mathbf{g}_0) > 0, \quad \mathbf{g}_0 = \mathbf{p}(0) \in \mathscr{F}, \tag{4.28}$$

where $\Delta E(\mathbf{x}, \mathbf{g}) = d(\mathscr{M}(\mathbf{x}, \mathbf{g}), \mathscr{O}^+)$. *Then, the robot converges to goal region close to* $\mathscr{P}(1)$ *without collision, i.e.,* $\mathbf{p}(t) \in \mathscr{F}$, $\forall t \geq t_0$.

Proof. Note that, in this chapter we adopt new dynamic safety from reachable set approximation. The proofs follows from Theorem 2 and Proposition 5. □

The reference governor tracker can now guide the robot to navigate in a static unknown environment safely and efficiently. When a moving obstacle approaches to the robot ($\Delta E = 0$), robot will stop in place and will resume its motion only when the moving obstacles leaves ($\Delta E(t) > 0$). However, the non-at-fault stop-and-go strategy may not be efficient enough in certain applications where robots are required to navigate through crowds. To overcome this, we extend our algorithm in the following section.

4.5.2 Governor Control Policy Extension for Moving Obstacles

Merely adhering to the planned path is insufficient for promptly avoiding approaching obstacles. An intelligent algorithm should grant the robot the ability to temporarily deviate from the predetermined path and execute emergency maneuvers when deemed necessary. To acquire such capability, we not only need to bi-directional low-level motion control, but the leading signal shall also be flexible to provide proper guidance. The unicycle controller (4.10) shown in 4.2.3 already allows bi-direction movement. To satisfy the second requirement, we propose an optimization-based method to modify the input signal ($\bar{\mathbf{g}}$) in governor dynamics (4.23), allowing the governor to purse local goals that deviate from given path. So, when moving obstacles endanger robot's progression, it can take proper avoidance maneuver in a defensive fashion.

To ease the discussion, let us suppose there are k moving obstacles with circular shape centered at location $\{\mathbf{p}_i\}_{i=1}^k$ with radius $\{r_i\}_{i=1}^k$. All moving obstacles behave like an independent, linear, time-invariant system satisfying the following dynamics:

$$\dot{\mathbf{p}}_{\mathbf{i}} = \mathbf{v}_{\mathbf{i}}, \tag{4.29}$$

where $\mathbf{v}_i$ is the velocity of agent i. For simplicity, we assume the position and velocity of moving obstacles are known to the robot. In practice, this information can be obtained from the front-end perception module. Inspired by safety control techniques based on control barrier functions (CBFs) [2], we formulate a convex optimization problem to shift the governor input reference signal $\bar{\mathbf{g}}$ in a minimally invasive way to avoid incoming moving agents.

CBF construction for moving obstacle.

Let $r_i^+ = r_i + r_{cir}$ be the inflated radius of moving obstacle i, define the gap between two balls $\mathscr{B}(\mathbf{p}_i, r_i^+)$ and $\mathscr{B}(\mathbf{g}, \|\mathbf{p} - \mathbf{g}\|)$ as:

$$d_{hi} := \|\mathbf{g} - \mathbf{p}_i\| - \|\mathbf{g} - \mathbf{p}\| - r_i^+ \geq 0.$$

The gap between these two balls can be visualized in Fig. 4.12. One can prove [50, Prop.8] that the cone motion prediction set $\mathscr{M}(\mathbf{x}, \mathbf{g})$ is a subset of the ball $\mathscr{B}(\mathbf{g}, \|\mathbf{g} - \mathbf{p}\|)$, i.e.,

$$\mathscr{M}(\mathbf{x}, \mathbf{g}) \subset \mathscr{B}(\mathbf{g}, \|\mathbf{g} - \mathbf{p}\|). \tag{4.30}$$

Hence, the gap d_{hi} is a DSM between moving obstacle i and the robot. Therefore, we can construct a CBF for agent i from d_{hi}:

$$h_i(\mathbf{x}, \mathbf{g}, \mathbf{p}_i) = \|\mathbf{g} - \mathbf{p}_i\|^2 - \left(r_i^+ + \|\mathbf{g} - \mathbf{p}\|\right)^2. \tag{4.31}$$

To enforce collision avoidance, the CBF constraint becomes:

$$\dot{h}_i(\mathbf{x},\mathbf{g},\mathbf{p}_i) \geq -\alpha_i(h_i(\mathbf{x},\mathbf{g},\mathbf{p}_i)), \tag{4.32}$$

where $\alpha_i(\cdot)$ is a class-$\mathscr{K}$ function to be designed. Expanding out the expression of $\dot{h}$, we have:

$$\dot{h}_i(\mathbf{x},\mathbf{g},\mathbf{p}_i) = \frac{\partial h_i}{\partial \mathbf{g}}\mathbf{u_g} + \frac{\partial h_i}{\partial \mathbf{p}}\mathbf{u} + \frac{\partial h_i}{\partial \mathbf{p}_i}\mathbf{v}_i, \tag{4.33}$$

where $\mathbf{u} = [k_v e_v \beta \cos\theta, k_v e_v \beta \sin\theta]^\top$ is the control signal of robot and $\mathbf{u_g} \in \mathbb{R}^2$ is the governor control to be determined.

Local projected goal modification

With the above CBF for moving obstacles, we can push the local projected goal $\bar{\mathbf{g}}$ away from the path $\mathscr{P}$ temporarily if the nearest obstacle is getting too close to the robot. Let $h^* := \min_i h_i$. We formulate an optimization problem to determine the governor control input as follows:

$$\min_{\mathbf{u_g}\in\mathbb{R}^2,\ \hat{\mathbf{g}}\in\mathbb{R}^2} \|\hat{\mathbf{g}} - \bar{\mathbf{g}}\|^2 \tag{4.34a}$$

$$\text{subject to} \quad \dot{h}^* \geq -\alpha_i(h^*) \tag{4.34b}$$

$$\mathbf{u}_g = -k_g(\mathbf{g} - \hat{\mathbf{g}}) \tag{4.34c}$$

$$\hat{\mathbf{g}} \in \mathscr{L}\mathscr{S}(\mathbf{x},\mathbf{g}). \tag{4.34d}$$

This optimization problem tries to search a modified local projected goal $\hat{\mathbf{g}}$ within local safe zone $\mathscr{L}\mathscr{S}(\mathbf{x},\mathbf{g})$ that is close to the original one $\bar{\mathbf{g}}$. When a moving obstacle endangers the robot operation, the optimization will push the projected goal away from the path $\mathscr{P}$ to reduce the collision risk. An illustrative example is depicted in Fig. 4.12, where the optimized projected goal $\hat{\mathbf{g}}$ (green dot) is found in the vicinity of local projected goal $\bar{\mathbf{g}}$ (red dot) to avoid nearest incoming obstacle.

It is not hard to verify that this optimization problem is a convex program. By plugging (4.25) and (4.33), we can simplify the above optimization (4.34) as:

$$\min_{\hat{\mathbf{g}}\in\mathbb{R}^2} \|\hat{\mathbf{g}}-\bar{\mathbf{g}}\|^2 \tag{4.35a}$$

$$\text{subject to}\quad \mathbf{a}_h^\top \hat{\mathbf{g}} \geq b_h \tag{4.35b}$$

$$\|\hat{\mathbf{g}}-\mathbf{g}\|^2 \leq d(\mathcal{M},\mathcal{O}^+), \tag{4.35c}$$

where $\mathbf{a}_h = k_g \left(\frac{\partial h^*}{\partial \mathbf{g}}\right)^\top$ and $b_h = k_g \frac{\partial h^*}{\partial \mathbf{g}}\mathbf{g} - \frac{\partial h^*}{\partial \mathbf{p}}\mathbf{u} - \frac{\partial h^*}{\partial \mathbf{p}^*}\mathbf{v}^* - \alpha(h^*)$. The first constraint is linear in $\hat{\mathbf{g}}$ and the second one is a convex quadratic constraint in $\hat{\mathbf{g}}$ since $d(\mathcal{M},\mathcal{O}^+)$ is known. Hence, the optimization problem is a convex quadratically-constrained quadratic program (QCQP) and can be solved efficiently by a QCQP solver.

Once $\hat{\mathbf{g}}$ is obtained, replacing $\bar{\mathbf{g}}$ with the modified projected goal $\hat{\mathbf{g}}$ in (4.27d) leads to the governor dynamics:

$$\dot{\mathbf{g}} = -k_g(\mathbf{g}-\hat{\mathbf{g}}). \tag{4.36}$$

The effectiveness of this new active robot-governor control policy is verified in both simulation and hardware experiments in the next section. For easy reference, we call this optimization-based extension described in (4.34) and (4.36) **active governor control policy**.

In static environments, the proposed algorithm is provably safe as shown in Theorem 2 in Chapter 2. When moving obstacles are present, ensuring safety become much more involved because it depends on the behavior of the obstacles. There are no universally accepted criteria for safety among multiple agents in all situations. Sometimes, the problem is simply infeasible since many situations can result in unavoidable collisions, such as when a moving obstacle is adversarial and the environment and nonholonomic constraints restricts the motion of the robot from taking an avoidance maneuver. Due to this complexity, our formulation for

we do not provide a theoretical only verify the effective of our algorithm in hardware experiments. A rigorous proof is left as future work.

4.6 Evaluation

In this section, we will evaluate the performance of our path-following controller. We will with an overview of system design. Then we will test the performance of our controller using both simulations and hardware demonstrations. First, we describe our system design. In specific, we will introduce hardware platform including computation, sensing and communication module. Then, we will present our software architecture to enable safe navigation in unknown environments.

4.6.1 Hardware Overview.

We chose to use *ClearPath* Jackal Unmanned Ground Vehicle (UGV) as our experimental platform. As shown in Fig. 4.9, Jackal is a small field robotics research platform with onboard computer and customizable sensor configuration such as GPS, IMU and LiDAR. It is Robot Operating System (ROS) compatible, so we can quickly test and debug your algorithm in simulation and deploy it to hardware.

Mobile platform. Jackal UGV is a differential drive robot with four equivalent wheels, two on each side running at same speed. The external dimensions are $508 \times 430 \times 250$ mm. It weighs about 17 kg and can run at top speed 2.0 m/sec. For basic usage, it can run about 2-4 hours.

Computation, sensing and communication. Our custom Jackal is equipped with an Intel i7-9700TE CPU with 32GB RAM, an Ouster OS1-32 LiDAR and a UM7 9-axis IMU. This supplies sufficient computation power and sensing capability for autonomous navigation tasks. The UGV can be manually controlled by a Bluetooth joystick or remotely accessed by onboard Wi-Fi. We use the joystick to trigger our problem at the beginning and use it as an emergency stop controller. A local network is setup via a standard router, so we can use visualization tools that come with ROS development kit to monitor running status of our algorithm and save ROS

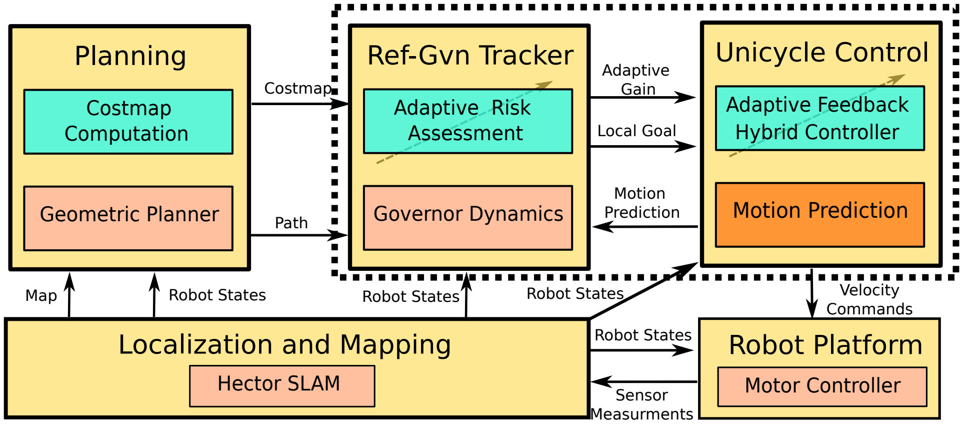


Figure 4.13. System architecture overview of environment aware safe tracking control.

bag for replay and debugging.

4.6.2 Software Architecture

The software components can be grouped into three categories: Localization and Mapping, Planning and Control. We use Robot Operation System (ROS) as the framework for all high-level tasks running on the robot. Figure 4.13 is an overview of our system architecture. On the top half, we have the planning block at left and the control modules (in dash rectangular), which are the focus of this paper. We use the out-of-box low-level velocity controller provided by *Clearpath Robotics* which takes linear and angular velocity command and transforms them to motor input signals. In the rest of this section, we will describe the localization and mapping package we used and briefly summarize our design in the planning and control.

Localization and mapping. For localization and mapping task, we use an open-source LiDAR-based 2D SLAM algorithm called Hector SLAM [57]. This algorithm takes 2D laser scan as inputs and outputs robot's poses (position and orientation) in an online constructed occupancy grid map. We use an ROS package named `pointcloud_to_laserscan` to convert 3D pointcloud from Ouster LiDAR to 2D laser scan.

Planning. The planning module consists of two parts, a costmap computation block and a standard geometric planner. As described in Sec. 4.4, we chose searched-based planning algorithm named A* as our 2D geometric planner. We create a distance-field based costmap as input for the planner. The customizable costmap can be easily tuned to generate optimal paths from max-clearance path to shortest travel distance path, leaving practitioner the freedom to balance safety and efficiency.

Control. The control system contains three parts: path tracking, unicycle velocity control and motor speed regulation. Motor control is handled by built-in embedded software from Clearpath. The *unicycle control* design including a bi-directional controller and an accurate motion prediction are described in Sec. 4.3.3 and Sec. 4.3.2.

To bridge the gap between a geometric path planner and unicycle controller, we develop *environmental-aware reference-governor tracker* in Section 4.5 that continuously generates proper local goals along the path and sends them to the controller. This tracker estimates running risk by considering local geometric information and system dynamics at the same time. In addition, it also updates an adaptive gain to speed up tracking process when it is safe to do so. Using this tracker, the robot can track the path safely and fast in challenging environments.

4.6.3 Experiment Setup and Parameters

In this section, all experiments share the same design parameters and control gains unless explicitly pointed out. For localization and mapping, the hector SLAM packages provide robot pose estimation at about 20 Hz with 0.1 m positional and 0.1 rad angular accuracy. The map resolution is chosen at 0.1 m. The control gains and directional matrix parameters are summarized in Table 4.3.

Unless pointed out explicitly, all experiments share the same control parameters in this table and the class-$\mathscr{K}$ function for moving obstacle experiment is:

$$\alpha_i(h_i) = \gamma_i h_i^2. \tag{4.37}$$

For all simulated experiments, we plot a 2D planning path as a black dash line with

Table 4.3. Parameter table of control gains.

Parameter	k_g	k_v	k_ω	γ_i	c_1	c_2
Value	2.0	0.5	1.5	0.2	1	9

start and goal denoted by a red and green star, respectively. The velocity profiles are shown as magenta arrows perpendicular to the robot path.

4.6.4 Baseline Comparison

In this section, we compare our work with a recent work called EVA-Planner [84], in which an environmental-aware adaptive planner is developed using MPC techniques with novel SDF gradient based directional safety metric. Two experiments have been done to compare these two methods with emphasis on difference on safety and adaptivity, respectively. The EVA planner solves the planning and control problem via a two-layer hierarchy MPC, all constraints are encoded as soft cost and solved by a gradient-based numerical solver (NLopt[2] + L-BFGS algorithm). It is worth pointing out that the EVA planner tries to solve are constrained nonlinear non-convex optimization problems. Therefore, the optimality and safety constraints satisfactions are not guaranteed. We tried our best to fine-tune it.

Safety ten point test. In the first experiment, we randomly sampled ten feasible points from Jackal race world as shown in Fig. 4.11. All goals are at integer grid intersections, markers positions are shifted slightly for visualization. From the plot, we can see that our algorithm can reach all ten goals safely, while EVA planner failed to reach one goal at $(6,-8)$. In our implementation, the EVA planner failed to find global optimal solution of the high MPCC optimization problem and outputs trajectory go through obstacles. Other paper find out that this EVA planner performance degrades (collision rate gets higher) when robot operates in environment with high obstacle density [71].

Adaptivity maze test. In the second experiment, we create a challenging maze-like environment to test the adaptivity of these two methods. The robot is asked to go from the start

[2] http://github.com/stevengj/nlopt

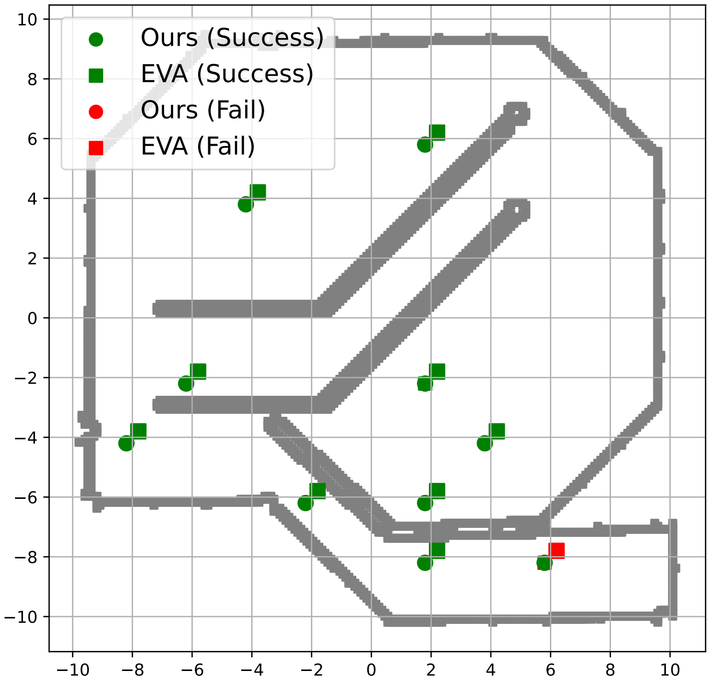


Figure 4.14. Safety test on ten random selected goals.

position at origin to goal at $(6,6)$ in the maze center. To accomplish this task, the robot must go through corridors that become narrower from outer (about 3m wide) to inner (less than 1 m wide). For both methods, we use the same 0.4 m inflation radius in A* planner. The low MPC used in EVA planner are set similar distance (0.7 m) clearance preference as our medium clearance cost curve design. Both methods use the same parameters as in Jackal race world. The results are shown in Fig. 4.15. The velocity profiles are depicted perpendicular along robot paths using purple lines (longer faster). The safety distance margin $d(\mathbf{p}, \mathscr{O}^+)$ is embedded in colorbar. As we can see from this plot, our method accomplishes this task successfully while adapting its speed according to local environments, i.e., faster when corridor is wide and slower when it becomes narrows. In contrast, the EVA planner cannot handle this task elegantly using the same hyper-parameters. Although the EVA planner runs faster at beginning when corridor is wide, the two cost terms of progress (pushing robot progress along planned path) and direction safety metric (reducing speed when robot getting closer to obstacle) are fighting each other when robot getting deep in the maze. This results in wobbly motion and snake trajectory, and eventually drive robot crash to wall.

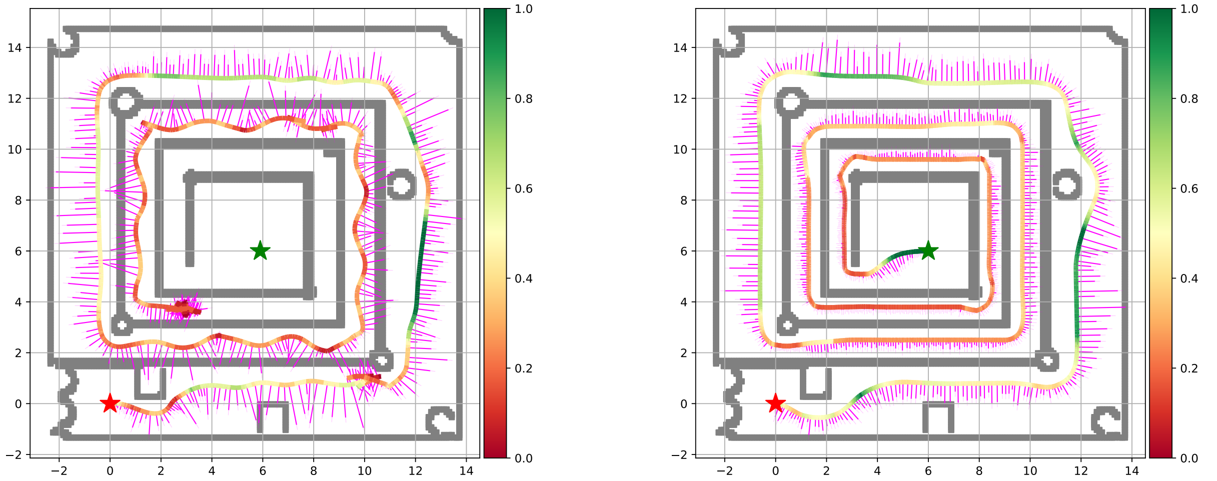
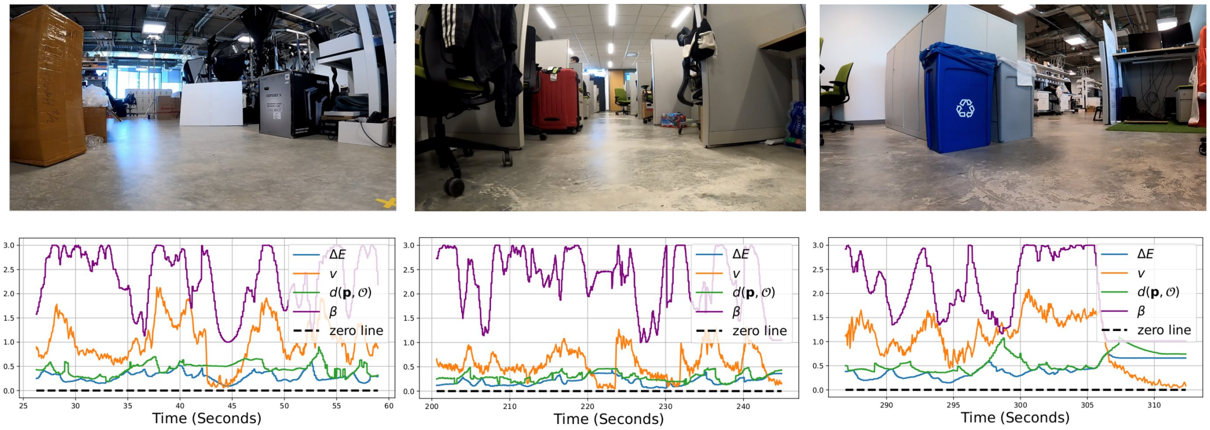


Figure 4.15. The result maze adaptivity experiment. The distance from robot to inflated obstacle space is embedded in robot path. The value of colorbar is saturated at 1 m for better visualization.

4.6.5 Hardware Demonstration in Clutter Unknown Environment

In this experiment, we test our algorithm in a large scale (40×20 m) office-like collaborative lab environment. The place is shared with multiple research lab groups with various robots, complex experiment platform, package boxes, working desks, etc., as shown in Fig. 4.16. During this experiment, we use RViz to specify a few goals over unknown regions and let the robot explore autonomously. This experiment takes about 10 minutes, a first-person video captured by onboard GoPro camera can be found at here. While exploring this place, the robot creates costmap online through occupancy grid and navigate itself safely without any collision. The experiment overview can be found in Figure. 4.16, where the final costmap is depicted Fig. 4.17. Three local scenes (marked in green boxes) are shown in the top row subplot (a)-(c) with associated quantitative results right down below each of them.

The robot cruise speed adapts to sensed local environments, slowing down when entering obstacle-dense areas and speed up when in wide open spaces as shown in these plots as shown all scenes. From the middle row plots, we can see that the speed of Jackal robot (orange curve) is higher when it is distance to obstacle (green line) is bigger. The adaptive directional SDDM boost (purple curve) increases when motion direction is aligned with local environments. This can be seen clearly in scene (2) in Fig. 4.16 when robot tries to go through a long narrow aisle. It allows robot maintains relatively high speed in straight motions and slows down when turn-and-go are required to pass obstacles (red luggage in front and green chair at behind) in the midway.



(a) scene 1

(b) scene 2

(c) scene 3

Figure 4.16. Hardware experiment in a large-scale (about 40 m × 20 m) collaborative working space. Three representative scenes 1-3 are shown: scene 1) robot going from wide open area to cluttered obstacle dense region and leave; scene 2) robot going through narrow aisle with obstacles (red luggage and green chair); scene 3) robot going through one region to another while passing narrow gaps and stop. At top row, photos of these three scenes are shown. Associated quantitative plots is plotted down below each of these scenes.

4.6.6 Moving Obstacles Experiments

In this section, we will evaluate the path tracker extension using the active governor control policy. We will test the performance of our controller in dynamically changing environments with moving obstacles. The position and velocity information of moving agents are assumed to be known to the robot. For simulated experiments, the trajectories of moving obstacles are calculated ahead and replayed at runtime. For hardware demonstration, we use Vicon motion capture system to track motions of human actors and then pass this information to Jackal robot over wireless network.

Simulation experiment. In this experiment, the robot is asked to follow a predefined path (black dash line) while avoiding six known moving obstacles (light gray circles) with

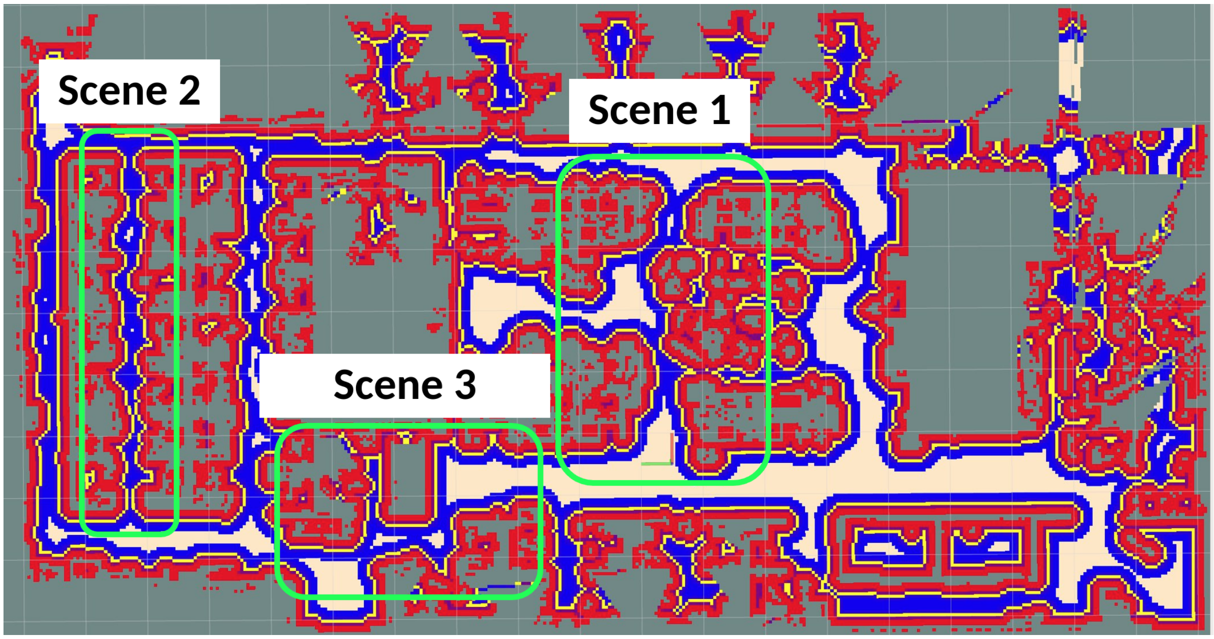


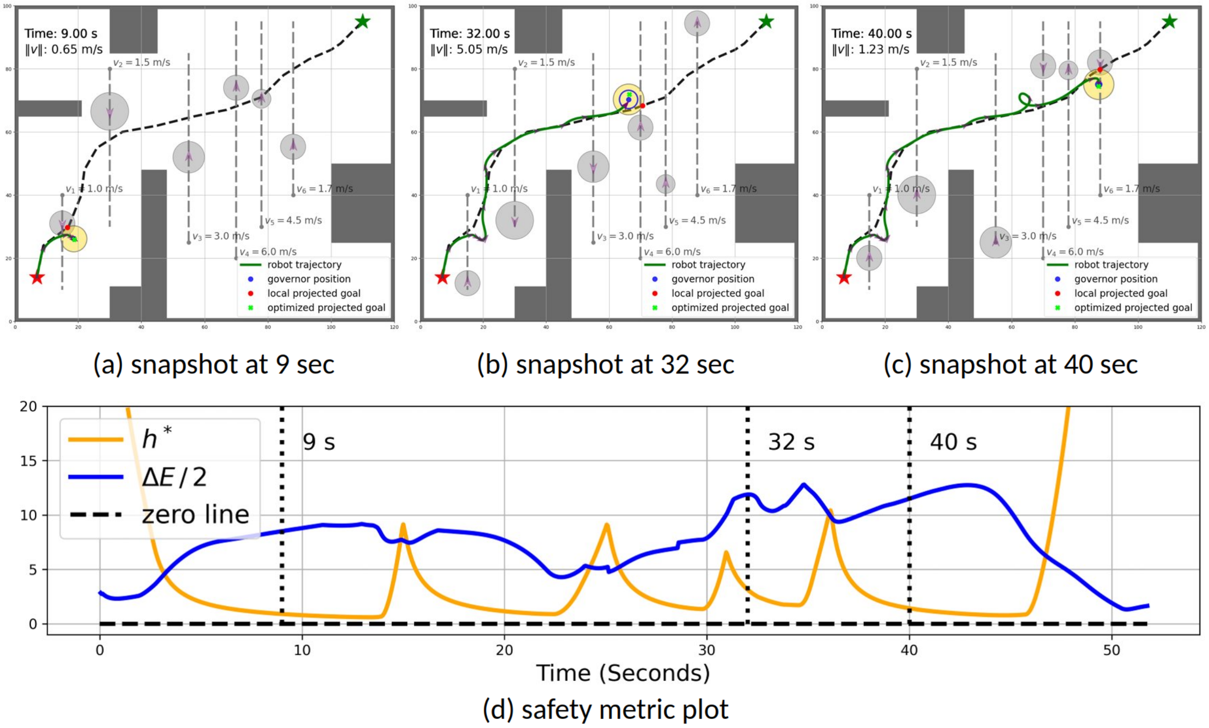
Figure 4.17. Costmap of working space. The cost value from high to low are depicted in red, yellow, blue to white.

different size and running speed. See snapshots in Fig. 4.18 for the simulation setup. Control parameters used in this experiment are summarized in Table 4.4. To handle fast moving obstacles, we use more conservative circular shape motion prediction set, i.e., $\mathscr{B}(\mathbf{g}, \|\mathbf{g}-\mathbf{p}\|)$ (blue circle) and increased control inputs bounds to $[-10, 10]$ m/sec and $[-4, 4]$ rad/s for linear and angular velocities, respectively. Three snapshots of this simulation are shown in Fig. 4.18. The resulting robot trajectory is depicted as a green line with purple arrows indicating its heading. Using active governor control policy in Sec. 4.5.2, we allow local projected goal to deviate from path temporarily, so robot can take necessary defensive maneuvers to avoid nearby incoming obstacles and quickly resume to normal path following behavior when is safe to do so.

Table 4.4. Parameter table of control gains used in moving obstacle simulation.

Parameter	k_g	k_v	k_ω	γ_i	c_1	c_2
Value	2.0	2.0	5.0	0.15	1	1

Hardware experiment. The site setup is shown in Fig. 4.19 (a) with static obstacles (boxes, walls, pillars, etc,.) and moving obstacles (two actors wearing helmets). The robot is



(a) snapshot at 9 sec

(b) snapshot at 32 sec

(c) snapshot at 40 sec

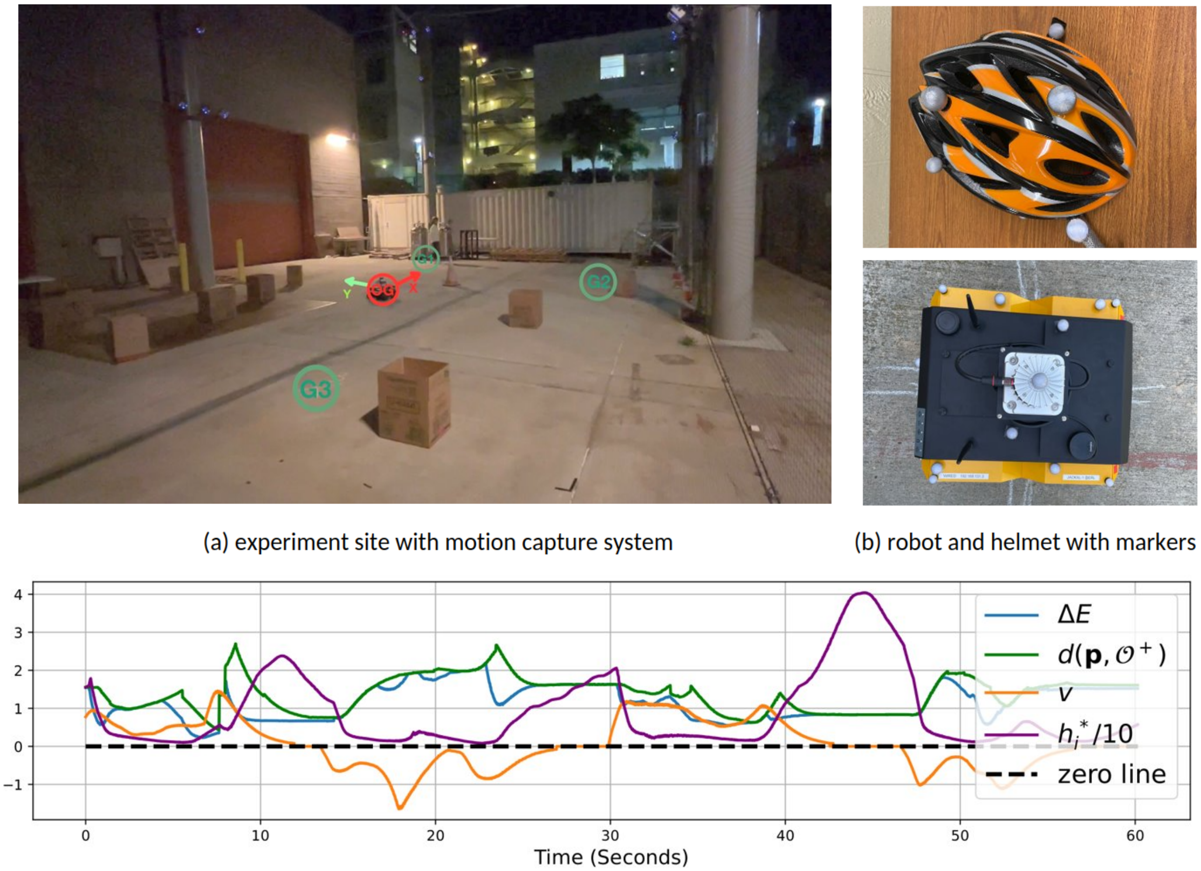
(d) safety metric plot

Figure 4.18. Summary of moving obstacle simulation with six dynamic obstacles at different running speed and sizes. When incoming obstacle endangers robot motions, the CBF constraints become active and push optimized local projected goal (green cross) away from local projected goal (red dot) allowing robot to take necessary defensive maneuvers. Three snapshots at different times are shown.

asked to go through three different goals (marked as green circle) and return to origin (red circle). The robot does not have any prior information of this environment, except receiving moving obstacle positions from Vicon system. Actors are instructed to interrupt robot motions to each goal but not being adversarial against it. Same controller parameter set are used as in Table. 4.3. Please check out experiment video at here for better visualization. From the bottom plot 4.19 (d), we see that the robot remains safe during the experiment (blue and purple line above zero).

4.7 Summary

This chapter presented an environment adaptive and safe path-following tracker via planning and control co-design. Through the usage of custom A* planner over costmap, directional safety metric and conic shape motion prediction, our reference governor-based path tracker can



(a) experiment site with motion capture system

(b) robot and helmet with markers

(d) safety metric for static and moving obstacles

Figure 4.19. Hardware experiment with Jackal robot and two moving obstacle. Actors' positions are captured by Vicon motion captures system and passed to the robot via local network. This robot re-plans at about 10 Hz online to reach different goals while avoiding moving pedestrians temporarily crossing the path. Video is available at here.

achieve safe and efficient safe navigation in challenging large-scale unknown environments. The effectiveness and adaptability are verified in various simulated and real experiments. By further integrating with control barrier function, our method can handle dynamic environments with moving obstacles under mild assumption. Future work will focus on extending geometric path planner for $SE(2)$ configuration space and developing semantic mapping algorithm can auto detect moving obstacles onboard.

Acknowledgements

Chapter 4, is based on material the following two papers. First, “Fast and safe path-following control using a state-dependent directional metric” authored by Zhichao Li, Omur Arslan and Nikolay Atanasov, which appears in International Conference on Robotics and Automation (ICRA), IEEE, 2020. The dissertation author was the primary investigator and author of this paper. Second, “EAST: Environment Aware Safe Tracking using Planning and Control Co-Design” by Zhichao Li, Yinzhuang Yi, Zhuolin Niu and Nikolay Atanasov, is currently being prepared for submission. The dissertation author was the primary investigator and author of this material.

Chapter 5

Concluding Remarks

5.1 Conclusion

In this dissertation, we developed an environment aware and safe tracking framework based on reference governor techniques. Our method bridges the gap between geometric planner and low-level stabilizing controller. With novel directional metric and dynamic safety margins, resulting control policy can navigate the robot fast and safely in complex unknown environments yet being alerted to surrounding dynamical obstacles and take necessary action to avoid them. With learning technique, our methods are robust to model error and measurement noise. With planning and control co-design, our algorithm balances safety and travel distance leaving the freedom to practitioners. Through extensive simulation and hardware experiments, we demonstrated the effectiveness of our design framework algorithms. The code is open sourced to facilitate the research community.

5.2 Future Work

There are potential extensions that can be done beyond this dissertation. For example, extending safe tracking framework to more complex dynamical systems like manipulators, under-actuated non-rigid body dynamics. Designing advance planner over $SE(2)$ and $SE(3)$ to achieve agile maneuverability. Integrated learning-based perception module to detect moving obstacles in real-time and design algorithm being robust against prediction errors.

Appendix A

Proofs of Chapter 2

Lemma 5 ([31]). *Let $(\mathscr{X},\mu)$ be a metric space and let $\mathscr{A} \subseteq \mathscr{X}$ be nonempty. The point-to-set distance function $d(\cdot,\mathscr{A}) : \mathscr{X} \mapsto \mathbb{R}$ defined by $d(\cdot,\mathscr{A}) := \inf\{\mu(\mathbf{x},\mathbf{a}) \mid \mathbf{a} \in \mathscr{A}\}$ is* 1*-Lipschitz:*

$$|d(\mathbf{x},\mathscr{A}) - d(\mathbf{y},\mathscr{A})| \leq \mu(\mathbf{x},\mathbf{y}), \quad \forall \mathbf{x},\mathbf{y} \in \mathscr{X},$$

and, hence, uniformly continuous.

Lemma 6. *The rate of change of the governor state in* (2.12) *is uniformly bounded by a constant, $\|\dot{\mathbf{g}}(t)\| \leq k_g\beta\sqrt{l^{-1}}$, and $\mathbf{g}(t)$ is continuous.*

Proof. Considering (2.12), Def. 5, and Def. 4:

$$\|\dot{\mathbf{g}}(t)\| = k_g\|\mathbf{g}(t) - \bar{\mathbf{g}}(t)\| \leq k_g\sqrt{\Delta E(t)/l}. \tag{A.1}$$

With $\Delta E(\mathbf{x},\mathbf{g}) = b(\mathbf{x},\mathbf{g})$, from Def. 1 and (2.2), we know that $\Delta E(\mathbf{x},\mathbf{g}) \leq d_s^2(\mathbf{g}(t),\mathscr{O}) \leq \beta^2$. Hence, $\|\dot{\mathbf{g}}(t)\| \leq k_g\beta\sqrt{l^{-1}}$ and $\mathbf{g}(t)$ is continuous [30]. □

Lemma 7. *The function $d_s(\mathbf{g}(t),\mathscr{O})$ is continuous when $\mathbf{g}(t) \in \mathscr{F}$.*

Proof. When $\mathbf{g}(t) \in \mathscr{F}$, from definition (2.2), we have $d_s(\mathbf{g}(t),\mathscr{O}) = \min\{d(\mathbf{g}(t),\partial\mathscr{O}),\beta\}$. The min operation is continuous, $d(\mathbf{g}(t),\partial\mathscr{O})$ is continuous because $\mathbf{g}(t)$ is continuous by Lemma 6 and $d(\cdot,\partial\mathscr{O})$ is continuous by Lemma 5. □

Lemma 8. *The Lyapunov function $V(t) = V(\mathbf{x}(t), \mathbf{g}(t))$ in* (2.8) *and the dynamic safety margin $\Delta E(t) = b(\mathbf{x}(t), \mathbf{g}(t))$ in* (2.10) *are continuous functions in time.*

Proof. From Lemma 6, we know that $\mathbf{g}(t)$ is continuous. Because $V(\mathbf{x}, \mathbf{g})$ is continuous in $\mathbf{x}$, $\mathbf{g}$, we have $V(t) = V(\mathbf{x}(t), \mathbf{g}(t))$ continuous in time. By Lemma 7, $d_s(\mathbf{g}(t), \mathscr{O})$ is continuous in time and, hence, $\Delta E(t) = d_s^2(\mathbf{g}(t), \mathscr{O}) - L^2 V(t)$ is continuous in time. □

Lemma 9. *For $\Delta E(t) = b(\mathbf{x}(t), \mathbf{g}(t))$ defined in* (2.10)*, let $T_0 \geq t_0$ be such that $\Delta E(T_0) = 0$. Then, the lower-right Dini derivative of $\Delta E(t)$:*

$$D_+ \Delta E(t) := \liminf_{h \to 0^+} \frac{\Delta E(t+h) - \Delta E(t)}{h} \tag{A.2}$$

satisfies $D_+ \Delta E(T_0) > 0$ and there exists $h > 0$, such that $\Delta E(T_0 + h) > 0$.

Proof. Note that $\Delta E(t)$ is not differentiable everywhere due to the truncated signed distance function $d_s(\mathbf{g}(t), \mathscr{O})$ in its definition. We use the lower-right Dini derivative $D_+ \Delta E(t)$ instead. Let $\hat{d}(t,h) := \max\{d_s(\mathbf{g}(t), \mathscr{O}), d_s(\mathbf{g}(t+h), \mathscr{O})\}$. From Lemma 5:

$$d_s(\mathbf{g}(t+h), \mathscr{O}) - d_s(\mathbf{g}(t), \mathscr{O}) \geq -\|\mathbf{g}(t) - \mathbf{g}(t+h)\|.$$

Hence, $\Delta E(t+h) - \Delta E(t) \geq -2\|\mathbf{g}(t) - \mathbf{g}(t+h)\| \hat{d}(t,h) + L^2(V(t) - V(t+h))$ and $D_+ \Delta E(t) \geq -2\|\dot{\mathbf{g}}(t)\| d_s(\mathbf{g}(t), \mathscr{O}) - L^2 \dot{V}(t)$. From (2.8), with $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{X}\mathbf{g}$:

$$\dot{V}(t) = 2\bar{\mathbf{x}}^\top \mathbf{P}\left((\mathbf{A} + \mathbf{B}\mathbf{K})\bar{\mathbf{x}} - \mathbf{X}\dot{\mathbf{g}}\right) = -\bar{\mathbf{x}}^\top \mathbf{Q}\bar{\mathbf{x}} - 2\bar{\mathbf{x}}^\top \mathbf{P}\mathbf{X}\dot{\mathbf{g}} \leq -\bar{\mathbf{x}}^\top \mathbf{Q}\bar{\mathbf{x}} + 2\|\mathbf{X}^\top \mathbf{P}\bar{\mathbf{x}}\| \|\dot{\mathbf{g}}\|. \tag{A.3}$$

From Lemma 6, $\|\dot{\mathbf{g}}(t)\| \leq k_g \sqrt{\Delta E(t)/l}$, and therefore:

$$D_+ \Delta E(t) \geq -2M(t)\|\dot{\mathbf{g}}(t)\| + L^2 \bar{\mathbf{x}}(t)^\top \mathbf{Q}\bar{\mathbf{x}}(t) \geq -2k_g M(t)\sqrt{\Delta E(t)/l} + L^2 \bar{\mathbf{x}}(t)^\top \mathbf{Q}\bar{\mathbf{x}}(t)$$

where $M(t) = \bar{d}(\mathbf{g}(t), \mathscr{O}) + L^2\|\mathbf{X}^\top \mathbf{P}\bar{\mathbf{x}}(t)\| \le \beta + L^2\|\mathbf{X}\|\|\mathbf{P}\|\|\bar{\mathbf{x}}(t)\| < \infty$ for bounded $\|\bar{\mathbf{x}}(t)\|$. From (2.10),

$$\lambda_{\min}(\mathbf{P})\|\bar{\mathbf{x}}(t)\|^2 \le V(t) \le \frac{\beta^2 - \Delta E(t)}{L^2}, \tag{A.4}$$

which implies $\|\bar{\mathbf{x}}\|$ and $M(t)$ are bounded, for $t = T_0$ such that $\Delta E(T_0) = 0$.

Note that $\mathbf{g}(t) \in \mathscr{LS}(\mathbf{x}(t), \mathbf{g}(t)) \subset \mathscr{F}$, so $\Delta E(t) > 0$ when $V(t) = \bar{\mathbf{x}}(t)^\top \mathbf{P}\bar{\mathbf{x}}(t) = 0$. Therefore, $\Delta E(t) = 0$ and $\bar{\mathbf{x}}(t) = 0$ cannot happen simultaneously. Plugging $t = T_0$ such that $\Delta E(T_0) = 0$ into (A.4),

$$D_+\Delta E(T_0) \ge L^2\bar{\mathbf{x}}(T_0)^\top \mathbf{Q}\bar{\mathbf{x}}(T_0) > 0 \tag{A.5}$$

since $\bar{\mathbf{x}}(T_0) \ne 0$, $\mathbf{Q} \in \mathbb{S}^n_{>0}$, and $L > 0$. Let $\dot{\gamma} \equiv 0$ with initial condition $\gamma(T_0) = 0$. By the Comparison Lemma [54], $\Delta E(T_0 + h) > \gamma(T_0 + h) = 0$ for some $h > 0$. □

Lemma 10. *Consider the closed-loop system in* (2.16).

$$\mathscr{S} := \{(\mathbf{x}, \mathbf{g}) \in \mathbb{R}^n \times \mathbb{R}^m \mid \Delta E(\mathbf{x}, \mathbf{g}) \ge 0\}$$

is positively invariant.

Proof. By inequality (A.5), $\mathscr{S}$ is forward invariant if and only if the vector field defining the joint $(\mathbf{x}, \mathbf{g})$ system in (2.16) belongs to the Bouligand tangent cone of $\mathscr{S}$ for all $(\mathbf{x}, \mathbf{g}) \in \mathscr{S}$. The tangent cone is trivial except on the boundary of $\mathscr{S}$ (see [12]) so the condition needs to be checked only for $(\mathbf{x}, \mathbf{g}) \in \partial\mathscr{S}$. Since $\mathscr{S}$ is defined implicitly by the function $\Delta E(\mathbf{x}, \mathbf{g})$, the tangent cone is equal to the hypograph of the lower-right derivative of $\Delta E(\mathbf{x}, \mathbf{g})$ [7, Prop. 3.3.2]. In other words, $\mathscr{S}$ is forward invariant if and only if for all t such that $\Delta E(t) := \Delta E(\mathbf{x}(t), \mathbf{g}(t)) = 0$, we have $D_+\Delta E(t) \ge 0$, where $D_+\Delta E(t)$ is the lower-right Dini derivative of $\Delta E(t)$ evaluated along the flow of (2.16). This is concluded in Lemma 9. □

Appendix B

Proofs of Chapter 3

B.1 Proof of Theorem 3

We do not write function arguments explicitly to simplify the notation. We also introduce the following notation for the components of $\mathbf{e}$ and $\mathfrak{p}_e$ in (3.23):

$$\begin{aligned} \mathbf{e} &= \begin{bmatrix} \mathbf{e}_{\mathbf{p}} \\ \mathbf{e}_{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} k_{\mathbf{p}} \mathbf{R}^\top \mathbf{p}_e \\ \frac{1}{2} k_{\mathbf{R}} \left(\mathbf{R}_e - \mathbf{R}_e^\top \right)^\vee \end{bmatrix}, \\ \mathfrak{p}_e &= \mathbf{M} \begin{bmatrix} \mathbf{e}_{\mathbf{v}} \\ \mathbf{e}_{\omega} \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{v} - \mathbf{R}_e^\top \mathbf{v}^* \\ \omega - \mathbf{R}_e^\top \omega^* \end{bmatrix}. \end{aligned} \tag{B.1}$$

Consider the Lyapunov function candidate in (3.24):

$$\mathscr{V} = \frac{1}{2} \mathfrak{p}_e^\top \mathbf{M}^{-1} \mathfrak{p}_e + \mathscr{U}_d + \rho \frac{d}{dt} \mathscr{U}_d, \tag{B.2}$$

where $\mathscr{U}_d = \frac{k_{\mathbf{p}}}{2} \|\mathbf{p}_e\|^2 + \frac{k_{\mathbf{R}}}{2} \operatorname{tr}(\mathbf{I} - \mathbf{R}_e)$. In the domain $\mathscr{A}$, we have [63, Prop. 1]:

$$k_{\mathbf{R}}^{-2} \|\mathbf{e}_{\mathbf{R}}\|_2^2 \leq \operatorname{tr}(\mathbf{I} - \mathbf{R}_e) \leq \frac{4 k_{\mathbf{R}}^{-2}}{4 - \alpha} \|\mathbf{e}_{\mathbf{R}}\|_2^2. \tag{B.3}$$

By the chain rule and (3.23), we have:

$$\frac{d}{dt}\mathscr{U}_d = \nabla_{\mathfrak{q}_e}\mathscr{H}_d^\top \dot{\mathfrak{q}}_e = \nabla_{\mathfrak{q}_e}\mathscr{H}_d^\top \mathbf{J}\mathbf{M}^{-1}\mathfrak{p}_e = \mathbf{e}^\top \mathbf{M}^{-1}\mathfrak{p}_e \tag{B.4}$$

Using (B.3) and (B.4), together with the Cauchy-Schwartz inequality and the sub-multiplicative property of the Euclidean norm, the Lyapunov function candidate is bounded as:

$$\begin{aligned}
\mathscr{V} &\le \frac{\lambda_2}{2}\|\mathfrak{p}_e\|^2 + \frac{k_{\mathbf{p}}^{-1}}{2}\|\mathbf{e}_{\mathbf{p}}\|^2 + \frac{2k_{\mathbf{R}}^{-1}}{4-\alpha}\|\mathbf{e}_{\mathbf{R}}\|^2 + \rho\lambda_2\|\mathbf{e}\|\|\mathfrak{p}_e\|.\\
\mathscr{V} &\ge \frac{\lambda_1}{2}\|\mathfrak{p}_e\|^2 + \frac{k_{\mathbf{p}}^{-1}}{2}\|\mathbf{e}_{\mathbf{p}}\|^2 + \frac{k_{\mathbf{R}}^{-1}}{2}\|\mathbf{e}_{\mathbf{R}}\|^2 - \rho\lambda_2\|\mathbf{e}\|\|\mathfrak{p}_e\|.
\end{aligned}$$

The bounds can be stated compactly in quadratic form using $\mathbf{z} = [\|\mathbf{e}\| \; \|\mathfrak{p}_e\|]^\top$ and $\mathbf{Q}_1$, $\mathbf{Q}_2$ in (3.26):

$$\frac{1}{2}\mathbf{z}^\top \mathbf{Q}_1 \mathbf{z} \le \mathscr{V} \le \frac{1}{2}\mathbf{z}^\top \mathbf{Q}_2 \mathbf{z}. \tag{B.5}$$

The time derivative of the Lyapunov candidate satisfies:

$$\frac{d}{dt}\mathscr{V} = \mathfrak{p}_e^\top \mathbf{M}^{-1}\dot{\mathfrak{p}}_e + \mathbf{e}^\top \mathbf{M}^{-1}\mathfrak{p}_e + \rho\mathbf{e}^\top \mathbf{M}^{-1}\dot{\mathfrak{p}}_e + \rho\dot{\mathbf{e}}^\top \mathbf{M}^{-1}\mathfrak{p}_e.$$

The term $\dot{\mathfrak{p}}_e$ is from (3.23). The term $\dot{\mathbf{e}}$ is obtained from (B.1):

$$\begin{aligned}
\dot{\mathbf{e}} &= \begin{bmatrix} \dot{\mathbf{e}}_{\mathbf{p}} \\ \dot{\mathbf{e}}_{\mathbf{R}} \end{bmatrix} = \begin{bmatrix} -\hat{\omega}\mathbf{e}_{\mathbf{p}} + k_{\mathbf{p}}\mathbf{e}_{\mathbf{v}} \\ k_{\mathbf{R}}\mathbf{E}_{\mathbf{R}}\mathbf{e}_{\omega} \end{bmatrix} \\
&= -\begin{bmatrix} \hat{\omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{e} + \begin{bmatrix} k_{\mathbf{p}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_{\mathbf{R}}\mathbf{E}_{\mathbf{R}} \end{bmatrix}\mathbf{M}^{-1}\mathfrak{p}_e,
\end{aligned} \tag{B.6}$$

where $\mathbf{E_R} = \frac{1}{2}\left[\mathrm{tr}(\mathbf{R}_e^\top)\mathbf{I} - \mathbf{R}_e^\top\right]$ satisfies $\|\mathbf{E_R}\|_2 \leq 1$ [64, Prop. 1]. Hence, we have:

$$\begin{aligned}
\frac{d}{dt}\mathscr{V} &= -\mathfrak{p}_e^\top \mathbf{M}^{-1}\mathbf{K_d}\mathbf{M}^{-1}\mathfrak{p}_e + \mathfrak{p}_e^\top \mathbf{M}^{-1}\mathbf{d} \\
&\quad - \rho\mathbf{e}^\top\mathbf{M}^{-1}\mathbf{e} - \rho\mathbf{e}^\top\mathbf{M}^{-1}\mathbf{K_d}\mathbf{M}^{-1}\mathfrak{p}_e + \rho\mathbf{e}^\top\mathbf{M}^{-1}\mathbf{d} \\
&\quad - \rho\mathfrak{p}_e^\top\mathbf{M}^{-1}\begin{bmatrix}\hat{\omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{e} + \rho\mathfrak{p}_e^\top\mathbf{M}^{-1}\begin{bmatrix}k_\mathbf{p}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_\mathbf{R}\mathbf{E_R}\end{bmatrix}\mathbf{M}^{-1}\mathfrak{p}_e.
\end{aligned}$$

To find an upper bound on $\frac{d}{dt}\mathscr{V}$, we need a few intermediate steps. First, on the domain $\mathscr{A}$, we have:

$$\left\|\begin{bmatrix}\hat{\omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\right\|_2 = \|\hat{\omega}\|_2 = \|\omega\| \leq \|\mathbf{M}^{-1}\mathfrak{p}\| \leq \lambda_2\beta. \tag{B.7}$$

Second, an upper bound on

$$\xi_1 := -\lambda_{\min}(\mathbf{K_d})\|\mathbf{M}^{-1}\mathfrak{p}_e\|^2 + \|\mathbf{M}^{-1}\mathfrak{p}_e\|\|\mathbf{d}\| \tag{B.8}$$

can be found using Young's inequality [87]:

$$-\varepsilon\|\mathbf{a}\|^2 + \eta\|\mathbf{a}\|\|\mathbf{b}\| \leq -\frac{\varepsilon}{2}\|\mathbf{a}\|^2 + \frac{\eta^2}{2\varepsilon}\|\mathbf{b}\|^2 \tag{B.9}$$

with $\varepsilon = \lambda_{\min}(\mathbf{K}_d)$, $\eta = 1$, $\mathbf{a} = \mathbf{M}^{-1}\mathfrak{p}_e$, $\mathbf{b} = \mathbf{d}$:

$$\xi_1 \leq -\frac{\lambda_{\min}(\mathbf{K}_d)}{2}\|\mathbf{M}^{-1}\mathfrak{p}_e\|^2 + \frac{1}{2\lambda_{\min}(\mathbf{K}_d)}\|\mathbf{d}\|^2. \tag{B.10}$$

Similarly, we have:

$$\xi_2 := -\lambda_1\|\mathbf{e}\|^2 + \lambda_2\|\mathbf{e}\|\|\mathbf{d}\| \leq -\frac{\lambda_1}{2}\|\mathbf{e}\|^2 + \frac{\lambda_2^2}{2\lambda_1}\|\mathbf{d}\|^2. \tag{B.11}$$

Using (B.7), (B.10), and (B.11), $\frac{d}{dt}\mathscr{V}$ is bounded by:

$$\frac{d}{dt}\mathscr{V} \leq -\frac{1}{2}\mathbf{z}^\top \mathbf{Q}_3 \mathbf{z} + k_\gamma \|\mathbf{d}\|^2, \tag{B.12}$$

where the elements of $\mathbf{Q}_3$ are provided in (3.27) and $k_\gamma = \frac{1}{2\lambda_{\min}(\mathbf{K_d})} + \frac{\rho\lambda_2^2}{2\lambda_1}$. Since the parameters ρ, $k_\mathbf{p}$, $k_\mathbf{R}$, $\mathbf{K_d}$ can be chosen arbitrarily, there exists some choice that ensures the matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$ are positive definite as shown below. The inequalities in (3.25) are obtained from (B.5) and (B.12) using the Rayleigh-Ritz inequality.

Design Parameter Choice

We use the invariant sets $\mathscr{S}_c = \{\mathbf{x} \mid \mathscr{V}(\mathbf{x}, \mathbf{x}^*) \leq c\}$ induced by the Lyapunov function to restrict the error dynamics inside the domain $\mathscr{A}$ and estimate the region of attraction.We determine $c_1 \geq 0$ such that $\dot{\mathscr{V}}$ is positive on $\mathscr{S}_{c_1}$. From (3.25), $\dot{\mathscr{V}}$ is positive when $k_\gamma \delta_\mathbf{d}^2 - k_3\|\mathbf{z}\|^2 \geq 0$, which happens when $\frac{\mathscr{V}}{k_2} \leq \frac{k_\gamma}{k_3}\delta_\mathbf{d}^2$. Hence, with $c_1 = k_2 k_\gamma \delta_\mathbf{d}^2 / k_3$, we have $\dot{\mathscr{V}} \geq 0$ on $\mathscr{S}_{c_1}$. Then, we determine $c_2 \geq 0$ such that $\mathscr{S}_{c_2} \subseteq \mathscr{A}$. From (B.3) and (B.5), we have:

$$\frac{4-\alpha}{4} k_\mathbf{R}^2 \operatorname{tr}(\mathbf{I} - \mathbf{R}_e) \leq \|\mathbf{e_R}\|^2 \leq \|\mathbf{z}\|^2 \leq \frac{\mathscr{V}}{k_1}. \tag{B.13}$$

Hence, if $\mathscr{V} \leq \frac{1}{4} k_1 k_\mathbf{R}^2 \alpha (4-\alpha)$, then $\operatorname{tr}(\mathbf{I} - \mathbf{R}_e) \leq \alpha$. Similarly, if $\mathscr{V} \leq k_1 \beta^2$, then $\|\mathfrak{p}_e\|^2 \leq \|\mathbf{z}\|^2 \leq \frac{\mathscr{V}}{k_1} \leq \beta^2$. Hence, to ensure that $\mathscr{S}_{c_2} \subseteq \mathscr{A}$, we define c_2 as:

$$c_2 := k_1 \min\left\{k_\mathbf{R}^2 \alpha(4-\alpha)/4, \beta^2\right\}. \tag{B.14}$$

To ensure that $c_1 < c_2$, the disturbance bound $\delta_\mathbf{d}$ must satisfy $\delta_\mathbf{d} < \sqrt{\frac{c_2 k_3}{k_2 k_\gamma}}$. Then, any closed-loop system trajectory that starts in $\mathscr{S}_{c_2}$ converges exponentially to $\mathscr{S}_{c_1}$ and remains within it. Recall that $\mathbf{e_p} = k_\mathbf{p} \mathbf{R}^\top \mathbf{p}_e$ and from (B.5):

$$k_\mathbf{p}^2 \|\mathbf{p}_e\|^2 = \|\mathbf{e_p}\|^2 \leq \|\mathbf{e}\|^2 \leq \|\mathbf{z}\|^2 \leq \frac{\mathscr{V}}{k_1}. \tag{B.15}$$

Hence, on $\mathscr{S}_{c_1}$, $\|\mathbf{p}_e\|^2 \leq c_1/(k_1 k_{\mathbf{p}}^2)$ and the uniform ultimate bound on the position error trajectory in (3.29) is satisfied.

Design Parameter Choice

We propose a systematic way to select parameters ρ, $k_{\mathbf{p}}$, $k_{\mathbf{R}}$, $\mathbf{K_d}$, ensuring that the matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$ are positive definite. Suppose $k_{\mathbf{p}} < \frac{4-\alpha}{4} k_{\mathbf{R}}$ and $\mathbf{K_d} = \gamma_d \mathbf{I}$ for some $\gamma_d > 0$, then we have

$$\mathbf{Q}_1 = \begin{bmatrix} k_{\mathbf{R}}^{-1} & -\rho\lambda_2 \\ -\rho\lambda_2 & \lambda_1 \end{bmatrix} \quad \mathbf{Q}_2 = \begin{bmatrix} k_{\mathbf{p}}^{-1} & \rho\lambda_2 \\ \rho\lambda_2 & \lambda_2 \end{bmatrix} \tag{B.16}$$

$$q_2 = -\rho\lambda_2^2(\gamma_d+\beta) \qquad q_3 = \gamma_d\lambda_1^2 - 2\rho\lambda_2^2 k_{\mathbf{R}}.$$

To guarantee the positive definiteness of $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$, the following requirements must be satisfied:

$$\frac{\lambda_1}{k_{\mathbf{R}}} - \rho^2\lambda_2^2 > 0,\ \frac{\lambda_2}{k_{\mathbf{p}}} - \rho^2\lambda_2^2 > 0,\ \gamma_d\lambda_1^2 - 2\rho\lambda_2^2 k_{\mathbf{R}} > 0$$

$$\rho\lambda_1\left(\gamma_d\lambda_1^2 - 2\rho\lambda_2^2 k_{\mathbf{R}}\right) - \rho^2\lambda_2^4(\gamma_d+\beta)^2 > 0.$$

All these constraints put upper bounds on ρ:

$$\rho \leq \min\left\{\sqrt{\frac{\lambda_1}{k_{\mathbf{R}}\lambda_2^2}},\ \sqrt{\frac{1}{k_{\mathbf{p}}\lambda_2}},\ \frac{\gamma_d\lambda_1^2}{2k_{\mathbf{R}}\lambda_2^2},\ \bar{\rho}_{\mathbf{Q}_3}\right\}, \tag{B.17}$$

where $\bar{\rho}_{\mathbf{Q}_3} = \frac{\gamma_d\lambda_1^3}{\lambda_2^2\left[2\lambda_1 k_{\mathbf{R}} + \lambda_2^2(\gamma_d+\beta)^2\right]}$.

Bibliography

[1] Jose Angel Acosta, Romeo Ortega, Alessandro Astolfi, and Arun D Mahindrakar. Interconnection and damping assignment passivity-based control of mechanical systems with underactuation degree one. *IEEE Transactions on Automatic Control*, 50(12):1936–1955, 2005.

[2] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *IEEE European Control Conference (ECC)*, pages 3420–3431, 2019.

[3] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control (TAC)*, 59(4):876–891, 2014.

[4] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *IEEE Conference on Decision and Control (CDC)*, pages 6271–6278, 2014.

[5] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control (TAC)*, 62(8):3861–3876, 2017.

[6] Omur Arslan and Daniel E Koditschek. Smooth extensions of feedback motion planners via reference governors. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[7] Jean-Pierre Aubin. The method of characteristics revisited. a viability approach. In *arXiv preprint: math/9906178*, 1999.

[8] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-Jacobi Reachability: A Brief Overview and Recent Advances. In *IEEE Conference on Decision and Control (CDC)*, pages 2242–2253, 2017.

[9] Andrew J Barry, Anirudha Majumdar, and Russ Tedrake. Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 484–490, 2012.

[10] Alberto Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control (TAC)*, 43(3):415–419, 1998.

[11] Priyadarshi Bhattacharya and Marina L Gavrilova. Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path. *IEEE Robotics & Automation Magazine*, 15(2):58–66, 2008.

[12] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[13] Guido Blankenstein, Romeo Ortega, and Arjan J Van Der Schaft. The matching conditions of controlled Lagrangians and IDA-passivity based control. *International Journal of Control*, 75(9):645–665, 2002.

[14] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[15] Urs Borrmann, Li Wang, Aaron D Ames, and Magnus Egerstedt. Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, pages 68–73, 2015.

[16] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

[17] José Manuel Bravo, Teodoro Alamo, and Eduardo F Camacho. Robust mpc of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10):1745–1751, 2006.

[18] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.

[19] Robert Burridge, Alfred Rizzi, and Daniel Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research (IJRR)*, 18(6):534–555, 1999.

[20] Guy Campion and Georges Bastin. On adaptive linearizing control of omnidirectional mobile robots. In *Robust Control of Linear Systems and Nonlinear Control*, pages 531–538. Springer, 1990.

[21] Mo Chen, Sylvia Herbert, and Claire J Tomlin. Exact and efficient Hamilton-Jacobi guaranteed safety analysis via system decomposition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–92, 2017.

[22] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[23] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[24] Raj Dandekar, Karen Chung, Vaibhav Dixit, Mohamed Tarek, Aslan Garcia-Valadez, Krishna Vishal Vemula, and Chris Rackauckas. Bayesian neural ordinary differential equations. *arXiv preprint arXiv:2012.07244*, 2020.

[25] Brigitte d'Andrea Novel, Georges Bastin, and Guy Campion. Dynamic feedback linearization of nonholonomic wheeled mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2527–2528, 1992.

[26] Alessandro De Luca, Giuseppe Oriolo, and Claude Samson. Feedback control of a nonholonomic car-like robot. In *Robot motion planning and control*, pages 171–253. Springer, 1998.

[27] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[28] Thai Duong and Nikolay Atanasov. Hamiltonian-based Neural ODE Networks on the SE(3) Manifold For Dynamics Learning and Control. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[29] Thai Duong and Nikolay Atanasov. Learning adaptive control for se(3) hamiltonian dynamics. *arXiv preprint arXiv:2109.09974*, 2021.

[30] Aleksei Fedorovich Filippov. *Differential equations with discontinuous righthand sides: control systems*, volume 18. Springer Science & Business Media, 1988.

[31] Simon Fitzpatrick. Metric projections and the differentiability of distance functions. *Bulletin of the Australian Mathematical Society*, 22(2):291–312, 1980.

[32] Bruce A Francis. The linear multivariable regulator problem. *SIAM Journal on Control and Optimization*, 15(3):486–505, 1977.

[33] Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[34] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 2017.

[35] Emanuele Garone and Marco M Nicotra. Explicit reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control (TAC)*, 61(5):1379–1384, 2016.

[36] Elmer G Gilbert and Ilya Kolmanovsky. Discrete-time reference governors for systems with state and control constraints and disturbance inputs. In *IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1189–1194, 1995.

[37] Elmer G Gilbert and Ilya Kolmanovsky. Set-point control of nonlinear systems with state and control constraints: A Lyapunov-function, Reference-Governor approach. In *IEEE Conference on Decision and Control (CDC)*, volume 3, pages 2507–2512, 1999.

[38] Fabio Gomez-Estern, Romeo Ortega, Francisco R Rubio, and Javier Aracil. Stabilization of a class of underactuated mechanical systems via total energy shaping. In *IEEE Conference on Decision and Control*, volume 2, pages 1137–1143, 2001.

[39] Lars Grüne and Jürgen Pannek. *Nonlinear model predictive control*. Springer, 2017.

[40] Daniel Damir Harabor and Alban Grastien. Online graph pruning for pathfinding on grid maps. In *AAAI Conference on Artificial Intelligence*, 2011.

[41] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[42] Aaron Havens and Girish Chowdhary. Forced variational integrator networks for prediction and control of mechanical systems. In *Learning for Dynamics and Control*, pages 1142–1153. PMLR, 2021.

[43] Markus Heinonen, Cagatay Yildiz, Henrik Mannerström, Jukka Intosalmi, and Harri Lähdesmäki. Learning unknown ODE models with Gaussian processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1959–1968, 10–15 Jul 2018.

[44] Sylvia L Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F Fisac, and Claire J Tomlin. FaSTrack: A modular framework for fast and guaranteed safe motion planning. In *IEEE Conference on Decision and Control (CDC)*, pages 1517–1522, 2017.

[45] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6), 2019.

[46] Minh Long Hoang, Marco Carratù, Vincenzo Paciello, and Antonio Pietrosanto. Noise Attenuation on IMU Measurement For Drone Balance by Sensor Fusion. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6, 2021.

[47] Darryl D Holm. *Geometric Mechanics*. World Scientific Publishing Company, 2008.

[48] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[49] Jie Huang. *Nonlinear output regulation: theory and applications*. SIAM, 2004.

[50] Aykut İşleyen, Nathan van de Wouw, and Ömür Arslan. Feedback motion prediction for safe unicycle robot navigation. *arXiv preprint arXiv:2209.12648*, 2022.

[51] Juraj Kabzan, Lukas Hewing, Alexander Liniger, and Melanie N. Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.

[52] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research (IJRR)*, 30(7):846–894, 2011.

[53] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on robotics and automation (TRA)*, 14(1):166–171, 1998.

[54] Hassan Khalil. *Nonlinear systems*. Prentice Hall, 2002.

[55] Nathaniel Kingry, Logan Towers, Yen-Chen Liu, Yue Zu, Yuchen Wang, Briana Staheli, Yusuke Katagiri, Samuel Cook, and Ran Dai. Design, modeling and control of a solar-powered quadcopter. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1251–1258, 2018.

[56] Sven Koenig and Maxim Likhachev. Real-time adaptive A. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 281–288. ACM, 2006.

[57] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160, 2011.

[58] Ilya Kolmanovsky, Emanuele Garone, and Stefano Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *IEEE American Control Conference (ACC)*, 2014.

[59] Ilya Kolmanovsky, Emanuele Garone, and Stefano Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *IEEE American Control Conference (ACC)*, pages 226–241, 2014.

[60] Shreyas Kousik, Sean Vaskov, Fan Bu, Matthew Johnson-Roberson, and Ram Vasudevan. Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots. *The International Journal of Robotics Research (IJRR)*, 2020.

[61] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. Tr 98-11, Comp. Sci. Dept., Iowa State University, 1998.

[62] Steven LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[63] Taeyoung Lee. Robust Adaptive Attitude Tracking on SO(3) with an Application to a Quadrotor UAV. *IEEE Transactions on Control Systems Technology*, 21(5):1924–1930, 2013.

[64] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Geometric tracking control of a quadrotor UAV on SE(3). In *IEEE Conference on Decision and Control (CDC)*, 2010.

[65] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Nonlinear robust tracking control of a quadrotor uav on se (3). *Asian Journal of Control*, 15(2):391–408, 2013.

[66] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. *Global formulations of Lagrangian and Hamiltonian dynamics on manifolds*. Springer, 2017.

[67] Zhichao Li, Omur Arslan, and Nikolay Atanasov. Fast and Safe Path-Following Control using a State-Dependent Directional Metric. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[68] Zhichao Li, Thai Duong, and Nikolay Atanasov. Safe Robot Navigation in Cluttered Environments using Invariant Ellipsoids and a Reference Governor. *arXiv preprint arXiv:2005.06694*, 2020.

[69] Zhichao Li, Thai Duong, and Nikolay Atanasov. Safe Autonomous Navigation for Systems with Learned SE(3) Hamiltonian Dynamics. Technical report, University of California, San Diego, 2021.

[70] Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in neural information processing systems (NIPS)*, pages 767–774, 2004.

[71] Shulin Liu, Kexin Guo, Xiang Yu, Lina Ma, Lihua Xie, and Lei Guo. Safe maneuvering planning for flights in complex environments. *IEEE Transactions on Industrial Electronics*, 2023.

[72] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019.

[73] Anatolii Isakovich Lurie. *Analytical mechanics*. Springer Science & Business Media, 2013.

[74] M Lutter, K Listmann, and J Peters. Deep Lagrangian Networks for end-to-end learning of energy-based control for under-actuated systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[75] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research (IJRR)*, 36(8):947–982, 2017.

[76] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[77] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *IEEE American Control Conference (ACC)*, pages 322–328, 2016.

[78] Marco M Nicotra and Emanuele Garone. The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems. *IEEE Control Systems Magazine*, 38(4):89–107, 2018.

[79] Helen Oleynikova, Alexander Millane, Zachary Taylor, Enric Galceran, Juan Nieto, and Roland Siegwart. Signed distance fields: A natural representation for both mapping and planning. In *RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics*. University of Michigan, 2016.

[80] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[81] Romeo Ortega, Arjan Van Der Schaft, Bernhard Maschke, and Gerardo Escobar. Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. *Automatica*, 38(4):585–596, 2002.

[82] Vincent Pacelli, Omur Arslan, and Daniel E Koditschek. Integration of local geometry and metric information in sampling-based motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3061–3068, 2018.

[83] Alejandro Perez, Robert Platt, George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez. LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2537–2542, 2012.

[84] Lun Quan, Zhiwei Zhang, Xingguang Zhong, Chao Xu, and Fei Gao. Eva-planner: Environmental adaptive quadrotor planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 398–404. IEEE, 2021.

[85] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

[86] Sujit Rajappa, Markus Ryll, Heinrich H Bülthoff, and Antonio Franchi. Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *IEEE international conference on robotics and automation (ICRA)*, pages 4006–4013, 2015.

[87] Mutaz Ryalat and Dina Shona Laila. A robust ida-pbc approach for handling uncertainties in underactuated mechanical systems. *IEEE Transactions on Automatic Control (TAC)*, 63(10):3495–3502, 2018.

[88] Steindor Saemundsson, Alexander Terenin, Katja Hofmann, and Marc Deisenroth. Variational integrator networks for physically structured embeddings. In *International Conference on Artificial Intelligence and Statistics*, pages 3078–3087. PMLR, 2020.

[89] Mario Santillo and Mrdjan Jankovic. Collision Free Navigation with Interacting, Non-Communicating Obstacles. *arXiv preprint arXiv:2008.12092*, 2020.

[90] Tom Schouwenaars, Bart De Moor, Eric Feron, and Jonathan How. Mixed integer programming for multi-vehicle path planning. In *IEEE European control conference (ECC)*, pages 2603–2608, 2001.

[91] Sumeet Singh, Benoit Landry, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust Feedback Motion Planning via Contraction Theory. *The International Journal of Robotics Research (IJRR)*, 2019.

[92] Eduardo D Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and optimal control theory*, pages 163–220. Springer, 2008.

[93] Eric Squires, Pietro Pierpaoli, and Magnus Egerstedt. Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance. In *IEEE Conference on Control Technology and Applications (CCTA)*, pages 1656–1661, 2018.

[94] Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio A Vela. Synthesis of control barrier functions using a supervised machine learning approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7139–7145, 2020.

[95] Zhongqi Sun, Li Dai, Kun Liu, Yuanqing Xia, and Karl Henrik Johansson. Robust MPC for tracking constrained unicycle robots with additive disturbances. *Automatica*, 90:172–184, 2018.

[96] Russ Tedrake, Ian R. Manchester, Mark Tobenkin, and John W. Roberts. LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification. *The International Journal of Robotics Research (IJRR)*, 2009.

[97] Arjan Van Der Schaft and Dimitri Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3), 2014.

[98] Dustin J Webb and Jur van den Berg. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[99] Guofan Wu and Koushil Sreenath. Safety-critical control of a planar quadrotor. In *American Control Conference (ACC)*, pages 2252–2258, 2016.

[100] Winnie Xu, Ricky TQ Chen, Xuechen Li, and David Duvenaud. Infinitely deep bayesian neural networks with stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 721–738. PMLR, 2022.

[101] Burak Yüksel, Cristian Secchi, Heinrich H Bülthoff, and Antonio Franchi. Reshaping the physical properties of a quadrotor through ida-pbc and its application to aerial physical interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[102] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: learning Hamiltonian dynamics with control. In *International Conference on Learning Representations (ICLR)*, 2019.

[103] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Dissipative SymODEN: Encoding Hamiltonian dynamics with dissipation and control into deep learning. In *ICLR Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.