Data-Driven Predictive Control Beyond Linearity: An Autonomous Driving Perspective

By

Siddharth Nair

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Chair
Professor Kameshwar Poolla
Associate Professor Koushil Sreenath

Spring 2024

Data-Driven Predictive Control Beyond Linearity: An Autonomous Driving Perspective

Abstract

Data-Driven Predictive Control Beyond Linearity: An Autonomous Driving Perspective

By

Siddharth Nair

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

Model Predictive Control (MPC) has been widely adopted in the industry for constrained optimal control, because of its straightforward transcription of the control problem and the availability of mature convex optimization solvers for efficient control synthesis. However as the name suggests, MPC requires a model of the underlying system and a quantification of the uncertainty in the assumed model for reliable performance and robustness. A vast body of work has been devoted to the analyses and design of robust and efficient MPC for linear systems with convex constraints. However, the world inherently involves nonlinear phenomena and non-convex decision-making, be it the dynamics of a bicycle, the multi-modality of a human driver, or the combinatorial problem of optimizing a route. While various MPC designs for nonlinear, non-convex settings have been studied, most suffer from at least one of the following shortcomings: (1) lack closed-loop performance guarantees or safety guarantees in the presence of uncertainty, (2) have small regions of feasibility due to conservative convexification of constraints or local linearization of dynamics, or (3) are intractable for real-time, high-frequency control.

In this dissertation, we propose data-driven algorithms that exploit the nonlinearity and non-convexity in the optimal control problem for MPC designs that are robust to uncertainty and computationally efficient. The dissertation is divided into three parts focusing on robustness, performance, and computational efficiency. Each part presents algorithms that balance all three aspects, but are designed with an emphasis on the corresponding part's theme and demonstrated on practical applications in autonomous driving.

To my parents, Hari and Smitha Nair

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude towards my advisor, Professor Francesco Borrelli, for being an inspiring mentor and giving me the freedom to pursue exciting problems throughout my stay in Berkeley. I will always appreciate his solid feedback– every meeting with him has left me wiser and with more clarity. His approach to work and seemingly infinite capacity to balance multiple threads of research are two qualities that I intend to emulate and cultivate for the rest of my life.

I would also like to thank Professor Kameshwar Poolla, Professor Koushil Sreenath, Professor Laurent El Ghaoui, and Professor Venkat Anantharam for serving on my qualifying committee. I was incredibly anxious during my qualifying exam, but I am also incredibly fortunate to have had such an amazing and supportive committee to discuss my work with. In particular, I would like to acknowledge Professor Poolla and Professor Sreenath for giving me valuable advice throughout this journey, and for being on my dissertation committee. I have fond memories of all the classes I took with my committee members, and I hope to continue putting their teachings into practice.

The Model Predictive Control lab has been a big part of my identity over the last six years. I will cherish all the "Aha!" moments, the blackboard discussions and the coffee chats that I shared with my labmates, spanning four generations of the MPC lab: Charlott Vallon, Dimitris Papadimitriou, Edward Zhu, Eric Choi, Eunhyek Joa, Fiona Kopp, Hansung Kim, Hotae Lee, Jacopo Guanetti, Mark Pustilnik, Monimoy Bujarbaruah, Nitin Kapania, Roya Firoozi, Shengfan Cao, Thomas Fork, Tony Zheng, Ugo Rosolia, Vijay Govindarajan, Xiaojing Zhang, Xu Shen, Yeojun Kim, Yvonne Stuerz, Luigi Russo, Rui Oliveira. Special thanks to Monimoy, Ugo, and Yvonne for being fantastic mentors and my reference of excellence. I sincerely appreciate your patience and effort towards giving my stream of consciousness some much-needed structure during those early years. I would like to thank my collaborators Monimoy, Ugo, Vijay, Edward, Eunhyek, Hotae, Hansung, and Luigi for helping me make this thesis happen. I have learned much from you all during our brainstorming sessions, writing/coding sprints and experiment outings to Thunderhill, CPG and RFS.

My Ph.D. journey was made enjoyable and memorable because of my friends. A heartfelt thanks to Anurag, Aviral, Karan, and Shubham for the countless dinners, late-night walks, and being my family in Berkeley. I would also like to thank my friends from IIT Bombay— Abhijeet, Nitish, Sarath, Shobhit, Siddharth, Suraj, Suriya, Vikas, and Vivek. You all have been great sources of support, laughter, and motivation. Spending time with you all helped me maintain a balance between research and leisure, providing me with necessary moments of relaxation and joy amidst deadlines and existential crises.

Finally, to my family, who have been my bedrock of support and love, I owe an immeasurable debt of gratitude. Mummy and Papa–your unconditional love, your belief in me, your sacrifices, and your unwavering support have been the cornerstone of my achievements. You have always encouraged me to dream big and push beyond my limits. I dedicate this dissertation to you. Ammumma, Muthachha and Appuppa, I thank you

for your blessings and faith in me. I would also like to thank Suraj bhaiya and his family for giving me a sense of home outside of India. Last but certainly not least, I would like to thank Anushri. Your companionship during this journey was invaluable, offering a sense of normalcy and grounding that was essential for my well-being and success. I look forward to trying many more covfefes with you.

In wrapping up, I extend my deepest thanks to all who have been a part of my Ph.D. journey, contributing in various ways to my growth and success. This dissertation is not just a testament to my efforts but a reflection of the collective support and encouragement I received. As I move forward, I carry with me not only the knowledge and skills I have acquired but also the cherished memories made at Berkeley.

Chapter 1

Introduction

Autonomous Vehicle (AV) technologies have seen a surge in popularity over the last decade, with the potential to improve the flow of traffic, safety, and fuel efficiency [2]. Moreover, the autonomous driving problem serves as a benchmark for integrating autonomy into our society; an autonomous decision-making agent must be able to interact with living beings and other agents, navigate complex environments, learn from experience, take decisions quickly, and most importantly, be reliable. While existing *semi*-autonomous vehicle technology is being gradually introduced into structured scenarios such as highway driving and low-speed parking, control for fully autonomous driving in urban settings remains elusive. The main challenges in designing control algorithms for the AV are that it must be

1. robust to uncertainty arising from its dynamics (e.g. actuation delays, unmodeled higher-order dynamics), and also the environment (e.g. the behavior of the other agents in the vicinity),

2. performant in accomplishing its tasks such as following a comfortable, time-optimal route without compromising on safety,

3. efficient in computing control actions for making quick decisions even in complex environments such as unsignalized traffic intersections

This thesis investigates data-driven Model Predictive Control (MPC) algorithms to address these challenges. MPC has been widely adopted in the industry for constrained optimal control, because of its straightforward transcription of the control problem and the availability of mature convex optimization solvers for efficient control synthesis. However, as the name suggests, MPC requires a model of the underlying system and the quantification of the uncertainty in the assumed model for reliable performance and robustness. To address this shortcoming, several advances have been made over the past decade towards incorporating data-driven models into the MPC design and analyzing the closed-loop robustness and performance for linear dynamical systems subject to

convex constraints. However, the autonomous driving problem features nonlinear phenomena such as the dynamics of the vehicle or the multi-modal behavior of the human drivers, as well as non-convex, optimization problems such as avoiding collisions with other vehicles or the combinatorial problem of optimizing a route.

In this dissertation, we propose data-driven MPC algorithms that are robust to uncertainty and computationally efficient by exploiting the nonlinearity and non-convexity in optimal control problems arising in autonomous driving. The dissertation is divided into three parts focusing on robustness, performance, and computational efficiency. Each part presents algorithms that balance all three aspects, but are designed with an emphasis on the corresponding part's theme and demonstrated on practical autonomous driving applications. Next, we formalize the notions of robustness, performance, and computational efficiency and contextualize our contributions with respect to literature.

1.1 Problem Formulation

Consider a nonlinear system (e.g. the AV) with discrete-time dynamics

$$x_{t+1} = f(x_t, u_t) + d_t, \quad d_t \in D(x_t, u_t) \tag{1.1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system's state and input respectively at time t, and $f(\cdot, \cdot)$ represents the modeled dynamics. The model uncertainty or *disturbance* d_t is assumed to belong to compact set $D(x_t, u_t)$ containing the origin, where $D(\cdot)$ is a set-valued map, $D : \mathbb{R}^{n+m} \to 2^{\mathbb{R}^n}$. The system is subject to operational constraints given by compact and convex time-varying state and input constraint sets,

$$x_t \in \mathcal{X}_t, \; u_t \in \mathcal{U}_t, \quad \forall t \geq 0. \tag{1.2}$$

Let the environment denote everything else in the world except the system (1.1), and $o_t \in \mathbb{R}^q$ be the observation of the environment at time t. Suppose that the predicted observation o_{t+1} at time $t+1$ is distributed by some time-varying conditional distribution $p_t(o_{t+1} |\; o_t, x_t, x_{t+1})$, denoted (by abuse of notation) as $o_{t+1} \sim p_t(o_{t+1} |\; o_t, x_t, x_{t+1})$. The system is subject to additional time-varying *safety* constraints

$$(x_t, o_t) \in \mathcal{S}_t \quad \forall t \geq 0, \tag{1.3}$$

which are possibly non-convex, e.g., collision avoidance constraints between the AV and surrounding vehicles. The system must operate to minimize some performance measure given by $\sum_{k \geq t} c_k(x_k, u_k)$, e.g., tracking a desired path.

The Model Predictive Control (MPC) policy $u_t^{MPC}(x_t, o_t)$ is synthesized by solving the following finite-horizon optimal control problem

$$\begin{aligned}
\min_{u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)} \quad & \mathbb{E}\left[\sum_{k=t}^{t+N-1} c_k(\bar{x}_{k|t}, u_{k|t}(\bar{x}_{k|t}, o_{k|t})) + P_N(\bar{x}_{t+N|t})\right] && \text{(1.4a)}\\
\text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}(x_{k|t}, o_{k|t})) + d_k, && \text{(1.4b)}\\
& \bar{x}_{k+1|t} = f(\bar{x}_{k|t}, u_{k|t}(\bar{x}_{k|t}, o_{k|t})), && \text{(1.4c)}\\
& o_{k+1|t} \sim p_k(o_{k+1|t} |\ o_{k|t}, x_{k|t}, x_{k+1|t}), && \text{(1.4d)}\\
& (x_{k+1|t}, u_{k|t}(x_{k|t}, o_{k|t})) \in \mathcal{X}_k \times \mathcal{U}_k, && \text{(1.4e)}\\
& \mathbb{P}\left[(x_{k+1|t}, o_{k+1|t}) \notin \mathcal{S}_{k+1}\right] \leq \epsilon, && \text{(1.4f)}\\
& \bar{x}_{t+N|t} \in \bar{\mathcal{X}}_N, && \text{(1.4g)}\\
& \forall d_k \in D(x_{k|t}, u_{k|t}(x_{k|t}, o_{k|t})),\ \forall k \in \{t, \ldots, t+N-1\}, && \text{(1.4h)}\\
& x_{t|t} = x_t,\ \bar{x}_{t|t} = x_t,\ o_{t|t} = o_t && \text{(1.4i)}
\end{aligned}$$

where the decision variables are the sequence of policies $\{u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)\}$. The feedback from the state and environments appears in (1.4i) and the MPC policy is given from the optimal solution as

$$u_t^{MPC}(x_t, o_t) = u_{t|t}^{\star} \tag{1.5}$$

The predicted state of the system at time k is denoted as $x_{k|t}$ for any realization of the model uncertainties in (1.4b), whereas $\bar{x}_{k|t}$ denotes the nominal state prediction when there is no uncertainty (1.4c). The expectation in the cost (1.4a) is computed over the probability distribution (1.4d) which induces a nominal trajectory distribution (1.4c) in closed-loop with the policies $u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)$. The terminal constraint set $\bar{\mathcal{X}}_N$ and terminal cost $P_N(\cdot)$ are designed such that the cost $\mathbb{E}[\sum_{k \geq t} c_k(x_k, u_k)]$ of the closed-loop trajectory is bounded. The policies $u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)$ are constrained such that the probabilistic environment constraints (1.4f) and the state-input constraints (1.4e) are satisfied robustly across all uncertainty realizations (1.4h). The challenges towards formulating the finite-horizon optimal control problem (1.4) and computing (1.5) are as follows.

Challenges for Robustness

A robust MPC policy (1.5) ensures that the closed-loop system trajectories satisfy the state-input constraints regardless of the uncertainty realizations, i.e.,

$$(x_t, u_t^{MPC}(x_t, o_t)) \in \mathcal{X}_t \times \mathcal{U}_t,\ \forall d_t \in D(x_t, u_t^{MPC}(x_t, o_t)),\ \forall t \geq 0,$$

and the policy is safe if it satisfies the time-varying safety constraints $(x_t, o_t) \in \mathcal{S}_t, \forall t \geq 0$. Towards ensuring robustness and safety, it is essential to quantify and address the uncertainty $D(\cdot)$ in the system model, and the probability distribution $p_t(\cdot)$ modeling the environment's evolution.

Model Uncertainty

The constraints (1.4e), (1.4h) in the MPC optimization problem are enforced such that the closed-loop system satisfies constraints robustly for any realization of the model uncertainty (provided that a feasible solution exists). Enforcing the robustified constraints requires the designer to quantify the state-input dependent uncertainty set $D(x_t, u_t)$ using data. Uncertainty quantification for MPC can be broadly classified into parametric approaches [122, 124] (e.g. deep learning, set-membership techniques), and non-parametric approaches [73, 31, 80] (e.g. Gaussian Processes (GPs), non-parametric regression) that compute state-input dependent uncertainty sets that can be efficiently incorporated into the MPC optimization problem. Non-parametric approaches such as GPs require minimal assumptions on the underlying uncertainty but are inefficient in terms of sample complexity and memory footprint. Parametric approaches enjoy smaller memory footprints and have better sample efficiency but are tractable and provide guaranteed outer approximations of the uncertainty sets only under restrictive assumptions (e.g. linearly parameterized with known uncertainty features, known function class).

In this dissertation, we develop convex programming-based set-membership techniques to quantify the state-input-dependent uncertainty that combines the strengths of both parametric and non-parametric approaches. We quantify the system's uncertainty using non-parametric set membership techniques using data and statistically estimated *side-information* (e.g., Lipschitz constants, sector bounds). Then, we compute an efficient uncertainty representation using convex programming that is guaranteed to outer-approximate the true uncertainty set $D(x_t, u_t)$ given accurate side-information. This technique features in an adaptive robust MPC algorithm and a learning-based MPC for nonlinear systems that guarantees robust constraint satisfaction in the presence of state-dependent uncertainty.

Environmental Uncertainty

Towards satisfying the time-varying environmental constraints $(x_t, o_t) \in \mathcal{S}_t, \forall t \geq 0$, the probabilistic constraints (1.4f) are enforced over the probability distribution (1.4d), which is often difficult to model with first principles but rather learned from large datasets instead [64, 29]. These distributions are typically dynamically varying, non-Gaussian and possibly unbounded, so robustness to environmental uncertainty requires that (1.5) can be computed reliably and tractably at each time step. In the context of autonomous driving, significant research has been devoted to modeling the behavior of other agents in traffic as multi-modal distributions [20, 37, 118]. These distributions model the un-

certainty induced by partial observability of human drivers, where the discrete modality/intention of the human driver is modeled as a discrete random variable and the variability in the execution of the intended maneuver is modeled as a continuous random variable. Motion planning with multi-modal distributions using MPC has become popular for synthesizing dynamically feasible trajectories and handling multi-modal predictions [146, 133, 111, 13, 38, 147]. The works [146, 133, 147] propose MPC algorithms that suitably reformulate collision avoidance chance constraints for a Gaussian Mixture Model (GMM) as a nonlinear program that can be solved efficiently using tailored solvers. However, the optimization problem is formulated to find a *single* open-loop input sequence that satisfies the chance constraints for all modes and possible evolutions of the environmental uncertainty (given by the GMM), which leads to either very conservative/sub-optimal solutions or worse, no solution at all (infeasibility of the MPC problem). The approaches in [38, 13, 111] propose MPC schemes that address this issue by optimizing over *multi-modal* input sequences, one for each discrete mode, which enhances the feasibility of the MPC optimization problem. The multi-modal input sequences encode feedback on the discrete modes over the prediction horizon of the finite-horizon optimal control problem. However, these approaches still suffer from infeasibility arising from the continuous uncertainty in the discrete modes. For example, a traffic vehicle can perform an unprotected left (discrete mode) at an intersection with varying turning radii and speeds (continuous uncertainty).

In this dissertation, we propose a Stochastic MPC formulation that optimizes over multi-modal feedback policies $u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)$ for multi-modal predictions. Finding feedback policies requires optimization over the infinite-dimensional space of functions which is computationally intractable. To overcome this issue, we propose a policy parameterization that can be optimized using convex optimization. For autonomous driving, the multi-modal predictions are obtained by training a deep neural network [37] over a traffic dataset [64] to output a GMM. The policy parameterization is designed for feedback over both, discrete modes and continuous observations of the TVs' states. Our formulation also includes a novel multi-modal chance constraint reformulation that simultaneously allocates risk levels for the various modes based on their probabilities. Our approach is evaluated in various autonomous driving scenarios via simulations and hardware experiments. Our findings indicate that the proposed approach exhibits a significant reduction in conservatism when compared to the conventional approaches that optimize over open-loop or multi-modal control sequences.

Challenges to Performance

The performance of the system in closed-loop with (1.5) can be measured as the trajectory cost $\sum_{t\geq 0} c_t(x_t, u_t^{MPC}(x_t, o_t))$ of the closed-loop trajectory provided that it satisfies constraints $(x_t, u_t) \in \mathcal{X} \times \mathcal{U}$, and is safe $(x_t, o_t) \in \mathcal{S}_t$. The key considerations for achieving good performance are that the policy (1.5) must be computable by solving (1.4) in

closed-loop $\forall t \geq 0$ and that the system, environment constraint formulations in (1.4) are not overly conservative in modeling the optimal control problem.

Local Approximations of Dynamics

Computing the closed-loop trajectory cost $\sum_{t\geq 0} c_t(x_t, u_t^{MPC}(x_t, o_t))$ necessitates that (1.4) is feasible $\forall t \geq 0$, and that the resulting trajectory cost is bounded. The MPC policy satisfies these conditions if the following properties hold:

1. The system and environment start from a state x_0, e_0 at $t = 0$ for which (1.4) is feasible and remains persistently feasible for all $t \geq 0$. The MPC policy is then said to be recursively feasible.

2. The policy (1.5) is *stabilizing* for a stage cost $c_k(x_k, u_k)$ that measures the distance from a system equilibrium, or the error in tracking a reference trajectory.

A popular approach for designing the MPC policy (1.5) to satisfy these properties is to design the terminal constraint set $\bar{\mathcal{X}}_N$ (1.4g) to be a *control invariant set* for the nominal system, i.e.,

$$\bar{x}_t \in \bar{\mathcal{X}}_N \implies \exists \tilde{u}_t(\bar{x}_t, o_t) \in \mathcal{U}_t : \bar{x}_{t+1} \in \bar{\mathcal{X}}_N, \tag{1.6}$$

and design the terminal cost $P_N(\cdot)$ as a convex, positive definite *Control Lyapunov Function* (CLF) on $\bar{\mathcal{X}}_N$,

$$P_N(\bar{x}_{t+1}) - P_N(\bar{x}_t) \leq c_t(\bar{x}_t, \tilde{u}_t(\bar{x}_t, o_t)). \tag{1.7}$$

The efficient computation of the terminal constraint and terminal cost is well-studied for linear systems using first principles techniques (involving polytopic computations, semi-definite programming) or learning these components from data. For nonlinear systems, it is difficult to compute control invariant sets and CLFs in general, especially within constraints. There are broadly two model-based approaches for computing these terminal components. The first approach uses linearized models of the nonlinear system to construct these components using techniques developed for linear systems. The resulting MPC schemes with this approach tend to have a small set of states from which (1.4) is feasible because of the locally designed terminal components. The second approach uses sum-of-squares (SOS) programming to construct these components globally for polynomial systems [127, 74]. However the resulting SOS programs are often difficult to solve for high-dimensional systems, and moreover, are challenging to incorporate into (1.4) for efficient computation of the MPC policy (1.5). The work in [113] proposes an approach to iteratively learn control invariant sets and CLFs for arbitrary nonlinear systems from historical data. These learned terminal components are convex by construction for constrained linear systems and can be efficiently incorporated into the MPC design. For nonlinear systems, the control invariant set is a set of discrete points in the state space

(given by the historical trajectory data), and the CLF is defined only on this discrete set. Consequently, the resulting optimization problem (1.4) is a Mixed-Integer Nonlinear Program (MINLP) which is impractical for real-time computation of the MPC policy (1.5).

In this dissertation, we build on the ideas of [113] to learn *convex* control invariant sets and CLFs from historical data for a specific class of nonlinear systems called *difference flat* systems [56, 135]. Such systems are often found in robotic applications, e.g. a kinematic bicycle, a quadcopter, and a DC motor. The constructed terminal components capture global properties of the nonlinear system via the trajectory data, in contrast to local, linearization-based approaches in the literature. These terminal components are integrated into a computationally efficient Learning MPC algorithm for nonlinear systems that is recursively feasible, stabilizing, and uses historical data to improve the closed-loop trajectory cost iteratively. The algorithm was successfully demonstrated for autonomous racing using a 1/10 scale vehicle for robust constraint satisfaction, and iterative lap-time improvement.

Conservative Environmental Constraints

The closed-loop performance of the system is also affected by the accuracy of the environment model $p(o_{t+1}|\, o_t, x_t, x_{t+1})$ and the transcription of the environmental constraints as the set-containment constraint $(x_t, o_t) \in \mathcal{S}_t$. Ideally, the satisfaction of the chance constraint (1.4f) in the MPC should imply safety without sacrificing performance. However, there is a trade-off between these aspects:

1. The distribution $p(o_{t+1}|\, o_t, x_t, x_{t+1})$ must capture the uncertainty in the realizable environment evolutions $o_t, o_{t+1}, \ldots$ to compute the policies $u_{t|t}, \ldots, u_{t+N-1|t}(\cdot)$ satisfying (1.4f). However, such policies may not exist if $p(o_{t+1}|\, o_t, x_t, x_{t+1})$ pessimistically over-estimates the uncertainty.

2. The constraint $(x_t, o_t) \in \mathcal{S}_t$ must model safety but also allow tractable solution of (1.4). A conservative model of safety may be computationally tractable but at the expense of closed-loop performance.

For designing the MPC (1.4) towards autonomous driving, the AV must avoid collisions with the surrounding vehicles, pedestrians while driving comfortably and completing the trip on time. The collision avoidance constraints are described by the safety constraints $(x_t, o_t) \in \mathcal{S}_t$, where the environment observation o_t comprises the surrounding agents. There are various choices for modeling the environmental uncertainty (1.4d) and formulating the chance constraints (1.4f) within the MPC. Gaussian distributions are a common modeling choice for the uncertainty owing to the invariance to affine transformations and closed-form expressions for affine chance constraints [26, 25]. For distributions with compact support, robust optimization approaches [123] are used where

$p(o_{t+1}|\ o_t, x_t, x_{t+1})$ is modeled as a uniform distribution and ϵ is set to 0 in (1.4f). Distributionally robust formulations are also becoming increasingly popular to improve robustness to distributional assumptions [139, 121], by robustifying the chance constraint against a set of distributions. The collision avoidance problem involves checking if there is a non-empty intersection between two general sets (corresponding to the geometries of the agent and obstacle). This is a non-convex problem and NP-hard in general [33], and application-specific simplifications are commonly used for the improving the tractability of the resulting optimal control problem. The most common simplifications involve convexifying one or both of the sets as a 1) point, 2) affine space, 3) sphere, 4) ellipsoid, or 5) polytope. Combinations of 1)-4) are convenient for their simplicity and computational tractability, but tend to be conservative since the shape of an actual car is not well represented by such sets. Polytopes offer compact representations for obstacles in tight environments and are popular in autonomous driving applications, but using them in the collision avoidance problem results in non-smooth constraints, which require specialized solvers or mixed-integer reformulations.

In this dissertation, we develop an MPC algorithm for collision avoidance for dynamic obstacles represented as general convex sets, with prediction uncertainty $p_t(\cdot)$. We use the dual formulation of collision avoidance; checking for existence of a separating hyperplane between two sets. This perspective provides smooth collision-checking conditions for convex sets (including polytopes) by introducing additional dual variables. We consider three different uncertainty descriptions: 1) Arbitrary distributions with polytopic support, 2) Gaussian distributions, and 3) Arbitrary distributions with the first two moments known. The MPC finds an optimal sequence of parameterized agent-obstacle state feedback policies $u_{t|t}, u_{t+1|t}(\cdot), \ldots, u_{t+N-1|t}(\cdot)$ to react to different trajectory realizations of the agent and obstacles along the prediction horizon. We present a systematic performance evaluation of the MPC along axes of (i) mobility, (ii) comfort, (iii) conservatism and (iv) computational efficiency at a simulated traffic intersection. Our findings indicate that the proposed approach exhibits a significant performance improvement when compared to the conventional approaches.

Efficiency

AVs need to be sufficiently reactive and make quick decisions to accomplish tasks in dynamic and uncertain environments. MPC offers a unifying framework for low-level feedback control and high-level planning but requires the solution of the optimization problem (1.4) at high frequency ($\geq$ 10 Hz) for real-time computation of the policy (1.5). This becomes increasingly difficult as the environment involves multiple agents interacting with the AV, or the time-varying constraints $(x_t, o_t) \in \mathcal{S}_t$ are combinatorial in nature, e.g., planning multiple lane changes.

Combinatorial Decision-making

Multi-parametric Mixed-Integer Programming (mp-MIP) is a convenient framework for modeling various non-convex motion planning and constrained optimal control problems [66]. The mixed-integer formulation can model constraints such as collision avoidance [86], mixed-logical specifications [128] and mode transitions for hybrid dynamics [61]. The multi-parametric nature of these mp-MIPs arises from requiring to solve these problems for different initial conditions, obstacles configurations or system constraints—all of which affect the MIP solution. However, computing MPC solutions given by MIP formulations of (1.4) is $\mathcal{NP}-$hard and challenging for real-time ($\geq$ 10Hz) applications. Prior approaches to address this issue can be broadly categorized into two categories: (i) Explicit MPC [21, 15] which involves offline computation of the mp-MIP solution map, and (ii) Machine Learning approaches to predict heuristics or partial solutions to recover the mp-MIP solution quickly [87, 149, 18, 34]. The first approach produces certifiably optimal solutions but is tractable only for moderately sized problems. The second approach scales favorably with problem size but lack safeguards against poor quality predictions during deployment.

In this dissertation, we use supervised learning to accelerate the computation of MPC policies given by multi-parametric Mixed Integer Linear Programs (mp-MILPs) for real-time control. We develop a supervised learning framework for fast solution of the mp-MILP to predict parametric strategies for fast solution computation, along with sub-optimality certificates. The sub-optimality certificates provide *a priori* quantification of the predictions' quality before applying the computed control to ensure safety. The approach also shows favorable performance in terms of computation time compared to state-of-the-art MILP solvers (Gurobi, Mosek, SCIP, GLPK) for real-time motion planning using mixed-integer MPC.

Scalability for Complex, Interactive Environments

Motion planning for AVs in complex urban environments involving multiple human-driven and autonomous vehicles poses a significant challenge, leading to the development of various solutions for planning and behavior prediction. These solutions can be categorized into three broad approaches: (i) Hierarchical Prediction and Planning [118, 38, 99]: where a sophisticated prediction architecture provides forecasts of the surrounding vehicles which is used for motion planning, (ii) Model-based Integrated Planning and Prediction [48, 148, 105]: where planning and behavior prediction are simultaneously obtained by game-theoretic and joint-optimization approaches for all vehicles, and (iii) End-to-End Learning-based Prediction and Control [78, 65]: where a sophisticated neural network, trained using imitation/reinforcement learning algorithms on realistic datasets, implicitly and jointly forecasts the behavior of surrounding vehicles and a motion plan for the autonomous vehicle. Each of these approaches suffers from either scalability for complex driving scenarios or safety of the computed motion plans.

In this dissertation, we develop a hierarchical approach for scalable motion planning using MPC for dense, interactive traffic scenarios. The main idea is to collect a dataset of sparse solutions (obtained using l_1-regularization) to the MPC problem, and then train a transformer-based [130] supervisor using imitation learning to prescribe which constraints and decision variables can be eliminated online before solving the MPC optimization problem. This defines a reduced MPC problem that can be solved more efficiently. To ensure safety online, we use strong duality and sensitivity analysis to robustify the prescriptions against the prediction errors of the supervisor. Our approach demonstrates considerable improvement in computing the motion plans at a simulated intersection compared to the MPC approach that considers all agents and constraints in the scene.

1.2 Outline

We now present an outline of this dissertation in the following section. The dissertation is divided into three parts: Robustness, Performance, and Efficiency.

Part 1: *Robustness*

Chapter 2: Data-driven Graph Approximations for Model Uncertainty Quantification and Robust Predictive Control

We propose a data-driven approach to obtain bounds on the unmodeled, uncertain dynamics via semi-definite programming for robust control. The uncertainty in the system is assumed additive, state-input dependent, and with knowledge of additional side-information (such as Lipschitz constants, sector bounds). We use a non-parametric technique for online identification of the system uncertainty by approximating its *graph* defined by quadratic inequalities. This is then used in developing a Robust Adaptive MPC algorithm which at any given time, by solving a set of *convex* optimization problems, guarantees robust constraint satisfaction for the closed-loop system for all possible values of system uncertainty. The uncertainty estimate is refined with data using set-membership techniques. We highlight the efficacy of the proposed framework via numerical examples.

This chapter is adapted from the articles:

1. Nair*, Siddharth H., Monimoy Bujarbaruah*, and Francesco Borrelli. ”Modeling of dynamical systems via successive graph approximations.” IFAC-PapersOnLine 53.2 (2020): 977-982.

2. Bujarbaruah*, Monimoy, Siddharth H. Nair*, and Francesco Borrelli. ”A semi-definite programming approach to robust adaptive MPC under state dependent uncertainty.” 2020 European Control Conference (ECC). IEEE, 2020.

Chapter 3: Predictive Control with Uncertain, Multi-modal Predictions of the Environment

We propose a Stochastic MPC (SMPC) formulation for path planning with autonomous vehicles in scenarios involving multiple agents with multi-modal predictions. The multi-modal predictions capture the uncertainty of urban driving in distinct modes/maneuvers (e.g., yield, keep speed) and driving trajectories (e.g., speed, turning radius), which are incorporated for multi-modal collision avoidance chance constraints for path planning. In the presence of multi-modal uncertainties, it is challenging to reliably compute feasible path planning solutions at real-time frequencies ($\geq 10Hz$). Our main technological contribution is a convex SMPC formulation that simultaneously (1) optimizes over parameterized feedback policies and (2) allocates risk levels for each mode of the prediction. The use of feedback policies and risk allocation enhances the feasibility and performance of the SMPC formulation against multi-modal predictions with large uncertainty. We evaluate our approach via simulations and road experiments with a full-scale vehicle interacting in closed-loop with virtual vehicles. We consider distinct, multi-modal driving scenarios: 1) Negotiating a traffic light and a fast, tailgating agent, 2) Executing an unprotected left turn at a traffic intersection, and 3) Changing lanes in the presence of multiple agents. For all of these scenarios, our approach reliably computes multi-modal solutions to the path-planning problem at real-time frequencies.

This chapter is adapted from the articles:

1. Nair*, Siddharth H., Govindarajan*, Vijay, et al. "Stochastic mpc with multi-modal predictions for traffic intersections." 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2022.

2. Nair*, Siddharth H., Govindarajan*, Vijay et al. "Stochastic mpc with dual control for autonomous driving with multi-modal interaction-aware predictions." 15th International Symposium on Advanced Vehicle Control, 2022.

3. Nair*, Siddharth H., Lee*, Hotae, Joa*, Eunhyek et al. "Predictive control for autonomous driving with uncertain, multi-modal predictions." arXiv:2310.20561 (2023).

Part 2 : *Performance*

Chapter 4: Output-lifted Learning Model Predictive Control

We propose an approach for designing Robust Learning Model Predictive Control (LMPC) policies for a class of nonlinear systems with additive, unmodeled dynamics, to iteratively improve closed-loop performance by learning value function approximations and model uncertainty estimates. The nominal dynamics are assumed to be difference flat, i.e., the state and input can be reconstructed using flat output sequences, and some

side-information is assumed to be available for the unmodeled dynamics. For the considered class of systems, we show how to use historical trajectory data to 1) construct value function approximations using *learned* convex control invariant sets and convex control Lyapunov functions in the space of output sequences and 2) obtain bounds on the unmodeled dynamics using techniques from Chapter 2 for robust control. Contrary to prior works for constructing control invariant sets and control Lyapunov functions for nonlinear systems, our construction does not require local approximation of the dynamics and instead, exploits the nonlinearity implicitly using the historical data. These components are used in the design of a Robust MPC state-feedback policy that iteratively collects data to learn the value function and system uncertainty while enjoying theoretical guarantees of robust constraint satisfaction, asymptotic convergence to a desired goal set, and non-decreasing closed-loop performance at each policy update. Finally, simulation results demonstrate the effectiveness of the proposed strategy on a minimum time control problem using a constrained nonlinear and uncertain vehicle model.

This chapter is adapted from the articles:

1. Nair, Siddharth H., Ugo Rosolia, and Francesco Borrelli. "Output-lifted learning model predictive control." IFAC-PapersOnLine 54.6 (2021): 365-370.

2. Nair, Siddharth H., and Francesco Borrelli. "Robust Output-Lifted Learning Model Predictive Control." arXiv:2303.12127 (2023).

Chapter 5: Optimization-based Collision Avoidance in Dynamic, Uncertain Environments

We propose a Model Predictive Control (MPC) for collision avoidance between an autonomous agent and uncertain, dynamic obstacles to improve closed-loop performance by designing tight formulations of collision avoidance in the presence of uncertainty. The collision avoidance constraints are imposed by enforcing positive distance between convex sets representing the agent and the obstacles, and tractably reformulating them using Lagrange duality. This approach allows for smooth collision avoidance constraints even for polytopes, which otherwise require mixed-integer or non-smooth constraints. We consider three widely used descriptions of the uncertain obstacle position: 1) Arbitrary distribution with polytopic support, 2) Gaussian distributions and 3) Arbitrary distribution with first two moments known, and obtain deterministic reformulations of the collision avoidance constraints. The proposed MPC formulation optimizes over feedback policies to reduce conservatism in satisfying the collision avoidance constraints. The proposed approach is validated using simulations of traffic intersections in CARLA, and the closed-loop performance gains are evaluated along axes of (i) mobility, (ii) comfort, (iii) conservatism and (iv) efficiency.

This chapter is adapted from the article:

1. Nair, Siddharth H., Eric H. Tseng, and Francesco Borrelli. "Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control." 2022 IEEE 61st Conference on Decision and Control (CDC). IEEE, 2022.

Part 3 : *Efficiency*

Chapter 6: Learning for Mixed-Integer Predictive Control with Parametric Sub-optimality Certificates

We propose a supervised learning framework for computing solutions of multi-parametric Mixed Integer Linear Programs (MILPs) that arise in Model Predictive Control. Our approach also quantifies sub-optimality for the computed solutions. Inspired by Branch-and-Bound techniques, the key idea is to train a Neural Network or Random Forest which, for a given parameter, predicts a strategy consisting of (1) a set of Linear Programs (LPs) such that their feasible sets form a partition of the feasible set of the MILP and (2) an integer solution. For control computation and sub-optimality quantification, we solve a set of LPs online in parallel. We demonstrate our approach for a motion planning example and compare against various commercial and open-source mixed-integer programming solvers.

This chapter is adapted from the article:

1. Russo*, Luigi, Nair*, Siddharth H., et al. "Learning for online mixed-integer model predictive control with parametric optimality certificates." IEEE Control Systems Letters (2023).

Chapter 7: Learning Safe Supervisors for Accelerating Model Predictive Control in Interactive Environments

We propose a hierarchical architecture designed for scalable real-time Model Predictive Control (MPC) in complex, multi-modal traffic scenarios. This architecture comprises two key components: 1) RAID-Net, a novel attention-based Recurrent Neural Network that predicts relevant interactions along the MPC prediction horizon between the autonomous vehicle and the surrounding vehicles using Lagrangian duality, and 2) a reduced Stochastic MPC problem that eliminates irrelevant collision avoidance constraints, enhancing computational efficiency. Our approach is demonstrated in a simulated traffic intersection with interactive surrounding vehicles, showcasing a 12x speed-up in solving the motion planning problem.

This chapter is adapted from the articles:

1. Kim*, Hansung, Nair*, Siddharth H., and Francesco Borrelli. "Scalable Multi-modal Model Predictive Control via Duality-based Interaction Predictions." arXiv: 2402.01116 (2024).

1.3 Notational Conventions

Sets

The index set $\{k_1, k_1+1, \ldots, k_2\}$ is denoted by $\mathbb{I}_{k_1}^{k_2}$. The cardinality of a discrete set $\mathcal{S}$ is denoted by $|\mathcal{S}|$ (e.g., $|\mathbb{I}_{k_1}^{k_2}| = k_2 - k_1 + 1$). $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$ and $\|\cdot\|_p$ denotes the $p-$norm. With slight abuse of notation, $\|\cdot\|$ also denotes the norm on $\mathbb{R}^{n\times n}$ induced by the Euclidean norm. The convex hull of a set S is denoted as $\text{cvx}(S)$. For a proper cone $\mathcal{K}$ and $x, y \in \mathcal{K}$, we have $x \succeq_{\mathcal{K}} y \Leftrightarrow x - y \in \mathcal{K}$. The dual cone of $\mathcal{K}$ is given by the convex set $\mathcal{K}^* = \{y | y^\top x \geq 0, \forall x \in \mathcal{K}\}$.

Matrices

$I_n \in \mathbb{R}^{n\times n}$ is the identity matrix. 0_n is a column vector of n zeros and O_n is a $n \times n$ matrix of zeros. $[x]_i$ denotes the ith element of vector $x \in \mathbb{R}^n$, $[M]_{ij}$ denotes the ith element in the jth column of matrix $M \in \mathbb{R}^{n\times m}$ and $[M]_i$ denotes its ith column. We denote the Hermitian of a real square matrix M as $\text{He}(M) = \frac{M+M^\top}{2}$. Given matrices $A_1, \ldots A_N \in \mathbb{R}^{n\times n}$, we denote by $\text{blkdiag}(A_1, \ldots, A_N) \in \mathbb{R}^{Nn\times Nn}$ as the matrix with block diagonals given by $A_1, \ldots, A_N$ and zeros everywhere else. As a shorthand, we write the symmetric block matrix $\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}$ as $\begin{bmatrix} A & B \\ \star & C \end{bmatrix}$

Operators

The binary operator $\otimes$ denotes the Kronecker product. Given two sets A and B, their Minkowski sum, Pontryagin difference and Cartesian product is defined as

$$\begin{aligned} A \oplus B &= \{a + b \mid a \in A, b \in B\}, \\ A \ominus B &= \{a \in A \mid a + b \in A, \forall b \in B\}, \\ A \times B &= \{(a, b) \mid a \in A, b \in B\} \end{aligned}$$

The partial derivative of function $f(x, u)$ with respect to x at $(x, u) = (x_0, u_0)$ is denoted by $\partial_x f(x_0, u_0)$. We use the Fraktur font $\mathfrak{f}(\cdot)$ for denoting functionals (a function that acts on functions).

Part I

Robustness

Chapter 2

Data-driven Graph Approximations for Model Uncertainty Quantification and Robust Predictive Control

2.1 Overview

Characterization of system model and associated uncertainties is of paramount importance while dealing with autonomous systems. In recent times, as data-driven decision making and control becomes ubiquitous [110, 115], system identification methods are being integrated with control algorithms for control of uncertain dynamical systems. In computer science community, data driven reinforcement learning algorithms [17, 136] have been extensively utilized for policy and value function learning of uncertain systems. In control theory, if the actual model of a system is unknown, adaptive control [119] strategies have been applied for simultaneous system identification and control. Techniques for system modelling and identification have been traditionally rooted in statistics and data sciences [52, 60]. Statistical models that describe observed data, can be classified into parametric [63], non-parametric and semi-parametric [58] models.

Parametric models assume a model structure a priori, based on the application and domain expertise of the designer. In almost all of classical adaptive control methods, parametric models are learned from data in terms of point estimates, and asymptotic convergence of such estimates are proven under persistence of excitation conditions. The concept of online model learning and adaptation has been extended to systems under constraints as well, after obtaining a set or a confidence interval containing possible realizations of the system model. Gaussian Mixture Modeling (GMM) [53, 70] has also been used to identify unknown system parameters via the expectation maximization algorithm. Although robust satisfaction of constraints is ensured with such set based parametric uncertainty modeling, but computational tractability can be obtained only for linear systems, due to the need for computation of robust invariant terminal sets

[21] for the MPC problem. For nonlinear uncertain systems subject to constraints, non-parametric modeling of uncertainty is therefore used to obtain lesser conservatism, at the expense of robust guarantees. Moreover, parametric models are restricted only to specified forms of function classes which may not accurately reflect reality.

Non-parametric models are increasingly being utilized for the richness of model estimates, whereby the model structure is also inferred from data. For non-parametric modeling of systems, Gaussian Process (GP) regression [109] has been one of the most widely used tools in control theory literature. GP regression keeps track of a Gaussian distribution over infinite dimensional function spaces, in terms of a mean function and a covariance kernel, which are updated with data. Given any system state, GP regression returns the mean function value at that state, along with a confidence interval. Kernel regression methods such as local linear regression [50, 112] and Nadaraya-Watson estimator [120] are some other non-parametric methods for system identification and control. MPC with GP and Kernel regression modeled uncertainty is presented in [62, 73, 124, 80]. Estimates obtained using these methods often come with deterministic or statistical intervals all possible realizations of the uncertainty at a given state. However, these non-parametric estimates are computationally expensive for real-time inference as the dataset grows. Non-parametric approaches are also only able to provide estimates of the model uncertainty at a point. For robust control, it is important to obtain estimates of the model uncertainty over *reachable* sets of the system for robust constraint satisfaction.

Contributions

- We propose a non-parametric approach to quantify state and input dependent model uncertainty via semi-definite programming using set-membership techniques. The approach relies on building outer-approximations of the *graph* of the model uncertainty using *side-information* (such as Lipschitz constants, sector bounds) to provide non-parametric model uncertainty estimates at a desired state-input point and also over a set of states and inputs.
- We show to quantify the model uncertainty over the reachable set of a linear system with unmodeled dynamics, and propose a Robust Adaptive MPC algorithm that robustly satisfies constraints and adaptively learns the model uncertainty during execution.

2.2 Problem Formulation

Consider a nonlinear discrete-time system given by the dynamics

$$x_{t+1} = f(x_t, u_t) + d_t,\ d_t \in D(x_t, u) \tag{2.1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system state and input respectively at time t, and $f(\cdot, \cdot)$ is a known, continuous function. The disturbance d_t is assumed to belong to

compact set $D(x_t, u_t)$, where $D(\cdot)$ is a set-valued map, $D : \mathbb{R}^{n+m} \to 2^{\mathbb{R}^n}$. The map $D(\cdot)$ is *unknown* and represents unmodeled dynamics. We assume that this map satisfies the incremental property stated by the following assumption.

Assumption 2.1 *The unknown set-valued map $D(\cdot)$ satisfies the following quadratic constraints, for any $q = (x, u), q' = (x', u')$ in $\mathbb{R}^{n+m}$:*

$$\begin{bmatrix} 1 \\ q - q' \\ d - d' \end{bmatrix}^\top Q \begin{bmatrix} 1 \\ q - q' \\ d - d' \end{bmatrix} \geq 0, \forall d \in D(q), \forall d' \in D(q'), \quad \forall Q \in \boldsymbol{\mathcal{Q}}$$

where $\boldsymbol{\mathcal{Q}} = \{Q^{(1)}, \ldots, Q^{(n_Q)}\}$ is a known, finite set of symmetric matrices. Additionally, $D(z)$ is compact for all $z \in \mathbb{R}^{n+m}$.

Each matrix Q in $\boldsymbol{\mathcal{Q}}$, captures side information on the unmodelled dynamics such as sector bounds, Lipschitz constants or Jacobian bounds [90, 59].

Example 2.1 *Suppose that the disturbance takes the form $d_t = w_t + d(x_t, u_t)$, where w_t lies within the set $\mathcal{W} := \{w | \|w\|_2 \leq \gamma\}$, $d(x, u)$ is an unknown, $L-$Lipschitz continuous function. and so*

$$D(x_t, u_t) = \{d_t \in \mathbb{R}^n \mid \exists w_t \in \mathcal{W}, d_t = w_t + d(x_t, u_t)\}.$$

Then for any $z = (x, u), z' = (x', u')$, $w_1, w_2 \in \mathcal{W}$, we have (using the Lipschitz constant L of $d(\cdot, \cdot)$ and bound γ for w_t) that $||d(z) + w_1 - d(z') - w_2\|_2^2 \leq (L\|z - z'\|_2 + 2\gamma)^2 \leq 2L^2\|z - z'\|_2^2 + 8\gamma^2$. Thus $\boldsymbol{\mathcal{Q}}$ consists of a single matrix $Q = blkdiag([8\gamma^2, 2L^2 I_n, -I_n])$.

Remark 2.1 *The side information in $\boldsymbol{\mathcal{Q}}$ are either statistically estimated, or chosen as hyper-parameters to be tuned. In the former case, without additional assumptions, robust constraint satisfaction can be guaranteed only in probability [81, 85, 102, 73, 131].*

Let the system (4.1) be subject to state and input constraints given by box sets,

$$\mathcal{X} = \{x \in \mathbb{R}^n | lb_x \leq x \leq ub_x\}, \tag{2.2}$$

$$\mathcal{U} = \{u \in \mathbb{R}^m | lb_u \leq u \leq ub_u\}. \tag{2.3}$$

In this chapter, we focus on solving two problems:

Problem 2.1 (Uncertainty Quantification) *Using trajectory data $\{(x_0, u_0), \ldots, (x_T, u_T)\}$ and the side-information from assumption 2.1 to construct bounds on the set-valued map $D(\cdot)$.*

Problem 2.2 (Robust Control) *Design Robust MPC ssing the bounds on the set-valued map $D(\cdot)$ for robust constraint satisfaction $(x_t, u_t) \in \mathcal{X} \times \mathcal{U} \forall \geq 0$.*

2.3 Uncertainty Quantification by Learning Graph Approximations

In this section, we first discuss how to quantify the uncertainty for the true system (4.1) using trajectory data and Assumption 2.1. Then this uncertainty description will be used for constructing tightened constraints for robust constraint satisfaction.

For quantifying the uncertainty $\mathcal{D}$, we use set-membership techniques and state-input trajectory data $\{x_t, u_t\}_{t\geq 0}$ collected from the true system (4.1) to outer-approximate the *graph* $G(D)$ of the set-valued map $D(\cdot)$, defined next.

Definition 2.1 (Graph) *The graph of the set-valued map $D : \mathbb{R}^{n+m} \to 2^{\mathbb{R}^n}$ is defined as the set*

$$G(D) = \{(x, u, d) \mid \forall (x, u) \in \mathbb{R}^{n+m}, \exists d \in D(x, u)\}. \tag{2.4}$$

The graph of $D(\cdot)$ encodes all possible values that the unmodelled system dynamics can attain, but since the set-valued map $D(\cdot)$ is unknown, the set $G(D)$ is unknown as well. However, our main insight is that Assumption (2.1) allows us to outer-approximate $G(D)$ as an intersection of quadratic inequalities (defined using the system data), as stated in the next proposition.

Proposition 2.1 *Given data $\{x_t, u_t\}_{t\geq 0}$ for system (4.1), define $d_t = x_{t+1} - f(x_t, u_t)$, $q_t = (x_t, u_t)$ for all $t \geq 0$. Let Assumption 2.1 hold and consider the set*

$$\mathcal{G}_t = \left\{ (q, d) \,\middle|\, \begin{bmatrix} 1 \\ q - q_t \\ d - d_t \end{bmatrix}^\top Q \begin{bmatrix} 1 \\ q - q_t \\ d - d_t \end{bmatrix} \geq 0,\ \forall Q \in \boldsymbol{\mathcal{Q}} \right\}. \tag{2.5}$$

Then the graph of $D(\cdot)$ satisfies,

$$G(D) \subseteq \bigcap_{t\geq 0} \mathcal{G}_t. \tag{2.6}$$

Thus, the set of realizable disturbances in (4.1) across the state-input space can now be outer-approximated as $\{d \mid \exists (x, u, d) \in \bigcap_{t\geq 0} \mathcal{G}_t\}$. The outer-approximation of the set $D(\bar{x}, \bar{u})$ at a given state-input pair $(\bar{x}, \bar{u})$ is similarly given as $\{d \| (\bar{x}, \bar{u}, d) \in \bigcap_{t\geq 0} \mathcal{G}_t\}$. These characterisations represent a non-parametric description of the uncertainty map $D(\cdot)$ because of its dataset-dependent definition. Figure 2.1 depicts these approximations when the unmodeled map $D(\cdot)$ is given by a Lipschitz continuous function.

We use these non-parametric uncertainty estimates to construct an efficient parametric representation of the map $D(\cdot)$ using semi-definite programming (SDP). In particular, we construct ellipsoidal outer-approximations of the set $D(x, u)$ and also the set:

$$D(\mathcal{X}, \mathcal{U}) := \{d \mid \forall (x, u) \in \mathcal{X} \times \mathcal{U} : \exists (x, u, d) \in D(x, u)\} \tag{2.7}$$

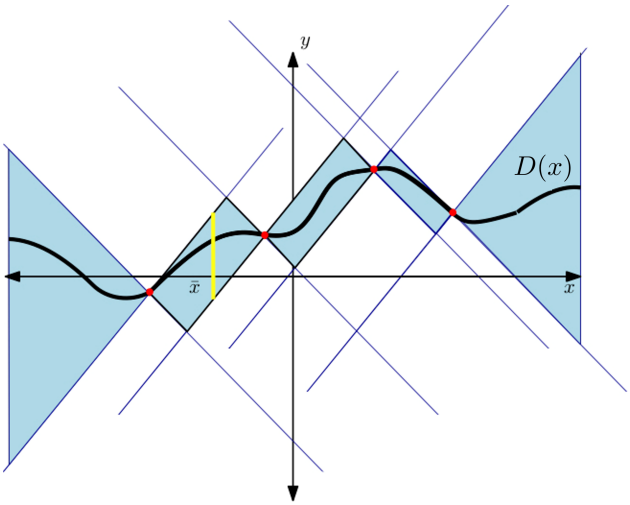


Figure 2.1: Construction of an envelope for a one-dimensional system to approximate the graph $G(D)$ (black curve) of a Lipschitz continuous uncertainty $D(x)$. Tuples $(x_t, D(x_t))$ (red points) obtained from trajectory data are used for constructing the outer-approximation (blue set) of $G(D)$. The yellow set depicts an outer-approximation of $D(\bar{x})$.

Let $E^D(\bar{x},\bar{u})$, $E^D(\mathcal{X},\mathcal{U})$, denote the ellipsoidal sets that outer-approximate $D(x,u)$ and (2.7) respectively. By proposition 2.1, notice that

$$E^D(\mathcal{X},\mathcal{U}) \supseteq \{d \mid \forall (x,u) \in \mathcal{X}\times\mathcal{U} : \exists (x,u,d) \in \bigcap_{t\geq 0} \mathcal{G}_t\} \implies E^D(\mathcal{X},\mathcal{U}) \supset \mathcal{D}(\mathcal{X},\mathcal{U}) \quad (2.8)$$

$$E^D(\bar{x},\bar{u}) \supseteq \{d \mid \exists (\bar{x},\bar{u},d) \in \bigcap_{t\geq 0} \mathcal{G}_t\} \implies E^D(\bar{x},\bar{u}) \supset \mathcal{D}(\bar{x},\bar{u}) \quad (2.9)$$

Notice that the sets $\mathcal{G}_t$, and the expression (2.10) are expressed as quadratic inequalities in $(1,x,u,d)$. So we use the S-procedure to cast the computation of the ellipsoidal sets $E^D(\bar{x},\bar{u})$, $E^D(\mathcal{X},\mathcal{U})$ satisfying the above inclusions as an SDP. To construct $E^D(\mathcal{X},\mathcal{U})$, we will require the following lemma that constructs a quadratic inequality that is satisfied for state-input pairs within constraints.

Lemma 2.1 *For any symmetric matrix $B_q \in \mathbb{R}^{n+m\times n+m}$ such that $B_q \geq 0$, we have that if $q = (x,u) \in \mathcal{X}\times\mathcal{U}$ then*

$$(q - lb_q)^\top B_q (ub_q - q) \geq 0, \quad (2.10)$$

where $ub_q = [ub_x^\top\, ub_u^\top]^\top$ and $lb_q = [lb_x^\top\, lb_u^\top]^\top$.

The variable B_q will provide additional degrees of freedom to minimize the size of the ellipsoidal sets, measured by the sum of its semi-axes. We now state our main result for quantifying the uncertainty $D(\cdot)$ via the ellipsoidal sets $E^D(\bar{x},\bar{u})$, $E^D(\mathcal{X},\mathcal{U})$.

Theorem 2.1 *Let $\{(x_t,u_t,d_t)\}_{t=0}^T$ be the trajectory data of system* (4.1) *satisfying Assumption 2.1.*

1. *Let $\lambda^\star>0, S^\star\succ 0, c^\star$ be the optimal solution of the SDP:*

$$
\begin{aligned}
&\min_{\substack{S,c,\lambda,B_q,\tau_t^l\\ \forall t\in\mathbb{I}_0^T,\ \forall l\in\mathbb{I}_1^{|\mathfrak{Q}|}}} \quad tr(S)\\
&s.t \begin{bmatrix} \lambda\cdot e_1e_1^\top - M_q - \sum_{t=0}^{T}\sum_{l=1}^{|\mathfrak{Q}|}\tau_t^l M_t^l & \begin{matrix}-c^\top\\ 0_n\end{matrix}\\ \star & \begin{matrix}\lambda\cdot I_n\\ S\end{matrix}\end{bmatrix}\succeq 0,\\
&M_q=\begin{bmatrix}-lb_q^\top B_q ub_q & \frac{1}{2}(ub_q+lb_q)^\top B_q & 0_n^\top\\ & -B_q & O_{n+m\times n}\\ \star & & O_n\end{bmatrix},\\
&M_t^l=(I_{2n+m+1}-\begin{bmatrix}0\\ q_t\\ d_t\end{bmatrix}e_1^\top)^\top Q^{(l)}(I_{2n+m+1}-\begin{bmatrix}0\\ q_t\\ d_t\end{bmatrix}e_1^\top),\\
&S\succ 0,\lambda>0,B_q\geq 0, B_q=B_q^\top,\tau_t^l\geq 0,\ \forall t\in\mathbb{I}_0^T, l\in\mathbb{I}_1^{|\mathfrak{Q}|},\\
&\begin{bmatrix}1 & -\tilde{c}_p^\top\\ \star & \tilde{S}_p\end{bmatrix}\succ\begin{bmatrix}\lambda & -c^\top\\ \star & S\end{bmatrix}
\end{aligned}
\tag{2.11}
$$

for any $\tilde{S}_p\succ 0$, $\tilde{c}_p\in\mathbb{R}^n$ that defines the ellipsoid $\mathcal{D}_p=\{d\mid (d-\tilde{c}_p)^\top S_p(d-c_p)\leq 1\}$ such that $D(\mathcal{X},\mathcal{U})\subset\mathcal{D}_p$. Then for $\tilde{S}=\frac{1}{\lambda^\star}S^\star, \tilde{c}=\frac{1}{\lambda^\star}c^\star$, the ellipsoid $E^D(\mathcal{X},\mathcal{U})=\{d\mid (d-\tilde{c})^\top\tilde{S}^{-1}(d-\tilde{c})\leq 1\}$ satisfies

$$D(\mathcal{X},\mathcal{U})\subseteq E^D(\mathcal{X},\mathcal{U})\subset\mathcal{D}_p.$$

2. *For a given $\bar{q} := (\bar{x}, \bar{u})$, let $\lambda_q^\star > 0, S_q^\star \succ 0, c_q^\star$ be the optimal solution of the SDP:*

$$
\begin{aligned}
&\min_{\substack{S_q, c_q, \tau_{tq}^l \\ \forall t \in \mathbb{I}_0^T,\ \forall l \in \mathbb{I}_1^{|\mathfrak{Q}|}}} \quad tr(S_q) \\
&s.t \begin{bmatrix} \lambda \cdot e_1 e_1^\top - \sum_{t=0}^{T} \sum_{l=1}^{|\mathfrak{Q}|} \tau_t^l M_{tq}^l & -c_q^\top \\ \star & S_q \end{bmatrix} \succeq 0, \\
&M_{tq}^l = H^\top Q^{(l)} H + \begin{bmatrix} h(q_t, \bar{q})^\top Q^{(l)} h(q_t, \bar{q}) & -h(q_t, \bar{q})^\top Q^{(l)} H \\ H^\top Q^{(l)} h(q_t, \bar{q}) & O_n \end{bmatrix}, \\
&H = \begin{bmatrix} 1 & 0_n^\top \\ 0_{n+m} & O_{n+m \times n} \\ 0_n & I_n \end{bmatrix}, \ h(q_t, \bar{q}) = \begin{bmatrix} 0 \\ q_t - \bar{q} \\ d_t \end{bmatrix}, \\
&S_q \succ 0, \lambda_q > 0, \tau_{tq}^l \geq 0, \ \forall t \in \mathbb{I}_0^T, l \in \mathbb{I}_1^{|\mathfrak{Q}|}, \\
&\begin{bmatrix} 1 & -\tilde{c}_p^\top \\ \star & \tilde{S}_p \end{bmatrix} \succ \begin{bmatrix} \lambda & -c_q^\top \\ \star & S_q \end{bmatrix}
\end{aligned} \tag{2.12}
$$

Then for $\tilde{S} = \frac{1}{\lambda_q^\star} S_q^\star, \tilde{c} = \frac{1}{\lambda_q^\star} c_q^\star$, the ellipsoid $E^D(\bar{x}, \bar{u}) = \{d \mid (d - \tilde{c})^\top \tilde{S}^{-1} (d - \tilde{c}) \leq 1\}$ satisfies

$$D(\bar{x}, \bar{u}) \subseteq E^D(\bar{x}, \bar{u}) \subset \mathcal{D}_p.$$

The SDPs presented in theorem 2.1 can be solved efficiently using commercial solvers and used online for robust control. The ellipsoid $\mathcal{D}_p$ models any prior bounds on the set $D(\mathcal{X}, \mathcal{U})$. If no such prior is available, then we can set $\mathcal{D}_p = \mathcal{X}$.

Example: Quantifying Drag and Road Grade Uncertainty

Consider the discrete-time longitudinal model of a vehicle, expressed as

$$
\begin{bmatrix} s_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_t \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ dt \end{bmatrix} a_t + \underbrace{d(v_t)}_{drag} + \underbrace{\begin{bmatrix} 0 \\ -g(C_r \cos \alpha_t + \sin \alpha_t) dt \end{bmatrix}}_{rolling\ friction\ +\ gravity} \tag{2.13}
$$

where s_t, v_t, a_t are the station, speed and acceleration of the vehicle at time t, and $dt = 0.1s$ is the discretization time-step. The drag $d(v_t)$ experienced by the vehicle and stochastic road-grade α_t constitute the model uncertainty. We assume that the drag term $d(v_t)$ is $L-$Lipschitz with $L = 0.0046$, the coefficient of rolling-friction is $C_r = 0.0065$ and the road-grade $\alpha_t \leq \Delta = 0.1$. The true evolution of the road-grade is simulated as $\alpha_t = \Delta \cdot sin(10t)$, and the true drag term is modeled as $d(v_t) = [0, \ \frac{1}{2m} \rho C_d A v_t^2]^\top$ where drag coefficient $C_d = 0.4$, air density $\rho = 1.08 kg m^{-3}$, area $A = 0.62 m^2$ and mass $m = 1000 kg$. The set-valued map describing the model uncertainty is given by

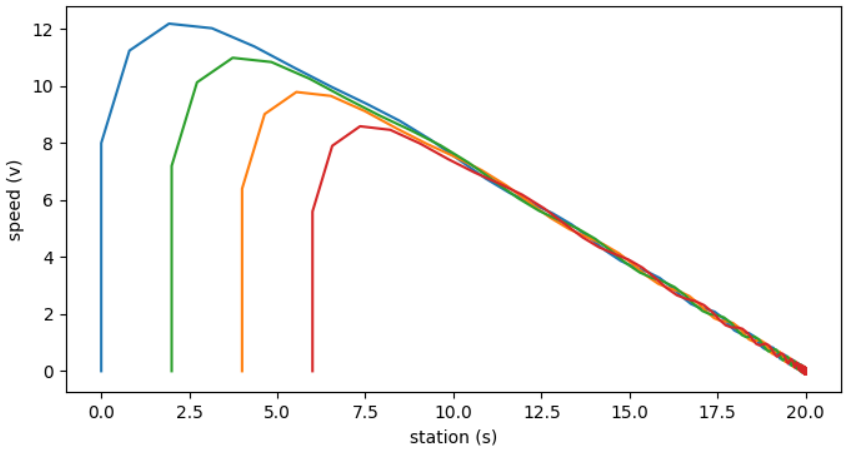


Figure 2.2: Trajectory data of (2.13)

$D(x_t) = \{w \mid \exists z \in [0,0] \times [-gdt(C_r + \sin\Delta), -gdt(C_r - \sin\Delta)] : w = d(v_t) + z\}$. The set of matrices $\mathcal{Q}$ considers the following side-information:

1. $Q^{(1)} = \text{blkdiag}([8\sin\Delta^2, 2L^2 I_2, -I_2])$ to capture Lipchitz drag and bounded uncertainty arising from the road-grade,
2. $Q^{(2)} = ([1, 0_2, 0_2]^\top \times [gdt(\sin\Delta - C_r), 0_2, I_2]) + ([1,_2, 0_2]^\top \times [gdt(\sin\Delta - C_r), O_2, I_2])^\top$ to model the fact that since the drag is always negative, the disturbance is bounded above by $gdt(\sin\Delta - C_r)$.

We estimate the model uncertainty $D(\cdot)$ via the ellipsoid $E^D(\mathcal{X}, \mathcal{U})$ and also demonstrate the uncertainty bounds $E^D(\bar{x}, \bar{u})$ at specific query points $(\bar{x}, \bar{u})$. Our experiment is summarized as follows:

- Trajectories up to a specified time instant N, starting from four different initial conditions are simulated within constraints $\mathcal{X} = [0m, 20m] \times [0ms^{-1}, 14ms^{-1}]$ and $\mathcal{U} = [0ms^{-2}, 2ms^{-2}]$ and stored, as shown below.
- Realizations of the uncertainty $d_t \in D(x_t)$ are recorded via consecutive state measurements (x_t, x_{t+1}).
- Having recorded the measurements (x_t, d_t) for $t = 0, 1, \ldots, N-1$ along all four trajectories, we use theorem 2.1 to construct the estimates $E^D(\mathcal{X}, \mathcal{U})$ and $E^D_N(\bar{x})$ at different speeds $\bar{v}$ and for $N = 50, 400$.

Discussion

The ellipsoid $E^D(\mathcal{X},\mathcal{U}$ is depicted in figure 2.3 which provides an outer-approximation of the true uncertainty set $D(\mathcal{X},\mathcal{U})$.

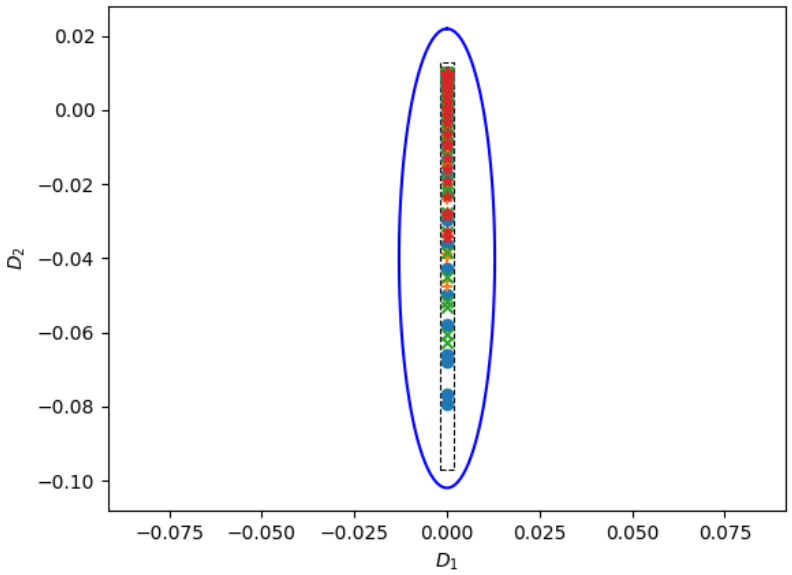


Figure 2.3: Uncertainty ellipsoid $E^D(\mathcal{X},\mathcal{U})$. The true uncertainty bound (calculated within $\mathcal{X},\mathcal{U}$) is given by the black dashed box

Table 2.1: Uncertainty range (up to 3 significant digits). $[\cdot,\cdot]$ denotes an interval

Speed $\bar{v}(m/s)$	$E^D_{50}(\bar{v})$	$E^D_{400}(\bar{v})$
0	$[-0.041, 0.033]$	$[-0.024, 0.020]$
2	$[-0.084, 0.026]$	$[-0.04, 0.01]$
10	$[-0.252, -0.027]$	$[-0.082, -0.041]$
12	$[-0.336, -0.035]$	$[-0.096, -0.059]$
14	$[-0.450, -0.037]$	$[-0.118, -0.038]$

From Table 2.1, we observe that the uncertainty estimates $E^D_N(\bar{x})$ shrink as more data is collected since $E^D_N(\bar{x})$ is obtained with successive intersection operations upon gathering new measurements. Notice that the speed $14ms^{-1}$ does not appear in the trajectory data from figure 2.2. Approximating the graph $D(x)$ allows us to extrapolate but at the expense of a larger uncertainty estimate.

2.4 Robust Adaptive Model Predictive Control

We will now propose a Robust MPC scheme for robust constraint satisfaction, with data-driven adaptation of the model uncertainty using the graph approximation approach from the preceding section. We restrict our attention to the following linear system,

$$x_{t+1} = Ax_t + Bu_t + d(x_t) \tag{2.14}$$

where the function $d(\cdot)$ is $L-$Lipschitz. We aim to solve the following Robust MPC problem at each time t, in a receding horizon fashion:

$$\begin{aligned}
\min_{u_{t|t}, u_{t+1|t}(\cdot), \ldots} \quad & \sum_{k=t}^{t+N-1} (\bar{x}_{k|t}^\top Q \bar{x}_{k|t} + u_{k|t}^\top(\bar{x}_{k|t}) R u_{k|t}(\bar{x}_{k|t})) \\
& \qquad + \bar{x}_{t+N|t}^\top P_N \bar{x}_{t+N|t} \\
\text{s.t.} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t}(x_{k|t}) + d(x_{k|t}), \\
& \bar{x}_{k+1|t} = A\bar{x}_{k|t} + Bu_{k|t}(\bar{x}_{k|t}) + \bar{d}(\bar{x}_{k|t}), \\
& lb_x \leq x_{k+1|t} \leq ub_x, \ \forall d(x_{k|t}) \in E^D(x_{k|t}), \\
& lb_u \leq u_{k|t}(x_{k|t}) \leq ub_u, \ \forall d(x_{k|t}) \in E^D(x_{k|t}), \forall k \in \mathbb{I}_t^{t+N-1}, \\
& x_{t+N|t} \in \mathcal{X}_N, \\
& x_{t|t} = \bar{x}_{t|t} = x_t,
\end{aligned} \tag{2.15}$$

where $x_{k|t}$ is the predicted state after applying the predicted policy $u_{t|t}(x_{t|t}), \ldots, u_{k-1|t}(x_{k-1|t})$ for $k = \{t+1, \ldots, t+N\}$ to system (2.14), $\mathcal{X}_N$ is the terminal set and $P_N \succ 0$ is the terminal cost. $E^D(x_{k|t})$ is the state-dependent uncertainty estimate that can be obtained using the graph approximation approach from section 2.3, and $\bar{d}(\bar{x}_{k|t}) \in E^D(\bar{x}_{k|t})$ is a nominal disturbance estimate. We proceed to address the three crucial challenges associated with finding solutions of (2.15):

(i) Obtaining uncertainty estimates $E^D(x_{k|t})$ for state predictions $x_{k|t}$ which are decision variables in the MPC problem

(ii) Obtaining tractable parametrization of input policy $u(\cdot)$ to avoid searching over infinite dimensional function spaces, and

(iii) Ensuring robust satisfaction of constraints for all times, if tractable reformulation of (2.15) is feasible once.

Uncertainty Estimates Along the MPC Horizon

Definition 2.2 *Robust Controllable States: The* 1-Step Robust Controllable States *from any*

set $\mathcal{A}$ is defined as

$$\begin{aligned}\text{Succ}(\mathcal{A},\mathcal{W}) :=& \{x^+ \in \mathcal{X} : \exists x \in \mathcal{A}, \exists u \in \mathcal{U}, \exists w \in \mathcal{W},\\ &\text{s.t. } x^+ = Ax + Bu + w\},\end{aligned}$$

with state-inputs constraints $\mathcal{X}$,$\mathcal{U}$ defined in (2.2)*.*

Given any state x_t, we can use an S-procedure based approach (similar to section 2.3) to obtain an ellipsoidal outer approximation $E^D(x_t)$. We then successively obtain ellipsoidal outer-approximations for uncertainty sets $E^D(\mathcal{X}_{k|t})$, that is, $E^D(\mathcal{X}_{k|t}) \supseteq \cup_{x_{k|t}\in\mathcal{X}_{k|t}} d(\mathcal{X}_{k|t})$, with

$$E^D(\mathcal{X}_{k|t}) = \bigcup_{x_{k|t}\in\mathcal{X}_{k|t}} E^D(x_{k|t}),$$

where

$$\begin{aligned}&\mathcal{X}_{k|t} \supseteq \text{Succ}(\mathcal{X}_{k-1|t}, E^D(\mathcal{X}_{k-1|t})), &\text{(2.16a)}\\ &\forall k = t+1, t+2, \ldots, t+N,\\ &\mathcal{X}_{t|t} = x_t,\ \mathcal{X}_{t+N|t} = \mathcal{X}. &\text{(2.16b)}\end{aligned}$$

Let sets $E^D(\mathcal{X}_{k|t})$ for any $k = \{t, t+1, \ldots, t+N\}$ be

$$\begin{aligned}E^D(\mathcal{X}_{k|t}) &:= \{d : (d - p^d_{k|t})^\top q^d_{k|t}(d - p^d_{k|t}) \le 1\},\\ &:= \begin{bmatrix} d \\ 1 \end{bmatrix}^\top \bar{P}^d_{k|t} \begin{bmatrix} d \\ 1 \end{bmatrix} \le 0, \end{aligned} \tag{2.17}$$

with $\bar{P}^d_{k|t} = \begin{bmatrix} q^d_{k|t} & -q^d_{k|t}p^d_{k|t} \\ -(p^d_{k|t})^\top q^d_{k|t} & (p^d_{k|t})^\top q^d_{k|t}(p^d_{k|t}) - 1 \end{bmatrix}$, and center $p^d_{k|t} \in \mathbb{R}^n$ and positive definite shape matrix $q^d_{k|t} \in \mathbb{S}^n_{++}$ are decision variables. We consider parametrizations of sets $\mathcal{X}_{k|t}$ as

$$\begin{aligned}\mathcal{X}_{k|t} &:= \{x \in \mathbb{R}^n : (x - p^x_{k|t})^\top q^x_{k|t}(x - p^x_{k|t}) \le 1\},\\ &:= \begin{bmatrix} x \\ 1 \end{bmatrix}^\top \bar{P}^x_{k|t} \begin{bmatrix} x \\ 1 \end{bmatrix} \le 0, \end{aligned} \tag{2.18}$$

where $\bar{P}^x_{k|t} = \begin{bmatrix} q^x_{k|t} & -q^x_{k|t}p^x_{k|t} \\ -(p^x_{k|t})^\top q^x_{k|t} & (p^x_{k|t})^\top q^x_{k|t}(p^x_{k|t}) - 1 \end{bmatrix}$ for any $k = \{t, t+1, \ldots, t+N\}$. Center $p^x_{k|t} \in \mathbb{R}^n$ and shape matrix $q^x_{k|t} \in \mathbb{S}^n_{++}$ can be successively chosen satisfying (2.16a), with $p^x_{t|t} = x_t$ and $q^x_{t|t} = \text{diag}(\infty, \ldots, \infty) \in \mathbb{S}^n_{++}$, if sets $E^D(\mathcal{X}_{k|t})$ are found.

Proposition 2.2 *Using s-procedure, $E^d(\mathcal{X}_{k|t})$ is obtained if the following holds true for some scalars $\{\rho_t^k, \tau_0^k, \tau_1^k, \ldots, \tau_{t-1}^k\} \geq 0$ at each $k = \{t, t+1, \ldots, t+N\}$, for all times $t \geq 0$:*

$$\begin{bmatrix} -\rho_t^k q_{k|t}^x & 0 & \rho_t^k q_{k|t}^x p_{k|t}^x \\ 0 & q_{k|t}^d & -q_{k|t}^d p_{k|t}^d \\ \rho_t^k (p_{k|t}^x)^\top q_{k|t}^x & -(p_{k|t}^d)^\top q_{k|t}^d & (p_{k|t}^d)^\top q_{k|t}^d (p_{k|t}^d) - 1 \\ & & +\rho_t^k - \rho_t^k (p_{k|t}^x)^\top q_t^x (p_{k|t}^x) \end{bmatrix} - \sum_{i=0}^{t-1} \tau_i^k Q_L^d(x_i) \preceq 0. \tag{2.19}$$

We reformulate the feasibility problem (2.19) as a SDP in the Appendix. After finding $E^D(\mathcal{X}_{k|t})$ using (2.19), to efficiently compute (2.16a), we use polytopic outer-approximations $\mathcal{P}^d(\mathcal{X}_{k|t}) \supseteq E^D(\mathcal{X}_{k|t})$ instead of $E^D(\mathcal{X}_{k|t})$, given by

$$\begin{aligned} &\mathcal{P}^d(\mathcal{X}_{k|t}) := \{d : H_{k|t}^d d \leq h_{k|t}^d\}, \\ &\forall k = \{t, t+1, \ldots, t+N\}. \end{aligned} \tag{2.20}$$

The choice of this polytope is designer specific.

Remark 2.2 *Consider the state $x_{k|t}$ for prediction step k at time t in* (2.15)*. From Proposition 2.2 we know that $d(x_{k|t}) \in E^D(\mathcal{X}_{k|t}) \Rightarrow d(x_{k|t}) \in \mathcal{P}^d(\mathcal{X}_{k|t})$, but $d(x_{k|t}) \in \mathcal{P}^d(\mathcal{X}_{k|t}) \not\Rightarrow d(x_{k|t}) \in E^D(\mathcal{X}_{k|t})$. As a consequence, $\mathcal{P}^d(\mathcal{X}_{k|t}) \nsubseteq \mathcal{P}^d(\mathcal{X}_{k|t-1})$ is possible. Hence, for ensuring recursive feasibility of solved MPC problem (detailed in Theorem 2.2), we impose constraints in* (2.15) *robustly for all $d(x_{k|t})$ satisfying*

$$\begin{aligned} &d(x_{k|t}) \in \mathcal{P}^d(\mathcal{X}_{k|t}) \cap \mathcal{P}^d(\mathcal{X}_{k|t-1}), \\ &\forall k \in \{t, \ldots, t+N-1\}, \end{aligned} \tag{2.21}$$

with the initialization $\{q_{-1|-1}^d, q_{0|-1}^d, \ldots, q_{N-2|-1}^d\} = \{0_{n\times n}, \ldots, 0_{n\times n}\} \in \mathbb{R}^{n \times Nn}$.

Control Policy Parametrization

We restrict ourselves to the affine disturbance feedback parametrization [54] for control synthesis in (2.15). For all $k \in \{t, \cdots, t+N-1\}$ over the MPC horizon (of length N), the control policy is given as:

$$u_{k|t}(x_{k|t}) = \sum_{l=t}^{k-1} M_{k,l|t} d(x_{l|t}) + v_{k|t}, \tag{2.22}$$

where $M_{k|t}$ are the *planned* feedback gains at time t and $v_{k|t}$ are the auxiliary inputs. Let us define $\mathbf{d}(x_t) = [d(x_{t|t}), \cdots, d(x_{t+N-1|t})]^\top \in \mathbb{R}^{nN}$. Then the sequence of predicted

inputs from (2.22) can be compactly written as $\mathbf{u}_t = \mathbf{M}_t \mathbf{d}(x_t) + \mathbf{v}_t$ at any time t, where $\mathbf{M}_t \in \mathbb{R}^{mN \times nN}$ and $\mathbf{v}_t \in \mathbb{R}^{mN}$ are

$$\mathbf{M}_t = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ M_{t+1,t} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ M_{t+N-1,t} & \cdots & M_{t+N-1,t+N-2} & 0 \end{bmatrix},$$
$$\mathbf{v}_t = [v_{t|t}^\top, \cdots, \cdots, v_{t+N-1|t}^\top]^\top.$$

Terminal Conditions

We use state feedback to construct terminal set $\mathcal{X}_N$, which is the maximal robust positive invariant set [21] obtained with a state feedback controller $u = Kx$, dynamics (2.14) and constraints (2.2). This set has the properties

$$\begin{aligned} &\mathcal{X}_N \subseteq \{x | lb_x \leq x \leq ub_x,\ lb_u \leq Kx \leq ub_u\}, \\ &(A + BK)x + d \in \mathcal{X}_N, \\ &\forall x \in \mathcal{X}_N,\ \forall d \in \mathcal{P}^d(\mathcal{X}). \end{aligned} \tag{2.23}$$

Fixed point iteration algorithms to numerically compute (2.23) can be found in [21].

Tractable MPC Problem

The tractable MPC optimization problem at time t is given by:

$$\begin{aligned} \min_{\mathbf{M}_t, \mathbf{v}_t} \quad & \sum_{k=t}^{t+N-1} (\bar{x}_{k|t}^\top Q \bar{x}_{k|t} + v_{k|t}^\top R v_{k|t}) + \bar{x}_{t+N|t}^\top P_N \bar{x}_{t+N|t} \\ \text{s.t} \quad & x_{k+1|t} = A x_{k|t} + B u_{k|t}(x_{k|t}) + d(x_{k|t}), \\ & \bar{x}_{k+1|t} = A \bar{x}_{k|t} + B v_{k|t} + \bar{d}_{k|t}, \\ & u_{k|t}(x_{k|t}) = \sum_{l=t}^{k-1} M_{k,l|t} d(x_{l|t}) + v_{k|t}, \\ & H_x x_{k+1|t} \leq h_x, \\ & H_u u_{k|t}(x_{k|t}) \leq h_u, \\ & \forall d(x_{k|t}) \in \mathcal{P}^d(\mathcal{X}_{k|t}) \cap \mathcal{P}^d(\mathcal{X}_{k|t-1}),\ \ \forall k \in \mathbb{I}_t^{t+N-1}, \\ & x_{t+N|t} \in \mathcal{X}_N,\ d(x_{N|t}) \in \mathcal{P}^d(\mathcal{X}), \\ & x_{t|t} = x_t,\ \bar{x}_{t|t} = x_t,\ \bar{d}_{k|t} \in \mathcal{P}^d(\mathcal{X}_{k|t}), \end{aligned} \tag{2.24}$$

where $x_{k|t}$ is the predicted state after applying the policy $[u_{t|t}(x_{t|t}), \ldots, u_{k-1|t}(x_{k-1|t})]$ for $k = \{t+1, \ldots, t+N\}$ to system (2.14), and the control invariant [19] terminal set is $\mathcal{X}_N$.

The parameters $\{p^d_{k|t}, q^d_{k|t}\}$ for $k = \{t, t+1, \ldots, t+N\}$, that is, uncertainty containment ellipses in (2.24), are computed before solving (2.24) at each time t, by finding solutions of (2.19). Nominal uncertainty estimate $\bar{d}_{k|t}$ is chosen as the Chebyshev center (i.e, center of the largest volume ℓ_2 ball in a set) of $\mathcal{P}^d(\mathcal{X}_{k|t})$. After solving (2.24) at time t, in closed-loop we apply

$$u_t(x_t) = v^\star_{t|t}, \tag{2.25}$$

to system (2.14) and then resolve (2.24) at $t+1$.

Remark 2.3 *Terminal set $\mathcal{X}_N$ might be empty initially, due to conservatism resulting from a large volume of the set $\mathcal{P}^d(\mathcal{X})$. As more data is collected and the graph of $d(\cdot)$ is refined as in* (2.6)*, $E^d(\mathcal{X})$, and so $\mathcal{P}^d(\mathcal{X})$ is refined with new data by solving* (2.19) *(for only $k = t+N$, if data collected until instant t) with an updated $Q^d_L(\cdot)$. This eventually results in a nonempty $\mathcal{X}_N$. Once* (2.24) *is feasible with this $\mathcal{X}_N$, during the control process one may further update and enlarge $\mathcal{X}_N$ to lower conservatism of* (2.24)*.*

Algorithm 1: Robust Adaptive MPC

Initialize: $\mathcal{P}^d(\mathcal{X}) = \mathbb{R}^n$; $j = 0$;

begin exploration (offline)

while $\mathcal{X}_N$ is empty **do**
  Apply exploration inputs u_j to (2.14). Collect $(x_j, d(x_j))$ at $j+1$. Set $j = j+1$;
  Solve (2.19) with $k = j+N$ to get $\mathcal{P}^d(\mathcal{X})$.
  Compute $\mathcal{X}_N$ from (2.23);
end while

end exploration **set** $j_{\max} \equiv t = 0$.

begin control process (online)

while during control for $t \geq 0$ **do**
  Obtain $\mathcal{P}^d(\mathcal{X}_{k|t})$ for $k = \{t, t+1, \ldots, t+N-1\}$ from feasibility of (2.19);
  if larger $\mathcal{X}_N$ desired **then**
    Update $\mathcal{P}^d(\mathcal{X})$ from (2.19) (with $k = t+N$).
    Update $\mathcal{X}_N$ from (2.23);
  end if
  Solve (2.24) and apply MPC control (2.25) to (2.14);
end while

Theorem 2.2 *Let optimization problem* (2.24) *be feasible at time* $t = 0$. *Assume the state dependent uncertainty* $d(\cdot)$ *bounds along the horizon are obtained using* (2.19), (2.16), *and* (2.20). *Then,* (2.24) *remains feasible at all times* $t \geq 0$, *if the state* x_t *is obtained by applying the closed-loop MPC control law* (2.25) *to system* (2.14).

Numerical Example

In this section we demonstrate both the aspects of exploration and robust control of our robust Adaptive MPC, highlighted in Algorithm 1. We wish to compute feasible solutions to the following infinite horizon control problem

$$\begin{aligned} \min_{u_0, u_1(\cdot), \ldots} \quad & \sum_{t \geq 0} \bar{x}_t^\top Q \bar{x}_t + u_t^\top(\bar{x}_t) R u_t(\bar{x}_t) \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t(x_t) + 0.05 \begin{bmatrix} \tan^{-1}(x_t(1)) \\ x_t(2) \end{bmatrix} \\ & \begin{bmatrix} -1 \\ -1 \\ -4 \end{bmatrix} \leq \begin{bmatrix} x_t \\ u_t(x_t) \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 3 \\ 1 \end{bmatrix}, \ (\mathcal{X} \times \mathcal{U}) \\ & \forall d(x_t) \in E^D(x_t), \\ & x_0 = \bar{x}_0 = x_S, \ t = 0, 1, \ldots, \end{aligned} \tag{2.26}$$

with initial state $x_S = [-1, 2]^\top$, where $A = \begin{bmatrix} 1.2 & 1.5 \\ 0 & 1.3 \end{bmatrix}$ and $B = [0, 1]^\top$. Algorithm 1 is implemented with a control horizon of $N = 3$, and the feedback gain K in (2.23) is chosen to be the optimal LQR gain for system $x^+ = (A + BK)x$ with $Q = 10I_2$ and $R = 2$.

Exploration for Uncertainty Learning

We initialize $\mathcal{P}^d(\mathcal{X}) = \mathbb{R}^n$, resulting in an empty terminal set $\mathcal{X}_N$ in (2.24). In this section, we present the ability of Algorithm 1 to explore the state-space with randomly generated inputs $u_j \sim \mathcal{N}(0, 1)$, in order to eventually obtain a nonempty $\mathcal{X}_N$ for starting the control process. Let the time indices during exploration phase be denoted by j.

Fig. 2.4 shows the sets $E^d(x)$ at four fixed *query* points $x_j = \{$[-1,2], [1,1], [-1,1], [-2,-1]$\}$ as data is collected until instant j. This can be obtained from feasibility of (2.19) (with $k = j$). As j increases, $E^D(x)$ for each x is contained in the successive intersections of ellipsoids, and consequently, the intersection shrinks for all points. This is seen in Fig. 2.4, which indicates improved information of $E^D(x)$ with added data, for all $x \in \mathcal{X}$. At $j_{\max} = 30$, a nonempty $\mathcal{X}_N$ is obtained, shown in Fig. 2.5. This is when we start control and set $t = 0$.

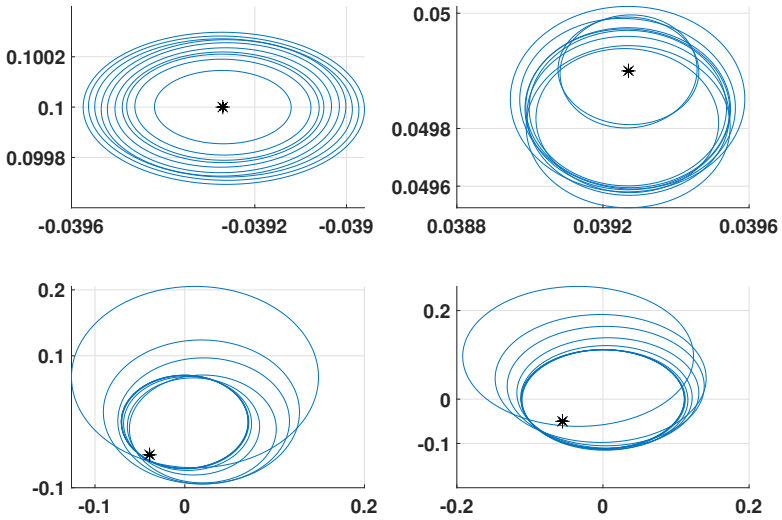


Figure 2.4: Uncertainty bound $E^D(x)$ estimation at *query* points with successive intersection of ellipses obtained from measured data. Star ($\star$) denotes the true value of $d(x)$, lying in the intersection.

Robust Constraint Satisfaction

If the MPC problem (2.24) is feasible for parameters defined in (2.26), it ensures robust satisfaction of constraints in (2.26) for all times $t > 0$. This is highlighted with a realized trajectory in Fig. 2.6. Furthermore, the terminal set is recomputed and improved at a $t > 0$ with (2.23), having refined $\mathcal{P}^d(\mathcal{X})$ estimation[1] from (2.19) (with $k = t + N$). The set grows, as seen in Fig. 2.5, resulting in lesser conservatism of (2.24).

2.5 Conclusions

We proposed a framework to quantify state-dependent model uncertainty $D(\cdot)$ from trajectory data and additional side-information by approximating the graph $G(D)$. We show how to derive global bounds on uncertainty $E^D(\mathcal{X},\mathcal{U}) \supseteq D(\mathcal{X},\mathcal{U})$, point estimates $E^D(\bar{x},\bar{u}) \supseteq D(\bar{x},\bar{u})$, as well as uncertainty estimates $E^D(\mathcal{X}_{k|t})$ over reachable sets of the system. We use the latter to propose a Robust Adaptive MPC algorithm to achieve robust satisfaction of state and input constraints for uncertain linear systems. The uncertainty estimate is adaptively refined with data as the system explores the state space. Upon collection of sufficient data, the system is able to solve a robust MPC problem for

[1] rectangles with sides of length equal to major and minor axes of $E^d(\cdot)$

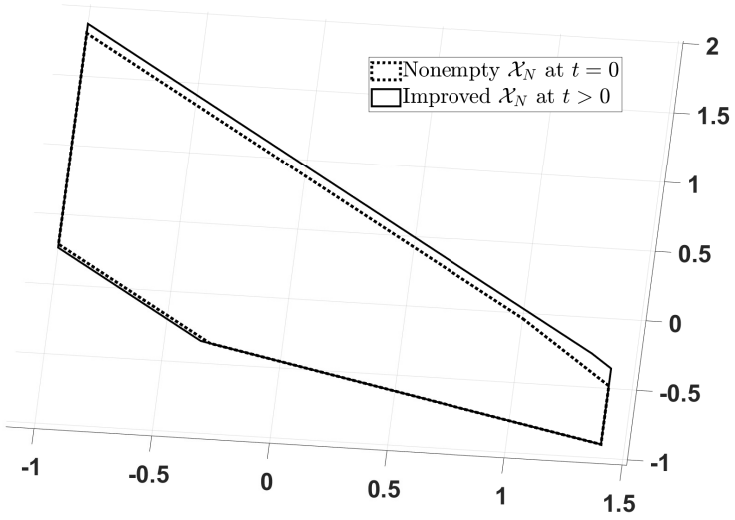


Figure 2.5: Terminal set construction. The set grows as estimation of $d(x)$ is improved from measurements.

all times from a given initial state. The algorithm further reduces its conservatism by incorporating online model adaptation during control.

2.6 Appendix

Proof of Proposition 2.1

By Assumption 2.1, we know that for any $q, q' \in \mathbb{R}^{n+m}$, $z \in D(q), z' \in D(q')$, we have the incremental inequalities

$$\begin{bmatrix} 1 \\ q - q' \\ z - z' \end{bmatrix}^\top Q^{(j)} \begin{bmatrix} 1 \\ q - q' \\ z - z' \end{bmatrix} \geq 0, \ \forall Q^{(j)} \in \boldsymbol{\mathcal{Q}},$$

which also holds for $(q, z), (q', z') \in G(D)$ by Definition 2.1. Now define the set $\mathcal{G}_t^i$ using the incremental inequalities and $(q_t^i, d_t^i) \in G(D)$ as in (2.5). Observe that for any $(q, z) \in G(D)$, we have $(q, z) \in \mathcal{G}_t^i$. Thus, $G(D) \subseteq \mathcal{G}_t^i$. Since this holds for any iteration i and time t for system (4.1), we have $G(D) \subseteq \mathcal{G}_t^i$, $\forall t \geq 0, \forall i \leq j - 1$ which further implies $G(D) \subseteq \cap_{i=0}^{j-1} \cap_{t \geq 0} \mathcal{G}_t^i$ ■

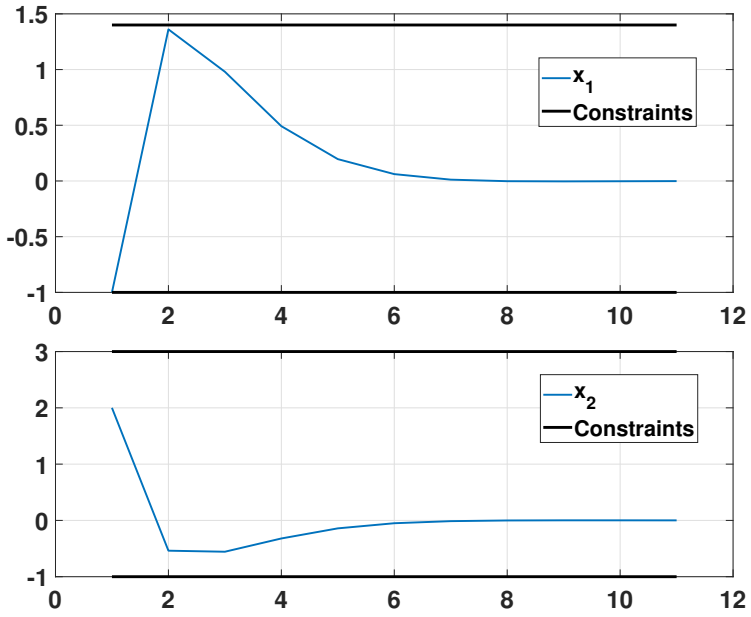


Figure 2.6: State trajectory with robust constraint satisfaction.

Proof of Lemma 2.1

For any $q = (x, u) \in \mathcal{X} \times \mathcal{U}$, we have $lb_q \leq q \leq ub_q$. For any matrix B_q with non-negative entries, the quantity $(q - lb_q)^\top B_q(ub_q - q) \geq 0$ because all terms $ub_q - q \geq 0, lb_q - q \geq 0, B_q \geq 0$ where the inequalities are implied element-wise. ■

Proof of Theorem 2.1

1. Take the Schur complement of the first LMI in (2.11), substitute for $\mathbf{M}_{11}, M_q, M_t^l$, and multiply from both sides by $[1\ q^\top\ d^\top]$ and $[1\ q^\top\ d^\top]^\top$ to get

$$\lambda^\star \left(1 - (d - \frac{c^\star}{\lambda^\star})^\top \left(\frac{S^\star}{\lambda^\star}\right)^{-1} (d - \frac{c^\star}{\lambda^\star})\right) \succ (q - lb_q)^\top B_q^\star (ub_q - q) + \sum_{t=0}^{T_{max}} \sum_{l=1}^{|\mathcal{Q}|} \tau_t^l \begin{bmatrix} 1 \\ q - q_t \\ d - d_t \end{bmatrix}^\top Q^{(l)} \begin{bmatrix} 1 \\ q - q_t \\ d - d_t \end{bmatrix}.$$

Then $\forall (x, u, d) \in G(D)$, $\forall q = (x, u) \in \mathcal{X} \times \mathcal{U}$, the two terms on the RHS are positive from Proposition 2.1 and Lemma 2.1. For $\lambda^\star > 0$, this in turn implies that we have $(d - \tilde{c})^\top (\tilde{S})^{-1} (d - \tilde{c}) \leq 1$, which proves $D(\mathcal{X}, \mathcal{U}) \subseteq E^D(\mathcal{X}, \mathcal{U})$.

Multiplying the last LMI in (2.11) from both sides by $[1\ d^\top], [1\ d^\top]^\top$ gives $1-(d-\tilde{c}_p)^\top(\tilde{S}_p)^{-1}(z-\tilde{c}_p) \geq \lambda^\star(1-(z-\tilde{c})^\top(\tilde{S})^{-1}(z-\tilde{c})$, which implies $E^D(\mathcal{X},\mathcal{U}) \subset \mathcal{D}_p$.

2. Similar to above, take the Schur complement of the LMI in (2.12), substitute for M^l_{tq}, and multiply from both sides by $[1\ d^\top]$ and $[1\ d^\top]^\top$ to get

$$\lambda^\star_q \left(1-(d-\frac{c^\star_q}{\lambda^\star_q})^\top \left(\frac{S^\star_q}{\lambda^\star_q}\right)^{-1}(d-\frac{c^\star_q}{\lambda^\star_q})\right) \succ \sum_{t=0}^{T}\sum_{l=1}^{|\mathcal{Q}|}\tau^l_{tq}\begin{bmatrix}1\\ \bar{q}-q_t\\ d-d_t\end{bmatrix}^\top Q^{(l)}\begin{bmatrix}1\\ \bar{q}-q_t\\ d-d_t\end{bmatrix}.$$

Then $\forall(\bar{x},\bar{u},d) \in G(D)$ for the given $\bar{q}=(\bar{x},\bar{u})$, the term on the RHS is positive from Proposition 2.1 and Lemma 2.1. For $\lambda^\star_q > 0$, this in turn implies that we have $(d-\tilde{c})^\top(\tilde{S})^{-1}(d-\tilde{c}) \leq 1$, which proves $D(\bar{x},\bar{u}) \subseteq E^D(\bar{x},\bar{u})$.

Multiplying the last LMI in (2.12) from both sides by $[1\ d^\top], [1\ d^\top]^\top$ gives $1-(d-\tilde{c}_p)^\top(\tilde{S}_p)^{-1}(z-\tilde{c}_p) \geq \lambda^\star_q(1-(z-\tilde{c})^\top(\tilde{S})^{-1}(z-\tilde{c})$, which implies $E^D(\bar{x},\bar{u}) \subset \mathcal{D}_p$.

■

Proof of Proposition 2.2

Consider any vector $[x^\top d^\top 1]^\top \in \mathbb{R}^{2n+1}$ such that $x \in E^d(\mathcal{X}_{k|t})$ and $[x^\top d^\top]^\top \in G(d)$. Given that (2.19) is feasible for each prediction instant $k=\{t+1,\dots,t+N\}$ at any time t, we multiply $[x^\top d^\top 1]^\top$ on both sides of (2.19)

$$-\rho^k_t\begin{bmatrix}x\\1\end{bmatrix}^\top \bar{P}^x_{k|t}\begin{bmatrix}x\\1\end{bmatrix}+\begin{bmatrix}d\\1\end{bmatrix}^\top \bar{P}^d_{k|t}\begin{bmatrix}d\\1\end{bmatrix} - \begin{bmatrix}x\\d\\1\end{bmatrix}^\top \sum_{i=0}^{t-1}\tau^k_i Q^d_L(x_i)\begin{bmatrix}x\\d\\1\end{bmatrix} \leq 0,$$

for some $\{\rho^k_t, \tau^k_0,\dots,\tau^k_{t-1}\} \geq 0$, where $\bar{P}^x_{k|t}$ and $\bar{P}^d_{k|t}$ are defined in Section 2.4. Now using Proposition 2.1, (2.18) and (2.21), we can infer $\begin{bmatrix}d\\1\end{bmatrix}^\top \bar{P}^d_{k|t}\begin{bmatrix}d\\1\end{bmatrix} \leq 0$. ■

SDP for Solving (2.19)

For all $k=\{t+1,\dots,t+N\}$, along MPC horizon, let us use the variable nomenclature $p(\mathcal{X}_{k|t}) = -\rho^k_t q^x_{k|t} + \sum_{i=0}^{t-1}\tau^k_i L^2_d I_n$, $q(\mathcal{X}_{k|t}) = \rho^k_t (q^x_{k|t})^\top p^x_{k|t} - \sum_{i=0}^{t-1}\tau^k_i L^2_d x_i$, $r(\mathcal{X}_{k|t}) =$

$-\sum_{i=0}^{t-1}\tau_i^k I_n$, $s(\mathcal{X}_{k|t}) = \sum_{i=0}^{t-1}\tau_i^k d(x_i)$, and $t(\mathcal{X}_{k|t}) = \rho_t^k\Big(1-(p_{k|t}^x)^\top q_{k|t}^x p_{k|t}^x\Big) - \sum_{i=0}^{t-1}\tau_i^k\Big(-L_d^2 x_i^\top x_i + d^\top(x_i)d(x_i)\Big) - 1$. Finding the minimum trace ellipsoid satisfying (2.19) is posed as an SDP [30, Section 11.4] as:

$$\begin{aligned}
\min_{\xi}\quad & \text{trace}((q_{k|t}^d)^{-1}) \\
\text{s.t.}\quad & \begin{bmatrix} p(\mathcal{X}_{k|t}) & O_n & q(\mathcal{X}_{k|t}) & O_n \\ O_n & r(\mathcal{X}_{k|t}) & s(\mathcal{X}_{k|t}) & -I_n \\ q^\top(\mathcal{X}_{k|t}) & s^\top(\mathcal{X}_{k|t}) & t(\mathcal{X}_{k|t}) & (p_{k|t}^d)^\top \\ O_n & -I_n\text{`4}- & p_{k|t}^d & -(q_{k|t}^d)^{-1} \end{bmatrix} \preceq 0, \\
& \rho_t^k \geq 0, \tau_i^k \geq 0,\ q_{k|t}^d \succ 0, \\
& \forall i = 0, 1, \ldots, t-1,
\end{aligned}$$

with $\xi = \{q_{k|t}^d, p_{k|t}^d, \rho_t^k, \tau_0^k, \ldots, \tau_{t-1}^k\}$ and $\mathbf{0} \in \mathbb{R}^{n\times n}$. ■

Proof of Theorem 2.2

Let the optimization problem (2.24) be feasible at time t. Let us denote the corresponding optimal input policies as $[\pi_{t|t}^\star(\cdot), \pi_{t+1|t}^\star(\cdot), \cdots, \pi_{t+N-1|t}^\star(\cdot)]$. Assume the MPC controller $\pi_{t|t}^\star(\cdot)$ is applied to (2.14) in closed-loop and $E^D(\mathcal{X}_{k|t+1})$ for $k = \{t+1, t+2, \ldots, t+N+1\}$ are obtained according to (2.19), (2.20) and (2.16). Consider a candidate policy sequence at the next time instant as:

$$\Pi_{t+1}(\cdot) = [\pi_{t+1|t}^\star(\cdot), \ldots, \pi_{t+N-1|t}^\star(\cdot), Kx_{t+N|t+1}]. \tag{2.27}$$

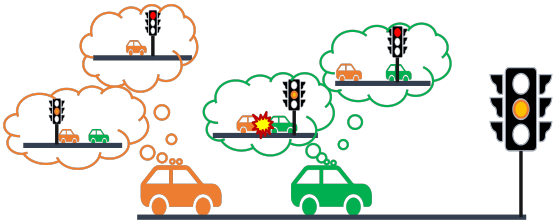
From (2.21) and definitions of $E^D(\mathcal{X}_{k|t})$ and $\mathcal{P}^d(\mathcal{X}_{k|t})$, we conclude that the policy sequence $\pi_{t+1|t}^\star(\cdot)$,$\pi_{t+2|t}^\star(\cdot)$, $\ldots$, $\pi_{t+N-1|t}^\star(\cdot)]$ is an $(N-1)$ step feasible policy sequence at $t+1$ (excluding terminal condition), since at previous time t, it robustly satisfied all stage constraints in (2.24). With this feasible policy sequence, $x_{t+N|t+1} \in \mathcal{X}_N$. From (2.23) we conclude that (2.27) ensures $x_{t+N+1|t+1} \in \mathcal{X}_N$. This concludes the proof. ■

Chapter 3

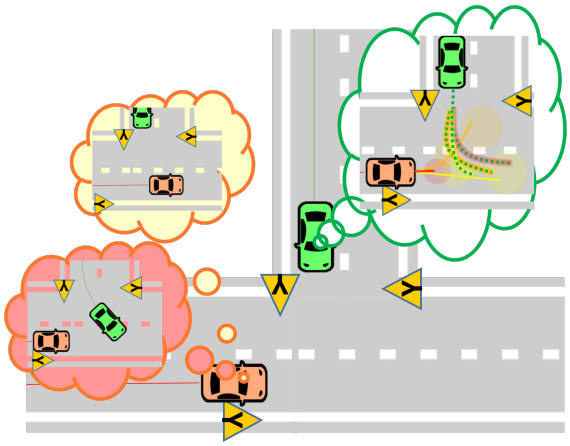
Predictive Control with Uncertain, Multi-modal Predictions of the Environment

3.1 Overview

Autonomous vehicle technologies have seen a surge in popularity over the last decade, making it increasingly common for them to navigate roads with mixed traffic with vehicles of varying automation levels [42]. While existing technology is being gradually introduced into scenarios such as highway driving [36] and low-speed parking [145] where other road users' intents are relatively easy to infer, autonomous driving in mixed traffic scenarios such as urban road driving and merging is an open challenge because of the variability in the possible behaviors of the surrounding agents [3], [138]. To address this difficulty, significant research has been devoted to modeling these agent predictions as multi-modal distributions [20, 37, 118]. Such models capture uncertainty in both high-level decisions (desired route) and low-level executions (agent position, heading, speed).



(a) The AV (green) must decide whether to cross a Traffic Light (TL) before it turns red or come to a stop, while managing the headway for the tailgating TV behind. The discrete modes are given as {TL goes red, TL stays yellow} × {TV stops for TL, TV doesn't stop for TL}. Inspired by the dashcam footage: `https://youtu.be/i3pvrpKDjRQ`.



(b) The AV (green) must find a feasible solution to make a left turn in the presence of an oncoming TV while accounting for its multi-modal behavior: {give AV right-of-way, go straight and turn left}. The AV computes a *policy tree* to address the multi-modal uncertainty. Inspired by the footage: `https://shorturl.at/aeJ12`.

Figure 3.1: Multi-modal planning in urban driving.

The focus of this work is to incorporate these multi-modal distributions for the surrounding agents (called Target Vehicles, or TVs) into a planning framework for the autonomous agent (called Ego Vehicle, or AV). We investigate the planning problem in the context of constrained optimal control and use Model Predictive Control (MPC) for computing feedback control policies. The main challenge in designing MPC for effectively addressing the multi-modal predictions is to find a good balance between *performance*, *safety*, and *computational efficiency*. Consider the situation in Fig. 3.1a, where the AV is approaching a traffic light with a tailgating TV behind. A *performant* MPC design would enable the AV to assess the risk associated with the multi-modalities of the TV and traffic light along the planning horizon so that the AV is able to cross the yellow light or stop at the red light. For ensuring *safety*, the AV must also manage a safe distance ahead of the TV despite the uncertain predictions. A conservative MPC design would either fail to find a feasible solution in the presence of large uncertainty, or sacrifice performance for safety by always choosing to stop. Prior works [13, 39, 101] show that planning using trees or feedback policies over the multi-modal distribution is effective for reliably finding high-quality solutions. Scenario trees [13, 39, 103] offer convenient structure that can be exploited in the SMPC to enhance the feasibility of the SMPC optimization problem by optimize over policies along the prediction horizon. The policies inherit a tree structure from the scenario tree, to encode feedback over the uncertainty

realization. This adds flexibility to find feasible solutions due to the ability to react to different realizations of the vehicles' trajectory predictions along the prediction horizon. Mixture models like Gaussian Mixture Models (GMMs), are more memory-efficient representations for the multi-modal uncertainty by using discrete random variables to capture distinct modalities and continuous variables to capture the spread within each mode. However, optimization over policies that assume feedback from the continuous variables, is infinite-dimensional in general and *computationally expensive* for real-time control. In this work, we propose a Stochastic Model Predictive Control (SMPC) framework that incorporates multi-modal predictions of agents given as GMMs to enforce probabilistic collision avoidance and state-input constraints.

Contributions

- We propose a convex formulation for Stochastic MPC that optimizes over tree-structured feedback policies for multi-modal predictions specified as Gaussian Mixture Models (GMMs).
- The policy parameterization is designed to receive feedback over both discrete modes and continuous observations of the TVs' states. Our formulation also includes a novel multi-modal chance constraint reformulation that simultaneously allocates risk levels for the various modes based on their probabilities.
- We evaluate our approach in various autonomous driving scenarios via simulations and hardware experiments. We demonstrate our SMPC via a hardware experiment for a lane change scenario, characterized by the presence of two TVs with uncertain, bi-modal predictions. Our findings indicate that the proposed approach exhibits a significant reduction in conservatism when compared to the conventional approaches that optimize over open-loop control sequences. Additionally, we show adaptability to variable probabilities of the modes of the TVs.

3.2 Problem Formulation

In this section we formally cast the problem of designing SMPC in the context of autonomous driving.

Preliminaries

AV modeling

We model the dynamics of the AV in the Frenet frame moving along a curve $\gamma(s) = [\bar{X}(s), \bar{Y}(s), \bar{\psi}(s)]$ parameterised by the arc length s, which describes the position and heading of the centerline of a lane in the road [51]. Let $x_t = [s_t,\ e_{y,t},\ e_{\psi,t},\ v_t]^\top$ be the

state of the AV at time t where $s_t, e_{y,t}, e_{\psi,t}$ are the arc length, lateral offset and relative heading with respect to the centerline $\gamma(\cdot)$, and v_t is the AV's speed. Then the dynamics of the AV can be described as

$$\dot{x}_t = \begin{bmatrix} \frac{v_t cos(e_{\psi,t})}{1 - e_{y,t}\kappa(s_t)} \\ v_t \sin(e_{\psi,t}) \\ \dot{\psi}_t - \frac{v_t cos(e_{\psi,t})\kappa(s_t)}{1 - e_{y,t}\kappa(s_t)} \\ a_t \end{bmatrix} \tag{3.1}$$

where $\kappa(s_t) = \frac{d\bar{\psi}(s_t)}{ds}$ describes the curvature of $\gamma(\cdot)$, a_t is the AV's acceleration and $\dot{\psi}_t$ is the AV's global yaw rate. The dynamics of the AV are time-discretized (with any explicit integration scheme) to obtain the model $x_{t+1} = f(x_t, u_t)$, with inputs $u_t = [a_t,\ \dot{\psi}_t]$. Given the state x_t, the AV's global pose can be obtained via a function $\mathcal{G}^\gamma(\cdot)$,

$$\begin{bmatrix} X_t \\ Y_t \\ \psi_t \end{bmatrix} = \mathcal{G}^\gamma(x_t) = \begin{bmatrix} \bar{X}(s_t) - e_{y,t}\sin(\bar{\psi}(s_t)) \\ \bar{Y}(s_t) + e_{y,t}\cos(\bar{\psi}(s_t)) \\ e_{\psi,t} + \bar{\psi}(s_t) \end{bmatrix}. \tag{3.2}$$

The system state and input constraints are given by polytopic sets which capture vehicle actuation limits and traffic rules,

$$\begin{aligned} \mathcal{X} &= \{x : a_{x,i}^\top x \leq b_{x,i}\ \forall i \in \mathbb{I}_1^{n_X}\}, \\ \mathcal{U} &= \{u : a_{u,i}^\top u \leq b_{u,i}\ \forall i \in \mathbb{I}_1^{n_U}\}. \end{aligned} \tag{3.3}$$

We assume a kinematically feasible reference trajectory,

$$\{(x_t^{ref}, u_t^{ref})\}_{t=0}^T \tag{3.4}$$

is provided for the AV. This serves as the AV's desired trajectory which can be computed offline (or online at lower frequency) accounting for the AV's route, actuation limits, and static environment constraints (like lane boundaries, traffic rules). However, this reference does not consider the dynamically evolving TVs for real-time obstacle avoidance.

TV predictions

Let n_{TV} be the number of TVs in consideration and denote the position of the ith TV at time t as $o_t^i = [X_t^i\ Y_t^i]^\top$, and define $o_t = [o_t^{1\top}, \ldots, o_t^{n_{TV}\top}]^\top$ which stacks the positions of all the TVs. For collision avoidance, we use an off-the-shelf prediction model [37, 118] trained on traffic datasets [64, 29] that provides $N-$step predictions of the TVs' positions given by a multi-modal Linear Time-Varying (LTV) model $\forall k \in \mathbb{I}_t^{t+N}, j \in \mathbb{I}_1^J$,

$$o_{k+1|t,j} = T_{k|t,j} o_{k|t,j} + c_{k|t,j} + n_{k|t,j}, \tag{3.5}$$

where $o_{k|t,j}$ is the prediction of the TVs' positions at time k for mode j, $T_{k|t,j}, c_{k|t,j}$ are time-varying matrices and vectors for the TVs' prediction for mode j, and the process noise is given by $n_{k|t,j} \sim \mathcal{N}(0, \Sigma_{k|t,j})$. The mode $j \in \mathbb{I}_1^J$ captures distinct interactions/maneuvers of the TVs as a group. We denote $p_t = [p_{t,1}, .., p_{t,j}]$ as the probability distribution over the modes at time t, and σ_t to be the true, *unknown* mode.

Stochastic Model Predictive Control Formulation

We aim to design a computationally efficient feedback control $u_t = \pi_t(x_t, o_t)$ for the AV to track the reference trajectory (3.4), satisfy state-input constraints and avoid collisions with the TVs by effectively addressing the uncertainty arising from the TVs' multi-modal predictions (3.5).

We propose a Stochastic Model Predictive Control (SMPC) formulation to compute the feedback control u_t. The optimization problem of our SMPC takes the form,

$$
\begin{aligned}
\min_{\{\theta_{t,j}\}_{j=1}^J} \quad & \sum_{j=1}^{J} p_{t,j} \mathbb{E}\left[C_t(\mathbf{x}_{t,j}, \mathbf{u}_{t,j})\right] && \text{(3.6a)}\\
\text{s.t.} \quad & x_{k+1|t,j} = f(x_{k|t,j}, u_{k|t,j}), && \text{(3.6b)}\\
& o_{k+1|t,j} = T_{k|t,j} o_{k|t,j} + c_{k|t,j} + n_{k|t,j}, && \text{(3.6c)}\\
& \mathbb{P}(g_k(x_{k+1|t}, o^i_{k+1|t}) \geq 0) \geq 1 - \epsilon, && \text{(3.6d)}\\
& (x_{k+1|t,j}, u_{k|t,j}) \in \mathcal{X} \times \mathcal{U}, && \text{(3.6e)}\\
& \mathbf{u}_{t,j} \in \Pi_{\theta_{t,j}}(\mathbf{x}_{t,j}, \mathbf{o}_{t,j}), && \text{(3.6f)}\\
& x_{t|t,j} = x_t,\ u_{t|t,j} = u_{t|t,1},\ o_{t|t,j} = o_t, && \text{(3.6g)}\\
& \forall i \in \mathbb{I}_1^{n_{TV}}, \forall j \in \mathbb{I}_1^J, \forall k \in \mathbb{I}_t^{t+N-1}.
\end{aligned}
$$

where $\mathbf{u}_{t,j} = [u_{t|t,j}, \ldots, u_{t+N-1|t,j}]$, $\mathbf{x}_{t,j} = [x_{t|t,j}, \ldots, x_{t+N|t,j}]$ and $\mathbf{o}_{t,j}$ (defined similarly to $\mathbf{x}_{t,j}$), denote stacked predictions along the horizon for mode j. The SMPC feedback control action is obtained from the optimal solution of (3.6) as

$$u_t = \pi_{\text{SMPC}}(x_t, o_t) = u^\star_{t|t,1}, \tag{3.7}$$

where the AV and TV state feedback enters the optimization problem in (3.6g). The function $C_t(\cdot, \cdot)$ in the objective (3.6a) penalizes the deviation of the AV's trajectory for mode j from the reference (3.4), and is weighted by the probability of the mode given by $p_{t,j}$. The collision avoidance constraints are imposed as chance constraints (7.1f) along with polytopic state and input constraints $\mathcal{X}$,$\mathcal{U}$ for the AV. The AV's controls along the prediction horizon are given by parameterized policies $\Pi_{\theta_{t,j}}(\mathbf{x}_{t,j}, \mathbf{o}_{t,j})$ (3.6f) that are functions of the AV's and TVs' states, as opposed to open-loop sequences. The policies are multi-modal, which makes the AV's closed-loop trajectories in (3.6b) multi-modal.

Consequently, the chance constraints (3.6d) are defined over the closed-loop multi-modal distributions, re-written using the law of total probability as:

$$\sum_{j=1}^{J} p_{t,j} \mathbb{P}(g_k(x_{k+1|t,j}, o^i_{k+1|t,j}) \geq 0) \geq 1 - \epsilon.$$

Deriving a deterministic reformulation of this chance constraint that is computationally efficient, but not too conservative is the key technical challenge in the SMPC design. Towards addressing this challenge, our SMPC formulation features 1) a novel multi-modal policy parameterization (3.6f) for shaping the multi-modal closed-loop distribution, 2) a convex inner-approximation technique for the multi-modal chance constraint (3.6d) involving mode-dependent, risk levels $r_{t,j} = 1 - \epsilon_{t,j}$ for $\epsilon_{t,j} \in [0, 1]$ and 3) simultaneous, convex optimization over policy parameters and risk levels for control computation. These features enable computationally efficient synthesis of (3.7) while enhancing the feasibility of (3.6) by effectively addressing the multi-modal uncertainty.

3.3 Stochastic MPC with Multi-Modal Predictions

In this section, we detail our SMPC formulation for the AV to track the reference (3.4) while incorporating multi-modal predictions (3.5) of the TV for obstacle avoidance.

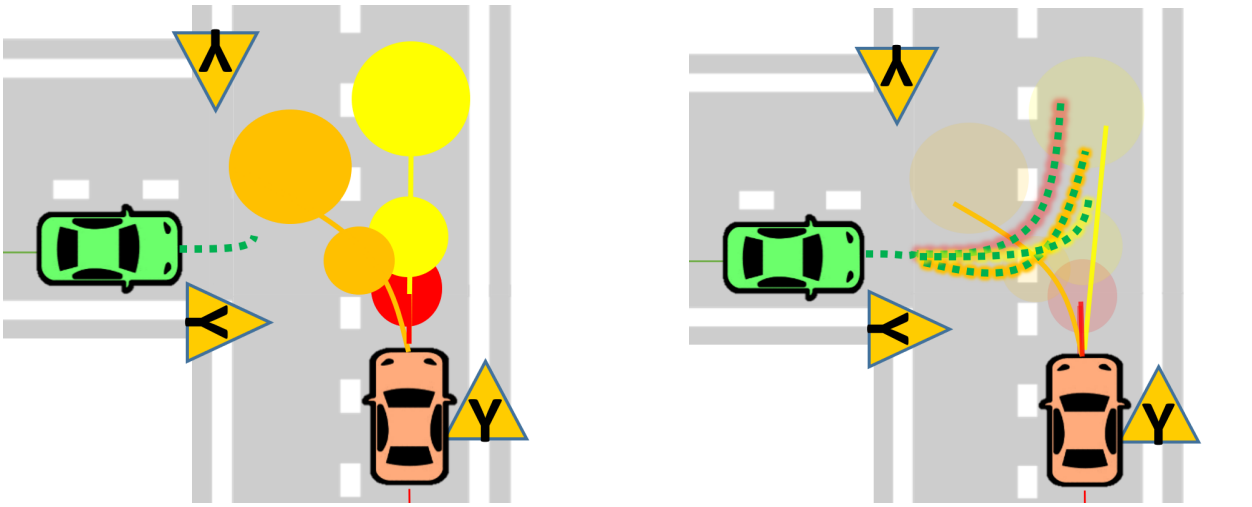
Vehicles' Prediction Models

The AV prediction model (3.6b) is a linear time-varying model (LTV), obtained by linearizing $f(\cdot)$ about the reference trajectory (3.4). At time t, let $\bar{t} = -t + \arg\min_{k \in \mathbb{I}_0^T} |s_t - s_k^{ref}|$ and define $\Delta x_{k|t} = x_{k|t} - x^{ref}_{k+\bar{t}}$, $\Delta u_{k|t} = u_{k|t} - u^{ref}_{k+\bar{t}}, \forall k \in \mathbb{I}_t^{t+N-1}$. Then the LTV model is given as

$$\begin{aligned} &\Delta x_{k+1|t} = A_{k|t} \Delta x_{k|t} + B_{k|t} \Delta u_{k|t} + w_{k|t} \\ &A_{k|t} = \partial_x f(x^{ref}_{k+\bar{t}}, u^{ref}_{k+\bar{t}}),\ B_{k|t} = \partial_u f(x^{ref}_{k+\bar{t}}, u^{ref}_{k+\bar{t}}) \end{aligned} \tag{3.8}$$

where the additive process noise $w_{k|t} \sim \mathcal{N}(0, \Sigma_w)$ (i.i.d with respect to k) models linearization error and other stochastic noise sources. The polytopic state and input constraints (3.6e) are replaced by the chance-constraints $\forall k \in \mathbb{I}_0^{N-1}$,

$$\begin{aligned} &\mathbb{P}((\Delta x_{k+1|t}, \Delta u_{k|t}) \in \Delta\mathcal{X}_k \times \Delta\mathcal{U}_k) \geq 1 - \epsilon, \\ &\Delta\mathcal{X}_k = \{\Delta x : a_{x,i}^\top \Delta x \leq b_{x,i} - a_{x,i}^\top x^{ref}_{\bar{t}+k}, \forall \in \mathbb{I}_1^{n_x}\}, \\ &\Delta\mathcal{U}_k = \{\Delta u : a_{u,i}^\top \Delta x \leq b_{u,i} - a_{u,i}^\top u^{ref}_{\bar{t}+k}, \forall \in \mathbb{I}_1^{n_u}\}. \end{aligned} \tag{3.9}$$



(a) Open-loop predictions, $\mathbf{u}_t \in \mathbb{R}^{2\times N}$ (b) Closed-loop predictions, $\mathbf{u}_t \in \Pi_{\theta_t}(\mathbf{x}_t, \mathbf{o}_t)$

Figure 3.2: In (a), solving (7.1) over open-loop sequences can be conservative because AV prediction (green-dashed) from a single sequence of control inputs must satisfy all the obstacle avoidance constraints. In (b), optimizing over policies (7.1d) allows for different AV predictions depending on the TV trajectory realizations (green-dashed with highlights corresponding to different TV trajectories).

Parameterized AV & TV State Feedback Policies

We propose to use parameterized polices $\Pi_{\theta_t}(\mathbf{x}_t, \mathbf{o}_t)$ so that the AV's control $\mathbf{u}_t$ are functions of the AV and TV trajectories $\mathbf{x}_t, \mathbf{o}_t$ along the prediction horizon (as depicted in Fig. 3.2). Given the AV model (3.8) and mode-dependent TV model (3.5), consider the following feedback policy $\Delta u_{k|t,j} = \pi_{k|t,j}(x_{k|t,j}, o_{k|t,j})$ for the AV :

$$\Delta u_{k|t,j} = h^j_{k|t} + \sum_{l=0}^{k-1} M^j_{l,k|t} w_{l|t} + K^j_{k|t}(o_{k|t,j} - \mu_{k|t,j}), \tag{3.10}$$

where $\mu_{k|t,j} = \mathbb{E}[o_{k|t,j}]$ denotes the expected prediction of the TVs in (3.5). The policy (3.10) uses State Feedback (SF) for the TV states but Affine Disturbance Feedback (ADF) for feedback over AV states (see [54] for equivalence to state feedback) instead of SF. SF policies for the TVs are beneficial towards scaling our approach to multiple TVs because we use $O(N^2 + n_{TV} \cdot N)$ parameters instead of $O(n_{TV} \cdot N^2)$ parameters for ADF.

Despite using SF for the TVs' states, $\Delta x_{k|t}$ are affine in $\{K_{s|t}\}_{s=0}^{k}$ $\forall k \in \mathbb{I}_0^{n-1}$ as shown next.For mode $j \in \mathbb{I}_1^J$, define the stacked quantities along the prediction horizon $\mathbf{\Delta u}_{t,j}, \mathbf{w}_t, \mathbf{o}_{t,j}, \boldsymbol{\mu}_{t,j}$ (e.g., $\mathbf{\Delta u}_{t,j} = [\Delta u^\top_{t|t,j} \dots \Delta u^\top_{t+N-1|t,j}]^\top$) and use (5.6) to get

$$\mathbf{\Delta u}_{t,j} = \mathbf{h}^j_t + \mathbf{M}^j_t \mathbf{w}_t + \mathbf{K}^j_t(\mathbf{o}_{t,j} - \boldsymbol{\mu}_{t,j})$$

using the stacked policy parameters $\mathbf{h}_t^j \in \mathbb{R}^{2N}, \mathbf{M}_t^j \in \mathbb{R}^{2N\times 4N}, \mathbf{K}_t^j \in \mathbb{R}^{2N\times 2N\cdot n_{TV}}$ ((3.24) in appendix 5.6). Denote the stacked EV closed-loop predictions for mode j as $\boldsymbol{\Delta x}_{t,j}$ and the TV's stacked process noise as $\mathbf{n}_{t,j}$. Using matrices $\mathbf{A}_t, \mathbf{B}_t, \mathbf{T}_t^j, \mathbf{C}_t^j, \mathbf{L}_t^j, \mathbf{E}_t$ defined by (3.25), (3.26) in the appendix, the closed-loop EV predictions are

$$\begin{aligned}\boldsymbol{\Delta x}_{t,j} =& \mathbf{A}_t \Delta x_{t|t} + \mathbf{B}_t(\mathbf{h}_t^j + \mathbf{K}_t^j(\mathbf{o}_{t,j} - \boldsymbol{\mu}_{t,j})) + (\mathbf{E}_t + \mathbf{B}_t\mathbf{M}_t^j)\mathbf{w}_t \\ =& \mathbf{A}_t \Delta x_{t|t} + \mathbf{B}_t(\mathbf{h}_t^j + \mathbf{K}_t^j\mathbf{L}_t^j\mathbf{n}_{t,j}) + (\mathbf{E}_t + \mathbf{B}_t\mathbf{M}_t^j)\mathbf{w}_t\end{aligned}$$

which are affine in $\mathbf{h}_t^j, \mathbf{M}_t^j, \mathbf{K}_t^j$. The last equality uses $\boldsymbol{\mu}_{t,j} = \mathbb{E}[\mathbf{o}_{t,j}] = \mathbf{T}_t^j o_t + \mathbf{C}_t^j + \mathbf{L}_t^j \mathbf{n}_{t,j}$ from (3.26).

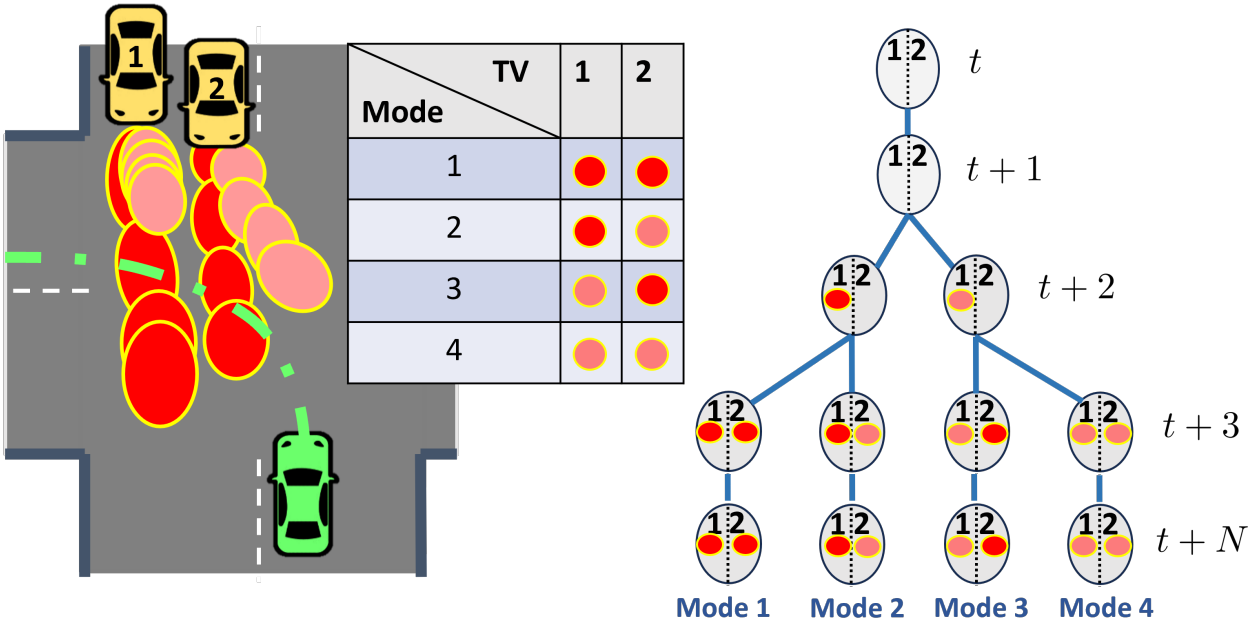


Figure 3.3: Tree for encoding mode feedback along the prediction horizon. TV1's mode is revealed at time-step $t+2$, whereas the TV2's mode is revealed at time $t+3$. Consequently, all the $J = 4$ EV feedback policies share the same parameters in the interval $\mathbb{I}_t^{t+1}$. For time-step $t+2$, parameters are shared by policies $1, 2$ and $3, 4$. After TV2's mode is revealed at time $t+3$, the $J = 4$ policies have independent parameters in the interval $\mathbb{I}_{t+3}^{t+N}$. The policy constraints are described by the set $\mathcal{T}_t = \{(t+1, \{1,2\}), (t+1, \{1,3\}), (t+1, \{1,4\}), (t+2, \{1,2\}), (t+2, \{3,4\})\}$. The intervals for mode 4 are $\mathcal{B}_{t,4} = \{\mathbb{I}_t^{t+1}, \mathbb{I}_{t+2}^{t+2}, \mathbb{I}_{t+3}^{t+N}\}$

The policy parameters $\boldsymbol{\Theta}_t(x_t, o_t) = \{\mathbf{h}_t^j, \mathbf{M}_t^j, \mathbf{K}_t^j\}_{j=1}^J$ define J feedback policies $\boldsymbol{\Delta u}_{t,j} = \mathbf{h}_t^j + \mathbf{M}_t^j \mathbf{w}_t + \mathbf{K}_t^j \mathbf{o}_{t,j}$ that assume feedback from $\mathbf{x}_{t,j}, \mathbf{o}_{t,j}$ along the prediction horizon and the mode $j \in \mathbb{I}_1^J$. To have J independent feedback policies starting from times-step $t+1$,

the mode must be revealed at the time-step $k = t + 1$. To model more complex mode feedback information structures along the horizon, we use the notion of an information tree as illustrated in the Fig. 3.3. Additional constraints are imposed on $\mathbf{\Theta}_t(x_t, o_t)$ incorporate the mode feedback information structure along the prediction horizon, so that (3.22) optimizes over *realizable* policies. For example, if the EV optimizes for J different policies along the prediction horizon but the TVs' mode is revealed at time-step $t + N - 1$, then it is ambiguous whether which of the J policies the EV must use in the interval $\mathbb{I}_{t+1}^{t+N-2}$. In practice, the predictions (3.5) and control (3.7) are computed in a receding horizon fashion, but optimistically optimizing over un-realizable policies incurs infeasibilities when solving (7.1).

To formalize the constraints on $\mathbf{\Theta}_t(x_t, o_t)$, let the mode feedback information structure be defined by the set $\mathcal{T}_t \subset \mathbb{I}_t^{t+N} \times \left(\mathbb{I}_1^J \times \mathbb{I}_1^J\right)$, where $(k, \{j_1, j_2\}) \in \mathcal{T}_t$ corresponds to modes j_1, j_2 being ambiguous in interval $\mathbb{I}_t^k$. Given $(k, \{j_1, j_2\}) \in \mathcal{T}_t$, we constrain the policy parameters of modes j_1, j_2 as:

$$h_{k'|t}^{j_1}, K_{k'|t}^{j_1}, \{M_{l,k'|t}^{j_1}\}_{l=t}^{k'-1} = h_{k'|t}^{j_2}, K_{k'|t}^{j_2}, \{M_{l,k'|t}^{j_2}\}_{l=t}^{k'-1}, \ \forall k' \in \mathbb{I}_t^k$$

so that the policies of modes j_1 and j_2 are same up to time k. For the example in Fig. 3.3, the policy parameterization for mode $j = 4$ and $N = 4$ is given as

$$\begin{aligned}
\mathbf{h}_t^4 &= [h_{t|t}^{1\top}\ h_{t+1|t}^{1\top}\ h_{t+2|t}^{3\top}, h_{t+3|t}^{4\top}, \dots h_{t+N-1|t}^{4\top}]^\top \\
\mathbf{K}_t^4 &= \text{blkdiag}\left(K_{t|t}^1, K_{t+1|t}^1, K_{t+2|t}^3, K_{t+3|t}^4, \dots, K_{t+N-1|t}^4\right) \\
\mathbf{M}_t^4 &= \begin{bmatrix} O & O & O & O \\ M_{t,t+1|t}^1 & O & O & O \\ M_{t,t+2|t}^3 & M_{t+1,t+2|t}^3 & O & O \\ M_{t,t+3|t}^4 & M_{t+1,t+3|t}^4 & M_{t+2,t+3|t}^4 & O \end{bmatrix}
\end{aligned}$$

For each mode j, we denote $\mathcal{B}_{t,j}$ as the set of time intervals between the branch points, defined formally as

$$\mathcal{B}_{t,j} = \left\{ \mathbb{I}_{k_1}^{k_2} \subset \mathbb{I}_t^{t+N} \middle| \begin{matrix} \exists j_1, j_2 : (k_1, \{j, j_1\}), (k_2, \{j, j_2\}) \in \mathcal{T}_t, \\ k_2 = \min(t+N, \min\limits_{k > k_1, (k, \{j,l\}) \in \mathcal{T}_t} k) \end{matrix} \right\}. \tag{3.11}$$

For example, if the modes are revealed at time-step $k = t+1$, we have $\mathcal{T}_t = \{(t, \{1, j\})\ \forall j \in \mathbb{I}_2^J\}$ and $\mathcal{B}_{t,j} = \{\mathbb{I}_t^{t+N}\}\ \forall j \in \mathbb{I}_1^J$. Then for each mode j, we can split the policy matrices as the sums, $\mathbf{M}_t^j = \sum_{b=1}^{|\mathcal{B}_{t,j}|} \bar{\mathbf{M}}_t^{j,b}$, $\mathbf{K}_t^j = \sum_{b=1}^{|\mathcal{B}_{t,j}|} \bar{\mathbf{K}}_t^{j,b}$, where $\bar{\mathbf{M}}_t^{j,b}, \bar{\mathbf{K}}_t^{j,b}$ have the same shape as in (3.24) but consist only of policy parameters corresponding to the interval $b \in \mathcal{B}_{t,j}$. We denote this policy parameterization as $\mathbf{\Theta}_t(x_t, o_t; \mathcal{T}_t)$. The tree structure $\mathcal{T}_t$ is assumed to be inferred from the predictions (3.5) using some measure of distance between distributions (e.g., Wasserstein distance).

Collision Avoidance Formulation

We assume that we are given or can infer the rotation matrices for the ith TV for each mode along the prediction horizon as $\{\{R^i_{k|t,j}\}_{k=1}^N\}_{j=1}^J$. For collision avoidance between the AV and the ith TV, we impose the following chance constraint

$$\sum_{j=1}^{J} p_{t,j}\mathbb{P}(g^i_{k|t,j}(P_{k|t,j}, o^i_{k|t,j}) \geq 1\ ) \geq 1-\epsilon \quad \forall k = t+1,\ldots,t+N \tag{3.12}$$

where the AV's position $P_{k|t,j} = [X_{k|t,j}, Y_{k|t,j}]$ is obtained from $\mathcal{G}^\gamma(\Delta x_{k|t,j} + x^{ref}_{t+k})$, and

$$g^i_{k|t,j}(P,o) = \left\| \begin{bmatrix} \frac{1}{a_{ca}} & 0 \\ 0 & \frac{1}{b_{ca}} \end{bmatrix} R^i_{k|t,j}(P-o) \right\|^2. \tag{3.13}$$

$a_{ca} = a_{TV} + d_{AV}, b_{ca} = b_{TV} + d_{AV}$ are semi-axes of the ellipse containing the TV's extent with a buffer of d_{AV}. $g_{k|t,j}(P,o) \geq 1$ implies that the AV's extent (modeled as a disc of radius d_{AV} and centre P) does not intersect the TV's extent which is modelled as an ellipse with semi-axes a_{TV}, b_{TV} and centre o, oriented by $R_{k|t,j}$. This constraint is non-convex because of the integral of the nonlinear function $g^i_{k|t,j}(\cdot)$ over the multi-modal distribution of $(P_{k|t}, o_{k|t}, p_t)$. To address the nonlinearity, we use the convexity of $g^i_{k|t,j}(\cdot)$ to construct its affine under-approximation $l^i_{k+1|t,j}(\cdot)$ by defining $P^i_{k|t,j} = \mu_{k|t,j} + \frac{1}{\sqrt{g^i_{k|t,j}(P^{ref}_{k|t}, \mu^i_{k|t,j})}}(P^{ref}_{k|t} - \mu^i_{k|t,j})$ to get:

$$\begin{aligned} l^i_{k|t,j}(P,o) =& \partial_P g^i_{k|t}(P^i_{k|t,j}, \mu^i_{k|t,j})(P - P^i_{k|t,j}) && (3.14)\\ &+ \partial_o g^i_{k|t}(P^i_{k|t,j}, \mu^i_{k|t,j})(o - \mu^i_{k|t,j}). && (3.15)\end{aligned}$$

Proposition 1 *For the affine function $l^i_{k|t,j}(P,o)$ defined in* (3.14)*, the multi-modal chance constraint can be inner-approximated $\forall k = t+1,\ldots,t+N$ as*

$$\begin{aligned} &\sum_{j=1}^{J} p_{t,j}\mathbb{P}(l^i_{k|t,j}(P_{k|t,j}, o^i_{k|t,j}) \geq 0\ ) \geq 1-\epsilon && (3.16)\\ \Rightarrow &\sum_{j=1}^{J} p_{t,j}\mathbb{P}(g^i_{k|t,j}(P_{k|t,j}, o_{k|t,j}) \geq 1) \geq 1-\epsilon \end{aligned}$$

The multi-modal, affine chance constraints for collision avoidance are given as

$$\sum_{j=1}^{J} p_{t,j}\mathbb{P}\left[l^i_{k+1|t,j}(P_{k+1|t}, o^i_{k+1|t,j}) \geq 0\right] \geq 1-\epsilon. \tag{3.17}$$

We define the curve $\gamma(\cdot)$ using piece-wise linear segments so that $\mathcal{G}^\gamma(\Delta x_{k|t} + x^{ref}_{t+k}) = \mathcal{G}^\gamma(x^{ref}_{t+k}) + \partial_x\mathcal{G}^\gamma(x^{ref}_{t+k})\Delta x_{k|t}$, and the constraint (3.17) is affine in the policy parameters $\mathbf{\Theta}_t(x_t, o_t; \mathcal{T}_t)$. Next, we discuss the reformulation of the multi-modal affine chance constraints (3.9), (3.17).

Reformulation of Multi-modal Chance Constraints

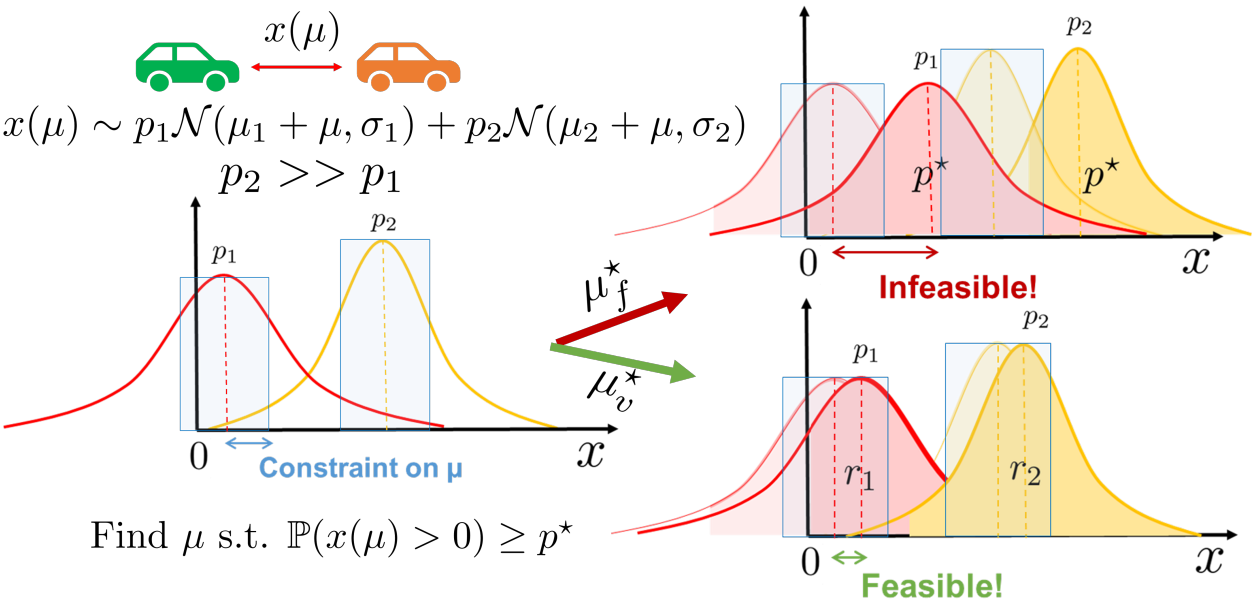


Figure 3.4: For $x(\mu)$ given by a GMM on the left (e.g., predicted distance from the lead vehicle), the figure depicts the fixed-risk (top-right) and variable-risk (bottom-right) formulations for shaping $x(\mu)$ for satisfying the constraint $\mathbb{P}(x(\mu) > 0) \geq p^\star$ by optimizing μ within the blue feasible region (e.g., dictated by actuation constraints). The variable-risk formulation can choose a smaller $\mu = \mu_v^\star$ within constraints by exploiting the difference in mode probabilities– since $p_2 \gg p_1$, assigning a larger risk level r_2 (shaded yellow region) to mode 2 yields feasible distributions. The fixed-risk formulation assigns $r_1 = r_2 = p^\star$ but the resulting $\mu = \mu_f^\star$ violates the constraints.

We propose a novel convex inner-approximation for multi-modal chance constraints, with the key feature of simultaneous risk allocation for reduced conservatism. The constraints (3.9), (3.17) can be generically represented as the multi-modal affine chance constraint:

$$\begin{aligned}
&\sum_{j=1}^{J} p_{t,j}\mathbb{P}\left[\begin{matrix} a_0 + a_1^\top \mathbf{h}_t^j + (a_2^\top \mathbf{M}_t^j + a_3^\top)\mathbf{w}_t \\ + (a_4^\top \mathbf{K}_t^j + a_5^\top)\mathbf{L}_t^j \mathbf{n}_t \end{matrix} \geq b\right] \geq 1-\epsilon \\
\Leftrightarrow &\sum_{j=1}^{J} p_{t,j} r_{t,j} \geq 1-\epsilon, \qquad (3.18a) \\
&\mathbb{P}\left[\begin{matrix} a_0 + a_1^\top \mathbf{h}_t^j + (a_2^\top \mathbf{M}_t^j + a_3^\top)\mathbf{w}_t \\ + (a_4^\top \mathbf{K}_t^j + a_5^\top)\mathbf{L}_t^j \mathbf{n}_t \end{matrix} \geq b\right] \geq r_{t,j},
\end{aligned}$$

$$\Leftrightarrow a_0 + a_1^\top \mathbf{h}_t^j - b \geq \Phi^{-1}(r_{t,j}) \left\| \begin{bmatrix} \boldsymbol{\Sigma}_t (\mathbf{M}_t^{j\top} a_2 + a_3) \\ \boldsymbol{\Sigma}_{t,j} \mathbf{L}_t^{j\top} (\mathbf{K}_t^{j\top} a_4 + a_5) \end{bmatrix} \right\|_2 \tag{3.18b}$$

where $\Phi^{-1}(\cdot)$ is the quantile function for $\mathcal{N}(0,1)$, and $r_{t,j}$ are the risk-levels $\forall j \in \mathbb{I}_1^J$. We assume that $1-\epsilon \geq \frac{1}{2}$ and $r_{t,j} \geq \frac{1}{2}$, $\forall j \in \mathbb{I}_1^J$. As such, this constraint is non-convex in the policy parameters $\boldsymbol{\Theta}_t(x_t, o_t; \mathcal{T}_t)$ and risk-levels $r_{t,j}$ because of (3.18b). The fixed-risk allocation approach [146, 101] fixes the risk-level $r_{t,j} = 1-\epsilon \ \forall j \in \mathbb{I}_1^J$ to obtain a convex-inner approximation of the multi-modal chance constraint. However as depicted in Fig. 3.4, this approach can be conservative and compromises the feasibility of the SMPC optimization problem because the risk levels are allocated disregarding the probability of the individual modes. Alternatively, iterative solution strategies [104], have been proposed for variable risk allocation where, alternating sub-problems are solved by fixing either the policy parameters or risk levels. This enhances the feasibility of the optimization problem but at the price of significant computational cost. Next, we propose a convex-inner approximation to the multi-modal chance constraint (3.18a), (3.18b) for simultaneous risk-allocation and policy synthesis, to alleviate the computational cost of iterative approaches, but also improve the feasibility of constraint compared to the fixed allocation approach.

First, we focus on reformulating (3.18b). We recall that the policy parameters can be split as $\mathbf{M}_t^j = \sum_{b=1}^{|\mathcal{B}_{t,j}|} \bar{\mathbf{M}}_t^{j,b}$, $\mathbf{K}_t^j = \sum_{b=1}^{|\mathcal{B}_{t,j}|} \bar{\mathbf{K}}_t^{j,b}$, and begin by introducing new variables

$$\begin{aligned} &\eta_{t,j} = \Phi^{-1}(r_{t,j}), \\ &\tilde{\mathbf{M}}_t^j = \Phi^{-1}(r_{t,j}) \bar{\mathbf{M}}_t^{j,|\mathcal{B}_{t,j}|}, \ \tilde{\mathbf{K}}_t^j = \Phi^{-1}(r_{t,j}) \bar{\mathbf{K}}_t^{j,|\mathcal{B}_{t,j}|} \end{aligned}$$

to rewrite the constraint (3.18b) as

$$\begin{aligned} &a_0 + a_1^\top \mathbf{h}_t^j - b \geq \\ &\left\| \begin{bmatrix} \boldsymbol{\Sigma}_t (\tilde{\mathbf{M}}_t^{j\top} a_2 + \eta_{t,j} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{M}}_t^{j,b\top} a_2 + a_3 \eta_{t,j}) \\ \boldsymbol{\Sigma}_{t,j} \mathbf{L}_t^{j\top} (\tilde{\mathbf{K}}_t^{j\top} a_4 + \eta_{t,j} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{K}}_t^{j,b\top} a_4 + a_5 \eta_{t,j}) \end{bmatrix} \right\|_2 \end{aligned} \tag{3.19}$$

The variable $\eta_{t,j}$ can be interpreted as the number of standard deviations by which the affine constraint is tightened. Since $r_{t,j} \geq \frac{1}{2}$, we have $\eta_{t,j} \geq 0$, and additionally, let $\eta_{t,j} \leq \eta_{max}$ to ignore the tail of the Gaussian distribution. The inequality (3.19) is non-convex in the new variables because of the bilinear terms $\eta_{t,j} \bar{\mathbf{M}}_t^{j,b\top}$, $\eta_{t,j} \bar{\mathbf{K}}_t^{j,b\top}$. However, fixing either variable in these terms yields a convex, second-order cone constraint. We use this insight to obtain a convex inner-approximation of this non-convex inequality as follows.

Proposition 2 *A convex inner approximation of* (3.19) *can be obtained as the intersection of the two second-order cone constraints in the variables* $\{\mathbf{h}_t^j, \tilde{\mathbf{M}}_t^j, \tilde{\mathbf{K}}_t^j, \{\bar{\mathbf{M}}_t^{j,b}, \bar{\mathbf{K}}_t^{j,b}\}_{b=1}^{|\mathcal{B}_{t,j}|-1}, \eta_{t,j}\}$*:*

$$
\begin{aligned}
&a_0 + a_1^\top \mathbf{h}_t^j - b \geq \left\| \begin{bmatrix} \boldsymbol{\Sigma}_t (\tilde{\mathbf{M}}_t^{j\top} a_2 + a_3 \eta_{t,j}) \\ \boldsymbol{\Sigma}_{t,j} \mathbf{L}_t^{j\top} (\tilde{\mathbf{K}}_t^{j\top} a_4 + a_5 \eta_{t,j}) \end{bmatrix} \right\|_2, \\
&a_0 + a_1^\top \mathbf{h}_t^j - b \geq \\
&\left\| \begin{bmatrix} \boldsymbol{\Sigma}_t (\tilde{\mathbf{M}}_t^{j\top} a_2 + \eta_{max} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{M}}_t^{j,b\top} a_2 + a_3 \eta_{t,j}) \\ \boldsymbol{\Sigma}_{t,j} \mathbf{L}_t^{j\top} (\tilde{\mathbf{K}}_t^{j\top} a_4 + \eta_{max} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{K}}_t^{j,b\top} a_4 + a_5 \eta_{t,j}) \end{bmatrix} \right\|_2
\end{aligned} \tag{3.20}
$$

With the new variable definitions, constraint (3.18a) takes the form $\sum_{j=1}^{J} p_{t,j} \Phi(\eta_{t,j}) \geq 1 - \epsilon$, which is convex in $\eta_{t,j}$, but difficult to enforce since $\Phi(\cdot)$ lacks a closed-form expression. For $\eta \in [0, \eta_{max}]$, we approximate $\Phi(\eta)$ by a concave function $\Psi(\eta) = \min_{i=1,..,\nu}\{q_i^1 \eta + q_i^0\}$ such that $\Phi(\eta) \geq \Psi(\eta)$, and replace (3.18a) with the convex inner-approximation:

$$
\sum_{j=1}^{J} p_{t,j} \Psi(\eta_{t,j}) \geq 1 - \epsilon \tag{3.21}
$$

A candidate approximation of $\Phi(\cdot)$ over $[0, 2]$ with $\nu = 2$ affine functions is shown in Fig. 3.5 Thus, the non-convex constraints (3.18a), (3.18b) can be replaced by the con-

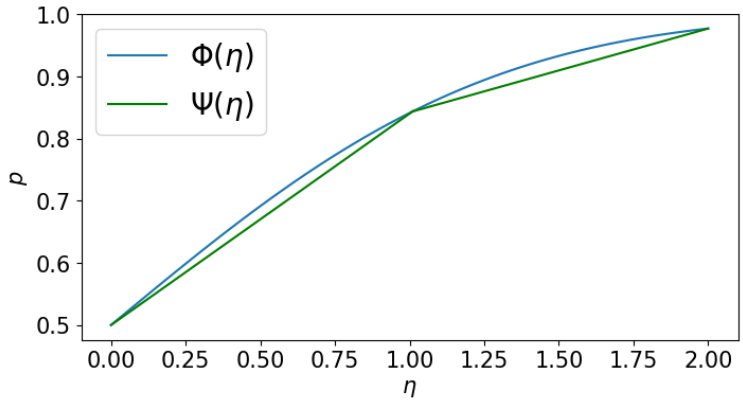


Figure 3.5: $\Psi(\eta)$ as a concave under-approximation of $\Phi(\eta)$

vex inner-approximations provided by the Second-Order Cone (SOC) constraints (3.21), (3.20).

SMPC Optimization Problem

The cost (7.1a) of the SMPC optimization problem (7.1) is chosen as a convex quadratic function to penalise deviations of the AV state and input trajectories from the reference

trajectory (3.4) as

$$C_t(\mathbf{x}_t, \mathbf{u}_t) = \sum_{k=0}^{N-1} \Delta x_{k+1|t}^\top Q \Delta x_{k+1|t} + \Delta u_{k|t}^\top R \Delta u_{k|t}$$

where $Q, R \succ 0$. Let $\mathbf{\Theta}_t^{\text{MPC}}(x_t, o_t; \mathcal{T}_t)$ denote the set of policy parameters and risk levels,

$$\boldsymbol{\theta}_t := \{\mathbf{h}_t^j, \tilde{\mathbf{M}}_t^j, \tilde{\mathbf{K}}_t^j, \{\bar{\mathbf{M}}_t^{j,b}, \bar{\mathbf{K}}_t^{j,b}\}_{b=1}^{|\mathcal{B}_{t,j}|-1}, \eta_{t,j}\}_{j=1}^J$$

such that they satisfy

1. the SOC reformulations (3.20), (3.21) of the multi-modal state-input and collision avoidance constraints (3.9), (3.17),
2. structural constraints given by $\mathcal{T}_t$.

We use the single-shooting/batch formulation for the optimal control problem using the linearized dynamics (3.8) so that the only decision variables in the optimization problem are the policy parameters $\boldsymbol{\theta}_t$. Since the state-input predictions $(\mathbf{\Delta x}_{t,j}, \mathbf{\Delta u}_{t,j})$ are affine in the policy parameters $\boldsymbol{\theta}_t$, the cost function $C_t(\cdot)$ is quadratic in $\boldsymbol{\theta}_t$.

Proposition 3 *The SMPC* (3.7) *for the AV can be synthesized by solving the second-order cone program:*

$$\begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \sum_{j=1}^{J} p_{t,j} \mathbb{E}\left[C_t(\mathbf{x}_{t,j}, \mathbf{u}_{t,j})\right] \\ s.t\ & \boldsymbol{\theta}_t \in \mathbf{\Theta}_t^{MPC}(x_t, o_t; \mathcal{T}_t) \end{aligned} \tag{3.22}$$

3.4 Numerical Validation

In this section, we investigate the proposed algorithm, both qualitatively and quantitatively. To assess the benefits of our proposed stochastic MPC formulation, we demonstrate our approach in three different scenarios: (A) *Longitudinal control with a traffic light and a following vehicle*, (B) *Unprotected left turn at an intersection*, and (C) *Lane change on a straight road*. In scenario (A), we validate the proposed algorithm in a simple 1-D simulation and show the qualitative behavior of our SMPC in managing the multi-modal predictions. In scenarios (B) and (C), we use CARLA [45] for the simulator and adopt the motion predictor MultiPath [37] to predict the multi-modal future motions of surrounding vehicles. In these scenarios, we provide a quantitative study of our approach against baselines.

Qualitative analysis: Longitudinal control with a traffic light and a following vehicle

Setup

Consider the situation in Fig.3.1a, where the AV is approaching a Traffic Light (TL) with a tailgating TV behind. All vehicles and the traffic light are simulated in a simple 1D simulator. Both vehicles are modelled as double integrators with Euler discretization ($@dt = 0.1s$) as follows:

$$\begin{aligned}
&x_t = \begin{bmatrix} s_t & v_t \end{bmatrix}^\top,\ u_t = a_t, \\
&x_{t+1} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0.5dt^2 \\ dt \end{bmatrix} u_t, \\
&o_t = \begin{bmatrix} s_t^o & v_t^o \end{bmatrix}^\top,\ u_t^o = a_t^o, \\
&o_{t+1} = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} o_t + \begin{bmatrix} 0.5dt^2 \\ dt \end{bmatrix} u_t^o + w_t^o, \\
&w_t^o \sim \mathcal{N}(0, 0.6I_2),
\end{aligned} \tag{3.23}$$

where each state comprises the longitudinal position and speed, each input is the acceleration, and w_t^o is an additive process noise in the TV dynamics. For our simulation, the initial states of the AV and the TV are set to $x_0 = [0,\ 13.9]$, $o_0 = [-12.75, 14]$ so that 1) TV has a $0.7s$ time headway behind the AV, 2) there is enough distance for the AV to brake at $0.7g$ and stop at the TL, which is located $50m$ ahead of the AV, i.e., $s_f = 50m$.

The AV is subject to state-input constraints $\mathcal{X} \times \mathcal{U} = \{(x, u) : v \in [v_{min}, v_{max}], a \in [a_{min}, a_{max}]\}$ and collision avoidance constraints $\mathcal{C} = \{(x, o) : s - o \geq d_{safe}\}$, where $v_{min} = 0m/s$, $v_{max} = 14m/s$, $a_{min} = -7m/s^2$, $a_{max} = 4m/s^2$, $d_{safe} = 7m$.

For simplicity, we assume that the driver in the TV has good decision-making skills based on the previous observations. In particular, we assume the following:

- When the TV's driver is confident that the TL will remain yellow until the TV crosses with a probability of 1, the TV will maintain its speed.
- When the TV's driver is not confident that the TL will remain yellow, they will choose to brake. Here, we assume the probability of either red or yellow light is 0.5 conditioned on the TV choosing to brake.
- The TV's driver will make a decision when the TV passes a certain point, i.e., $s_t^o \geq s_{dec}$.

Under this assumption, there are three possible modes:

- mode 0: TV keeps speed, TL stays yellow,
- mode 1: TV brakes, TL stays yellow,

- mode 2: TV brakes, TL goes red before AV crosses.

The AV does not know the true mode $\sigma \in \{0, 1, 2\}$. The AV estimates the probabilities of each mode as $p_t = [p_{t,0}, p_{t,1}, p_{t,2}]$ using Bayes' rule via observations of the TV's state history $o_0, o_1, ..o_t$ after crossing s_{dec}. The tree structure $\mathcal{T}_t$ for the predictions is determined by rolling out the TV's acceleration commands and branching into the three modes based on when it crosses s_{dec}.

The chance constraints are imposed with risk level $\epsilon = 0.01$. For deriving the variable-risk reformulations of multi-modal chance constraints (3.9),(3.17) as shown in Section 3.3, we use $\Psi(\eta)$ as depicted in Fig. 3.5 as the concave under-approximation of the CDF $\Phi(\eta)$ over $\eta \in [0, 2]$. For stopping at s_f in *mode* 2, we enforce the terminal constraint $\mathcal{X}_f = \{x = [s, v] \in \mathcal{X} : v^2 \leq -2a_{min}(s_f - s)\}$.

The SMPC cost is given as $C_t(\mathbf{x}_{t,j}, \mathbf{u}_{t,j}) = \sum_{k=t}^{t+N} -Qs_{k+1|t,j} + Ru_{k|t,j}^2$ over prediction horizon $N = 12$ with $Q = 10, R = 20$ to penalize slow progress and control effort.

Simulations

We run two different simulation scenarios, where true modes are $\sigma = 0$ and $\sigma = 2$, respectively. Note that the AV does not know the true mode and estimates the mode after TV crosses the decision point $s_t^o \geq s_{dec}$. We compare the proposed SMPC in (3.22) (*Proposed* in Fig. 3.6) with an open-loop approach (*OL* in Fig. 3.6), where the AV solves an SMPC with fixed-risk levels for each mode, and optimizes over a single open-loop sequence $\mathbf{h}_t$ i.e., the gains $K_{k|t}^j, M_{l,k|t}^j$ in (5.6) are eliminated. The results are illustrated in Fig. 3.6.

Discussion

As depicted in Fig. 3.6, the open-loop approach is unable to exploit the mode probabilities and is infeasible throughout the simulation, leading to a collision with the TV. In contrast, the proposed approach accordingly accelerates to cross the yellow TL in the first realization, while it decelerates to a stop in the second realization (without knowing the true mode σ) after the TV crosses the decision point $s_t^o \geq s_{dec}$ (around $t = 30$ sec). This closed-loop behavior results from the estimation of the mode probabilities and the incorporation of the multi-modal probability estimates in the chance constraints as in (3.21).

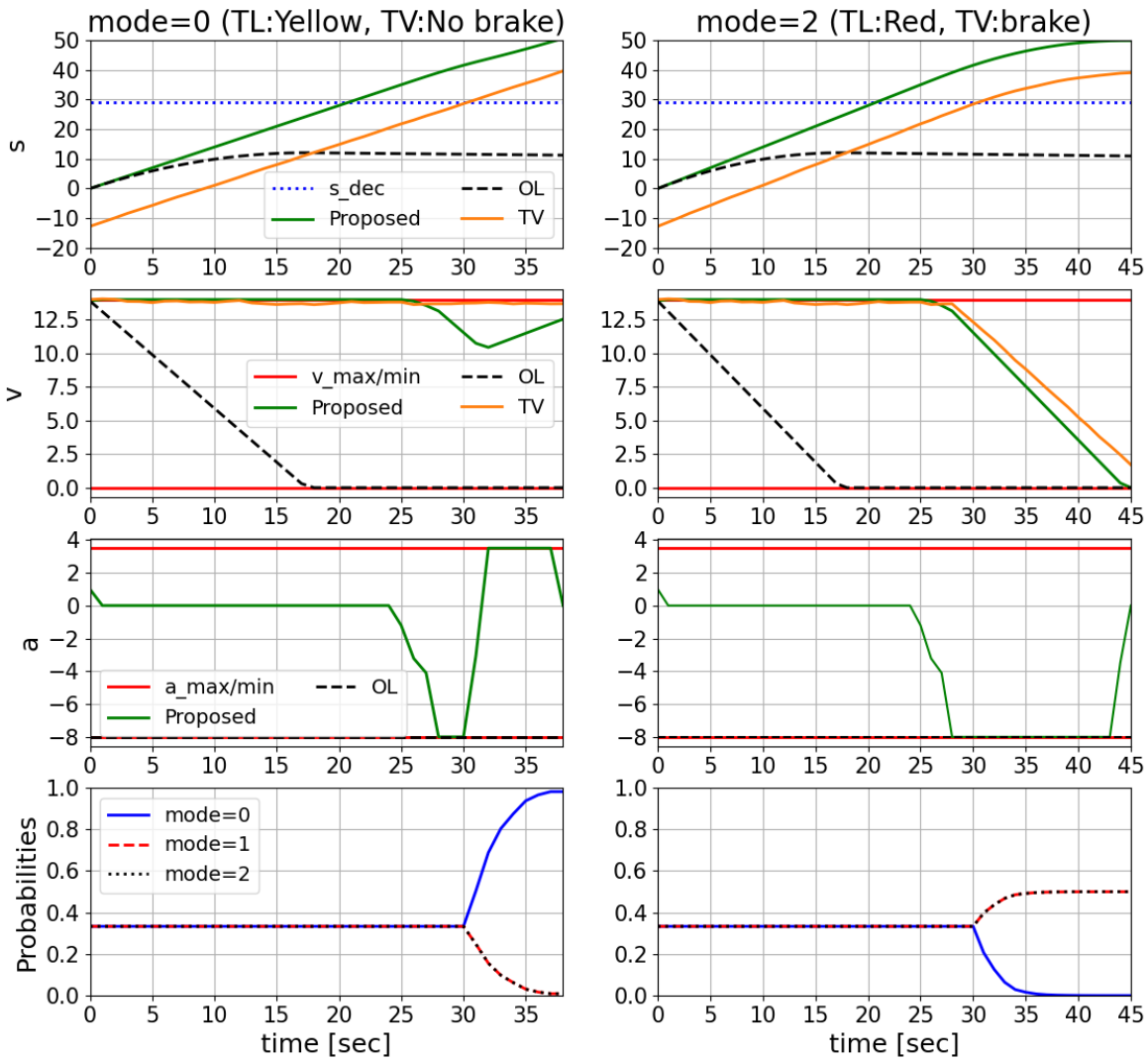


Figure 3.6: Closed-loop plots for the longitudinal control example for modes $\sigma = 0, 2$ by solving (3.22) for the SMPC. *Proposed* is our proposed approach and *OL* is the open-loop approach. The traffic light (TL) is located at $s_f = 50m$. The AV is unaware of the true mode σ, but estimates the mode probabilities from TV observations. Using the proposed approach, the AV is able to cross the TL for $\sigma = 0$ and safely stop before the TL for $\sigma = 2$. Code: `https://github.com/shn66/AV_SMPC_Demos/tree/TL_eg`.

Table 3.1: Closed-loop performance comparison across all scenarios.

Scenario	Policy	Safety		Solver Performance
		$\mathcal{F}$ (%)	$\bar{d}_{min}$ (m)	$\bar{T}_{solve}$ (ms)
Unprotected left	**OL**	98.37	3.09	31.5
	Fixed risk	98.37	3.09	31.5
	Proposed	**99.88**	3.07	39.9
Lane change	**OL**	82.55	3.42	**35.6**
	Fixed risk	96.08	3.21	325.20
	Proposed	**98.76**	3.19	397.19

Quantitative analysis: Unprotected left turn and Lane change with surrounding vehicles

(a) Unprotected left (b) Lane change

Figure 3.7: Carla simulation setup for unprotected left turn and lane change for AV (green) in the presence of TVs (orange) and multi-modal predictions (depicted by the ellipses). Code: `https://github.com/shn66/SMPC_MMPreds`

Setup

Consider the two scenarios as depicted in Fig. 3.7:

- *Unprotected left*: the AV makes a left turn through the intersection while avoiding an oncoming TV.
- *Lane change*: the AV changes into the left lane in the presence of two TVs: One ahead in the same lane and another behind in the adjacent lane.

All vehicles are simulated in a synchronous fashion in CARLA, ensuring that all processing (prediction, planning, and control) is complete before advancing the simulation.

Thus, our results only consider the impact of the control and not the time delays incurred in computation.

The TVs' controls are given by a simple nonlinear MPC to go straight, with a kinematic bicycle model for predictions and simple distance-based collision avoidance constraints. From the AV's perspective, the predictions of the TV's motion are given by a GMM for multi-modal trajectory predictions of the form (3.5). We obtain this multi-modal distribution along with mode probabilities online using the MultiPath prediction model from [37]. Given the predicted multimodal distributions for the TVs, we use the framework introduced in Section 3.3 to generate feedback control policies, and use Fig. 3.5 for the CDF approximation. A dynamically feasible AV reference trajectory (3.4) is obtained by solving a nonlinear trajectory optimization problem for a kinematic bicycle using IPOPT [132] to track a high-level route (provided by the CARLA waypoint API). Given the AV reference (3.4) and TV predictions (3.5), the SMPC optimization problem (3.22) is solved using Gurobi [57] to compute the acceleration and steering controls.

Policies

We evaluate and compare the following set of policies for this unprotected left scenario:

- **Proposed**: Our proposed framework, given by solving (3.22) which optimizes over both, policies and risk levels for the multi-modal chance constraint (3.18a), (3.18b).
- **Fixed risk**: An ablation of our approach, which optimizes over policies but with fixed risk levels $r_{t,j} = 1 - \epsilon$ for the multi-modal chance constraint (3.18a), (3.18b).
- **OL**: An ablation of our approach, where the gains $K^j_{k|t}, M^j_{l,k|t}$ are eliminated and risk levels are fixed too.

Note that **Proposed** and **OL** in this section are the same algorithms that were compared in Sec. 3.4.

Evaluation Metrics

We evaluate the closed-loop behavior of the policies in terms of safety and computational efficiency. The following metrics are used to assess these factors:

- **Safety**: 1) $\mathcal{F}$: Feasibility % of the SMPC optimization problem. A high $\mathcal{F}$ value is desirable, as infeasibility of the SMPC can potentially lead to accidents. 2) $\bar{d}_{min}$: Closest distance between the AV and TV of each algorithm, provided that the algorithm remains feasible. A higher $\bar{d}_{min}$ indicates that the algorithm should be more conservative to maintain safety. This caution can lead to reduced feasibility when the algorithm encounters congested urban road driving scenarios.
- **Solver Performance**: 1) $\bar{\mathcal{T}}_{solve}$: Average time taken by the solver; lower is better.

Discussion

Now, we present the results of the various SMPC policies. For each scenario, we roll out each policy for 50 different initial conditions by varying: 1) starting positions within $[-5\text{m}, 5\text{ m}]$ and 2) nominal speed in $[8m/s, 10m/s]$. For all the policies, we use a prediction horizon of $N = 10$, a discretization time-step of $dt = 0.2$ s, and a risk level of $\epsilon = 0.02$ for the chance constraints in the SMPC.

The performance metrics, averaged across the initial conditions, are shown in Table 5.1. There is a noticeable improvement in safety, as the **Proposed** can stay close to the TV-free reference trajectory and keep a safe distance from the TV. **Proposed** was also able to find feasible solutions for the SMPC optimization problem more often our experiments because the formulation optimizes over policies and risk levels for the multi-modal constraints. Finally, we see that the **OL** is the fastest in solve time (because of the missing policy and risk variables). However, we see that introducing the additional risk level variables only marginally increases the solve time on comparing **Proposed** and **Fixed risk**. The higher solve times for the proposed approach in the lane change scenario are because of the additional TV and its associated multi-modal predictions. To remedy this issue for the hardware experiments in the next section, we use a multi-threaded implementation as described in Fig. 3.10b to solve the SMPC.

The results highlight the benefits of optimizing over policies and incorporating the variable risk formulation in the SMPC formulation (7.1) for the AV, towards collision avoidance with multi-modal predictions of the TV.

3.5 Experimental Validation

In this section, we validate our approach in hardware vehicle experiments to assess the benefits of our proposed stochastic MPC formulation. The experiment videos can be accessed at `https://shorturl.at/ctQ57`.

Test scenario and key takeaways

In the hardware experiment, we consider the same lane-change scenario introduced in Sec. 3.4, wherein AV initiates a lane change maneuver with a leading TV ahead of the AV in the same lane and a trailing TV behind the AV in the adjacent lane as illustrated in Fig. 3.8. In this scenario, AV predicts multi-modal behaviors of other TVs and tries to minimize the risk of collisions for every possible mode. As illustrated in Fig. 3.8, AV predicts two different modes of TVs: keeping their lanes or changing lanes. However, TVs will not change their lane until the end of this scenario.

We compare the proposed SMPC (**Proposed**) with the **OL**, introduced in Sec. 3.4. The results in Fig. 3.12 show that while the **OL** cannot find a feasible solution due to its conservativeness of the constraint tightening formulation, **Proposed** successfully accomplishes the given scenario without any collisions. When **OL** problem becomes

infeasible, we change the control policy **OL** to a backup control policy: keeping the current lane and decelerating mildly. This abrupt policy change can deteriorate the comfort indices and it is more likely that tracking previous optimal trajectories from **OL** leads to smoother behaviors. However, due to safety concerns, we tried to stop the vehicle within the current lane.

Furthermore, we study how the predicted mode probability of the leading TV affects the closed-loop behavior. Compared to the case that the leading TV is likely to keep its lane ($p_{\text{lk}} = 0.9$), **Proposed** sets more margin in a lateral direction to avoid the collision in case the leading TV changes lanes ($p_{\text{lk}} = 0.1$) as illustrated in Fig. 3.13.

Figure 3.8: Drone view of the testing scenario including AV (red) and 2 TVs (blue) with predictions (yellow and green circles)

Hardware Architecture for Experiments

The test vehicle and the hardware setup are illustrated in Fig. 3.9. The computing unit of the system consists of three computers: a Linux-based laptop, a Linux-based rugged computer, and the dSPACE MicroAutoBox II (MABXII). The laptop is for simulating virtual environments and transmitting all information such as states of surrounding vehicles. The rugged PC is for implementing a planning and control software stack that plans the ego vehicle's behavior, generates dynamically feasible, safe trajectories, and calculates acceleration and yaw rate to track the generated trajectories. The MABXII is for implementing an actuator-level controller that calculates actuator control inputs and a fail-safe logic that provides safety features.

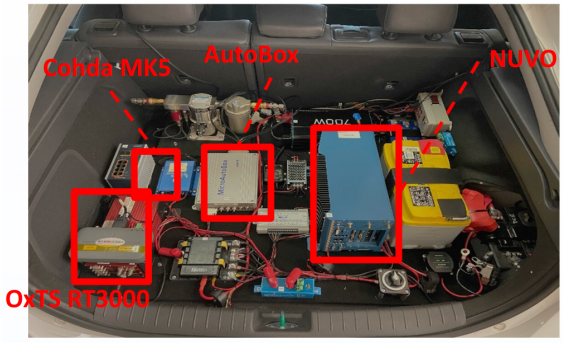
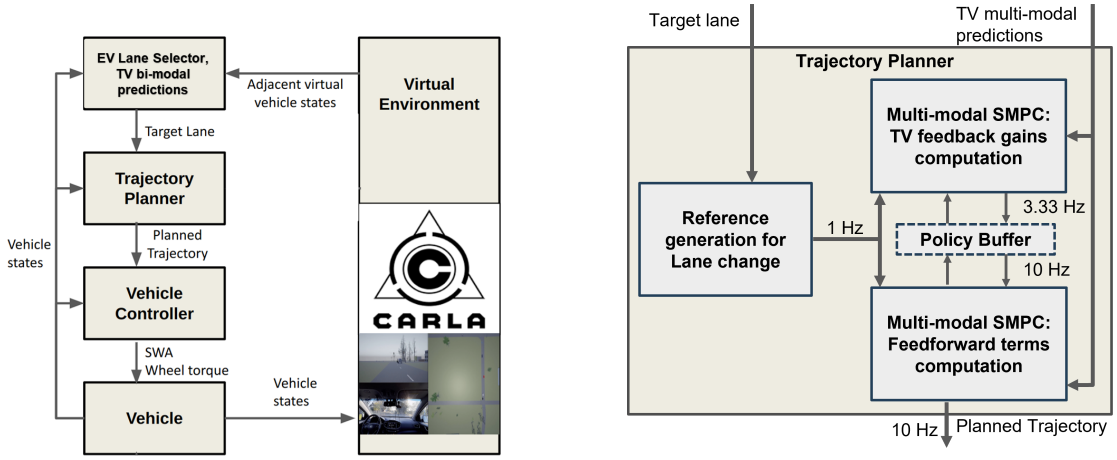


Figure 3.9: The hardware setup in the actual vehicle

The sensors of the system are an OxTS RT3000: a differential GPS to localize the ego vehicle and production vehicle sensors to acquire vehicle state information.

Software Architecture for Experiments

The overall block diagram of control architecture and the entire system is illustrated in Fig. 3.10a. In the following subsections, we describe the comprehensive details of this experimental setup.



(a) Control signal flow

(b) Trajectory planner architecture

Figure 3.10: Diagram of Control Architecture

Planning and Control Software Stack

The developed hierarchical control system consists of a lane selector, a trajectory planner, and a vehicle controller. First, the lane selector determines the target lane for the ego

vehicle. Second, given the target lane, the trajectory planner [68] generates a smooth, comfortable trajectory to the target lane by solving a nonlinear optimization problem. The calculated trajectory serves as the reference (3.4) for the proposed SMPC. Then the proposed SMPC (3.22) is solved to determine acceleration and yaw rate commands that satisfy state/input constraints. A kinematic bicycle model (3.1) is employed for the AV predictions in the SMPC optimization problem, which is modelled with CasADi and solved using Gurobi. For computing the SMPC commands at 10 Hz, we employ a multi-threaded architecture that computes the gains $K^j_{k|t}$, $M^j_{l,k|t}$ and feedforward terms $h^j_{k|t}$ at different frequencies as depicted in Fig. 3.10b. Finally, the vehicle actuator controller calculates steering wheel angle and wheel torque commands from the optimal inputs of the proposed SMPC (3.22).

Virtual Environment Simulator (Digital Twin)

To conduct real-world vehicle tests safely and efficiently, we employ a Vehicle-In-the-Loop (VIL) system as outlined in [68], which integrates the operation of an actual vehicle with a virtual environment simulation.

The CARLA [46] software is the primary simulator to build virtual environments and simulate a variety of scenarios with ease. The virtual environment simulator constructs all components such as road networks, other vehicles, traffic infrastructures, buildings, and so on to replicate the real-world map. Fig. 3.11 shows the generated CARLA map, the satellite image of the testing site, and the actual test vehicle.

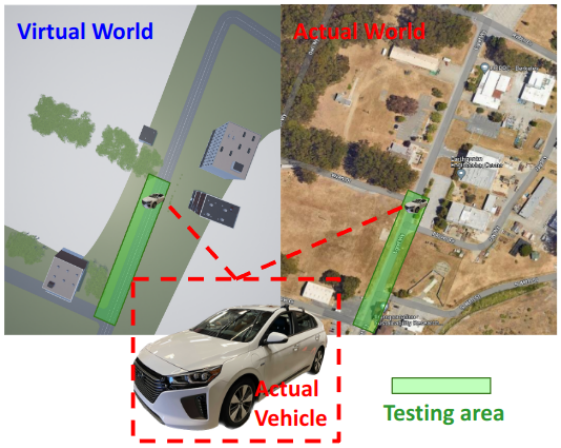


Figure 3.11: The CARLA image, the satellite image of the testing site (RFS) and the actual vehicle image

On the customized map, the CARLA simulates a traffic scenario with the same initial conditions such as the number of spawned vehicles, the locations of the vehicles, etc. It

is worth noting that the CARLA simulator exhibits inherent randomness in the motion of each virtual vehicle, resulting in variations in the resulting traffic scenario. We also synchronize the real world with the virtual world in terms of the physical AV. Based on the obtained coordinate data from dGPS/IMU sensors, the simulator generates an agent in the virtual world and teleports the vehicle by updating the position and orientation of the agent every time it receives data from the actual sensors.

Experiment Results

Our experiment setup parameters are described as follows. The AV initiates motion with an initial speed of zero. The leading TV begins its movement 10 meters ahead of the AV in the same lane, while the trailing TV starts from a position 25 meters behind the AV in the adjacent lane. Both Target Vehicles maintain a consistent average speed of 4m/s. Each of the TVs operates in two distinct modes: Lane Keeping (LK) and Lane Change (LC). The trailing TV equally splits its mode probability, with a 50% chance of LK and a 50% chance of LC. To investigate the impact of varying mode probabilities, we change the probability of the mode chosen by the leading TV.

The conducted tests are executed at the Richmond Field Station (RFS), as illustrated in Fig. 3.11. The hardware experiments are primarily divided into two segments. The first segment involves a comparison between our approach (which optimizers over multi-modal policies and risk levels) against a baseline that only optimizes over multi-modal open-loop sequences with fixed risk levels for each mode (this is **OL** from Section 3.4). The second segment focuses on assessing the behavioral outcomes resulting from alterations in the probability of the surrounding vehicle's lane change mode and lane keeping mode.

Proposed vs OL

Within the identical scenario, we conduct testing using two distinct control policies: **Proposed** in Sec. 3.4 and **OL** in Sec. 3.4. When the SMPC problem becomes infeasible, a lane-keeping controller with mild braking takes over the control. In an ideal practical application, a human driver should take over the control but due to our hardware limitations, we utilize the lane-keeping controller as a backup controller.

Fig. 3.12 presents the closed-loop behaviors: a lateral error e_y and a heading error e_ψ with respect to a centerline, vehicle speed, steering wheel angle, and longitudinal acceleration. The graphs clearly illustrate that the **OL** yields infeasible solutions, leading to abrupt and undesirable motions with constraint violations. Conversely, the **Proposed** consistently generates feasible solutions, facilitating smooth motions in accordance with the predefined constraints.

Change of mode probability

In the second part, we proceed with testing under the **Proposed** while varying the lane-keeping probability of the leading TV. Specifically, we compare the case that the leading TV is likely to keep its lane ($p_{\text{lk}} = 0.9$) with the case that the leading TV changes lanes ($p_{\text{lk}} = 0.1$). In Fig. 3.13's lateral error (e_y) graph, it is evident that the lateral distance is greater in the scenario with a lower lane-keeping probability. This observation aligns with the intuitive analysis that the resulting control policy prioritizes the lane change maneuver of the leading vehicle to the lane currently occupied by the AV. Due to the presence of collision avoidance constraints, the AV endeavors to evade the anticipated trajectory of the leading TV by maintaining larger lateral safety margins.

3.6 Conclusion

We proposed a Stochastic MPC formulation for autonomous driving with multi-modal predictions of surrounding vehicles. We provide a convex formulation for simultaneously (1) optimizing over parameterized feedback policies and (2) allocating risk levels to each mode for multi-modal chance constraint satisfaction. This enhances the feasibility and closed-loop performance of the SMPC algorithm, as demonstrated by our simulations and hardware experiments.

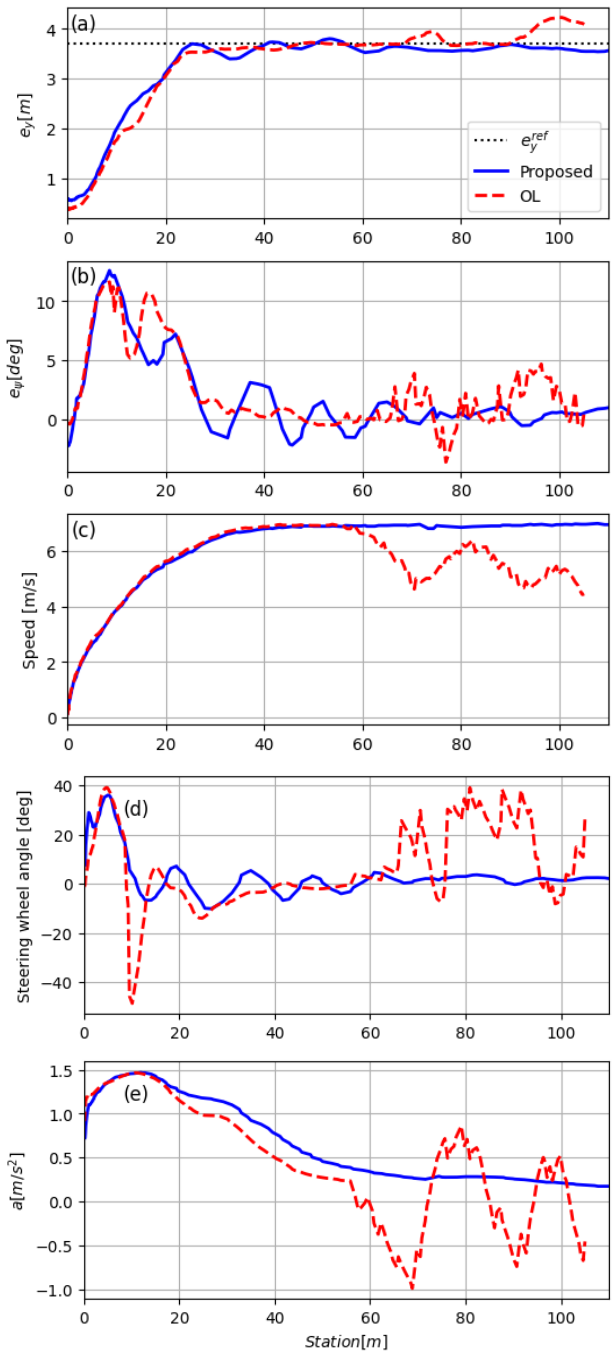


Figure 3.12: Comparison: **Proposed** vs **OL**. (a) Lateral error with respect to the centerline of the original lane. e_y^{ref} denotes the reference, (b) Heading error with respect to the centerline, (c) Vehicle longitudinal speed, (d) Steering wheel angle, and (e) Longitudinal acceleration. **Proposed** makes the ego vehicle keep the lateral distance (e_y) close to the reference while satisfying the multi-modal collision avoidance constraints. In contrast, the **OL** becomes infeasible during the task.

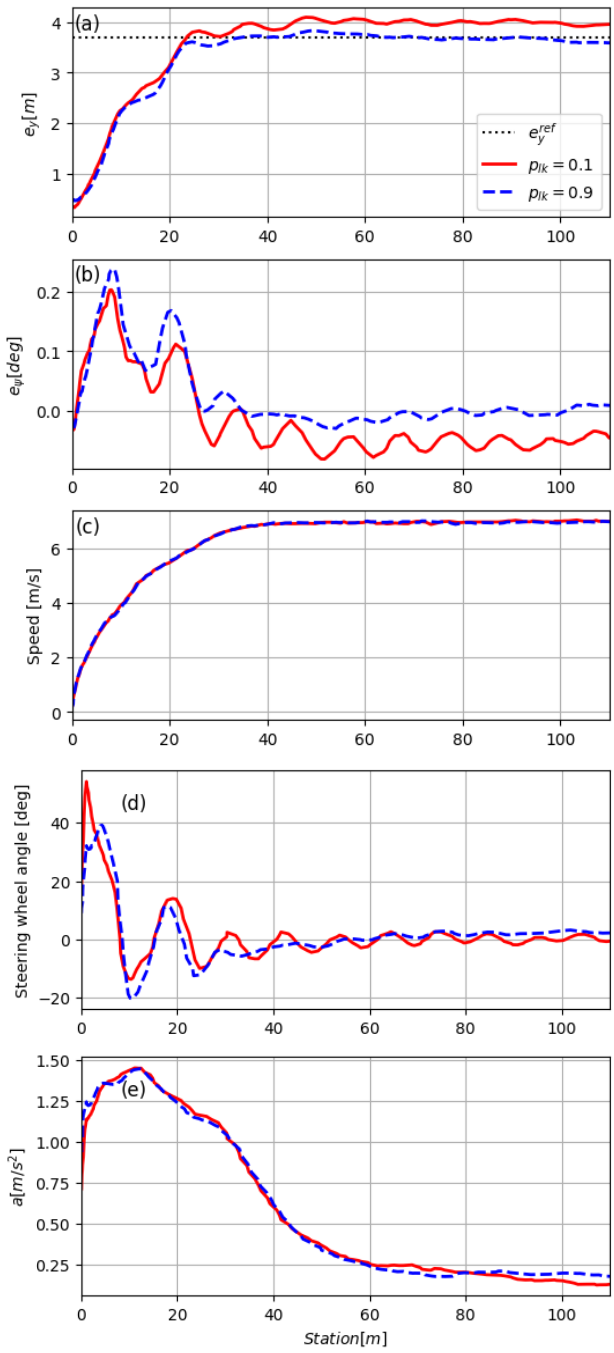


Figure 3.13: Comparison: $p_{\text{lk}} = 0.1$ vs $p_{\text{lk}} = 0.9$. (a) Lateral error with respect to the centerline of the original lane. e_y^{ref} denotes the reference, (b) Heading error with respect to the centerline, (c) Vehicle longitudinal speed, (d) Steering wheel angle, and (e) Longitudinal acceleration. Compared to the case that the leading TV is likely to keep its lane ($p_{\text{lk}} = 0.9$), **Proposed** sets more margin in a lateral direction to avoid the collision in case the leading TV changes lanes ($p_{\text{lk}} = 0.1$)

3.7 Appendix

Matrix definitions

$$\Delta \mathbf{u}_{t,j} = \mathbf{h}_t^j + \mathbf{M}_t^j \mathbf{w}_t + \mathbf{K}_t^j (\mathbf{o}_{t,j} - \boldsymbol{\mu}_{t,j}) \quad \{\textit{Stacked Control Policy}\}$$

$$\mathbf{h}_t^j = [h_{t|t}^{j\top} \dots h_{t+N-1|t}^{j\top}]^\top, \ \mathbf{K}_t^j = \text{blkdiag}\left(K_{t|t}^j, \dots, K_{t+N-1|t}^j\right),$$

$$\mathbf{M}_t^j = \begin{bmatrix} O & \dots & \dots & \dots & O \\ M_{t,t+1|t}^j & O & \dots & \dots & O \\ \vdots & \vdots & \vdots & \vdots & \\ M_{t,k|t}^j & \dots M_{k-1,k|t}^j & O & \dots & O \\ \vdots & \vdots & \vdots & \vdots & \\ M_{t,t+N-1|t}^j & \dots & \dots & M_{t+N-2,t+N-1|t}^j & O \end{bmatrix} \tag{3.24}$$

$$\Delta \mathbf{x}_{t,j} = \mathbf{A}_t \Delta x_{t|t} + \mathbf{B}_t \Delta \mathbf{u}_{t,j} + \mathbf{E}_t \mathbf{w}_t \quad \{\textit{Stacked EV Predictions}\}$$

$$\mathbf{A}_t = \begin{bmatrix} I_4 \\ A_{t|t} \\ \vdots \\ \prod_{k=t}^{t+N-1} A_{k|t} \end{bmatrix}, \ \mathbf{B}_t = \begin{bmatrix} O & \dots & \dots & O \\ B_{t|t} & O & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} A_{k|t} B_{t|t} & \dots & \dots & B_{t+N-1|t} \end{bmatrix},$$

$$\mathbf{E}_t = \begin{bmatrix} O & \dots & \dots & O \\ I_4 & O & \dots & O \\ A_{t+1|t} & I_4 & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} A_{k|t} & \dots & \dots & I_4 \end{bmatrix}, \tag{3.25}$$

$$\Delta \mathbf{o}_{t,j} = \mathbf{T}_t^j o_{t|t} + \mathbf{C}_t^j + \mathbf{F}_t^j \mathbf{n}_{t,j} \quad \{\textit{Stacked TV Predictions}\}$$

$$\mathbf{T}_t^j = \begin{bmatrix} I_2 \\ T_{t|t,j} \\ T_{t+1|t,j} T_{t|t,j} \\ \vdots \\ \prod_{k=t}^{t+N-1} T_{k|t,j} \end{bmatrix}, \ \mathbf{C}_t^j = \begin{bmatrix} O \\ c_{t|t,j} \\ c_{t+1|t,j} + T_{t+1|t,j} c_{t|t,j} \\ \vdots \\ c_{t+N-1|t,j} + \sum_{k=t}^{t+N-1} \prod_{l=k+1}^{t+N-1} T_{l|t,j} c_{k|t,j} \end{bmatrix},$$

$$\mathbf{L}_t^j = \begin{bmatrix} O & \dots & \dots & O \\ I_2 & O & \dots & O \\ T_{t+1|t,j} & I_2 & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} T_{k|t,j} & \dots & \dots & I_2 \end{bmatrix} \tag{3.26}$$

$$\boldsymbol{\Sigma}_w = I_N \otimes \Sigma_w, \ \boldsymbol{\Sigma}_n^j = \text{blkdiag}(\Sigma_{t|t,j}, \dots, \Sigma_{t+N-1|t,j}) \tag{3.27}$$

$$\mathbf{Q} = I_{N+1} \otimes Q, \ \mathbf{R} = I_N \otimes R, \tag{3.28}$$

Proof of proposition 1

First note that $P^i_{k|t,j}$ is defined such that $g^i_{k|t,j}(P^i_{k|t,j}, \mu^i_{k|t,j}) = 1$. For any convex function $f(x)$, we have $\forall x_0, x : f(x) \geq f(x_0) + \partial_x f(x_0)(x - x_0)$. Since we $g^i_{k|t,j}(\cdot)$ is convex, we have $g^i_{k|t,j}(P,o) \geq g^i_{k|t,j}(P^i_{k|t,j}, \mu^i_{k|t,j}) + l^i_{k|t,j}(P,o) = 1 + l^i_{k|t,j}(P,o)$. So $l^i_{k|t,j}(P,o) \geq 0 \Rightarrow g^i_{k|t,j}(P,o) \geq 1$, and $\forall k = t+1, \ldots, t+N$,

$$\begin{aligned} &\sum_{j=1}^{J} p_{t,j} \mathbb{P}(l^i_{k|t,j}(P^i_{k|t}, o^i_{k|t,j}) \geq 0\ ) \geq 1 - \epsilon \\ \Rightarrow &\sum_{j=1}^{J} p_{t,j} \mathbb{P}(g^i_{k|t,j}(P^i_{k|t}, o^i_{k|t,j}) \geq 1) \geq 1 - \epsilon \end{aligned}$$

■

Proof of Proposition 2

First, we note the following auxilliary result: For a convex function $f(\cdot)$ and $x \in [x_{min}, x_{max}] \subset \mathbb{R}$, if $f(x_{min} y) \leq 0$, $f(x_{max} y) \leq 0$, then $f(xy) \leq 0$ for any $x \in [x_{min}, x_{max}]$. Now suppose that (3.20) holds, and use the above result for an arbitrary $\tilde{\eta}_{t,j} = \gamma \eta_{max} + (1-\gamma)0$ with $\gamma \in [0,1]$ to get:

$$\begin{aligned} &a_0 + a_1^\top \mathbf{h}^j_t - b \geq \\ &\left\| \begin{bmatrix} \boldsymbol{\Sigma}_t (\tilde{\mathbf{M}}^{j\top}_t a_2 + \tilde{\eta}_{t,j} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{M}}^{j,b\top}_t a_2 + a_3 \eta_{t,j}) \\ \boldsymbol{\Sigma}_{t,j} \mathbf{L}^{j\top}_t (\tilde{\mathbf{K}}^{j\top}_t a_4 + \tilde{\eta}_{t,j} \sum_{b=1}^{|\mathcal{B}_{t,j}|-1} \bar{\mathbf{K}}^{j,b\top}_t a_4 + a_5 \eta_{t,j}) \end{bmatrix} \right\|_2 \end{aligned}$$

Since, $\tilde{\eta}_{t,j}$ is arbitrary, the desired result is obtained by setting $\tilde{\eta}_{t,j} = \eta_{t,j}$. ■

Proof of proposition 3

The set of feasible policy parameters $\Theta^{MPC}_t(x_t, o_t; \mathcal{T}_t)$ is given by SOC constraints by construction. It remains to show that the cost function is quadratic, presented below:

$$\begin{aligned} C_t(\mathbf{x}_t, \mathbf{u}_t) &= \sum_{j=1}^{J} p_{t,j} \mathbb{E} \left(\sum_{k=t}^{t+N-1} \Delta x^\top_{k+1|t} Q \Delta x_{k+1|t} + \Delta u^\top_{k|t} R \Delta u_{k|t} \right) \\ &= -\Delta x^\top_{t|t} Q \Delta x_{t|t} + \sum_{j=1}^{J} p_{t,j} \mathbb{E}(\boldsymbol{\Delta x}^\top_{t,j} \mathbf{Q} \boldsymbol{\Delta x}_{t,j} + \boldsymbol{\Delta u}^\top_{t,j} \mathbf{R} \boldsymbol{\Delta u}_{t,j}) \\ &= \sum_{j=1}^{J} p_{t,j} \Big[\mathbf{h}^{j\top}_t (\mathbf{B}^\top_t \mathbf{Q} \mathbf{B}_t + \mathbf{R}) \mathbf{h}^j_t + 2 \Delta x^\top_{t|t} \mathbf{A}^\top_t \mathbf{Q} \mathbf{B}_t \mathbf{h}^j_t + \Delta x^\top_{t|t} (\mathbf{A}^\top_t \mathbf{Q} \mathbf{A}_t - Q) \Delta x_{t|t} \\ &+ \operatorname{tr}\big(\boldsymbol{\Sigma}_w \mathbf{M}^{j\top}_t (\mathbf{B}^\top_t \mathbf{Q} \mathbf{B}_t + \mathbf{R}) \mathbf{M}^j_t + \mathbf{L}^j_t \boldsymbol{\Sigma}_n \mathbf{L}^{j\top}_t \mathbf{K}^{j\top}_t (\mathbf{B}^\top_t \mathbf{Q} \mathbf{B}_t + \mathbf{R}) \mathbf{K}^j_t + \boldsymbol{\Sigma}_w \mathbf{E}^\top_t \mathbf{Q} \mathbf{E}_t \big) \Big] \end{aligned}$$

Since the cost is quadratic in $\boldsymbol{\theta}_t$, we have that (3.22) is a SOCP. ■

Part II

Performance

Chapter 4

Output-lifted Learning Model Predictive Control

4.1 Overview

Infinite-horizon optimal control has a long and celebrated history, with the cornerstones laid in the 1950s by [106] and [14]. The problem involves seeking a control signal that minimizes the cost incurred by a trajectory of a dynamical system starting from an initial condition over an infinite time horizon. While certain problem settings admit analytical solutions (like unconstrained LQR [77]), the infinite-horizon optimal control problem for general nonlinear dynamical systems subject to constraints, is challenging to solve. This is because these problems require the numerical solution of an infinite-dimensional optimization problem, which is intractable even in the discrete-time setting (where the solution is an infinite sequence of control inputs instead of a control input signal). Model Predictive Control (MPC) is a methodology for tractable synthesis of feedback control of constrained nonlinear discrete-time systems. The control action at every instant requires the solution of a finite-horizon optimal control problem with a suitable constraint and cost on the terminal state of the system to approximate the infinite-horizon problem. These terminal components are designed so that the closed-loop system is stabilized to a desired *goal set* and satisfies constraints. This is achieved by constraining the terminal state to lie in a control invariant set containing the goal set, with an associated Control Lyapunov function (CLF) [88]. The computation of these sets with an accompanying CLF for nonlinear systems is challenging in general, and typically require local approximations of the nonlinear dynamics around the goal set [49, 21, 19]. Global constructions of these components has been studied for systems with polynomial dynamics using sum-of-squares (SOS) programming [127, 74]. However the resulting SOS programs are often difficult to solve for high-dimensional systems, and moreover, are challenging to incorporate into (1.4) for efficient computation of the MPC policy

The Learning Model Predictive Control (LMPC) strategy proposed in [114] iteratively

constructs a control invariant terminal set and an accompanying terminal cost function in an online fashion using historical data. The terminal set is defined as a discrete set of past states and the terminal cost is only defined at these points and so the LMPC relies on the solution of a Mixed-Integer Nonlinear Program (MINLP) at each instant for guaranteed stability and constraint satisfaction. At each iteration, a feasible solution to an infinite-horizon optimal control problem is obtained in an MPC framework. The terminal set is recursively defined as the set of states obtained from previous iterations and for each state in the set, the terminal cost is simply defined to be the cost incurred by the obtained trajectory starting from that state. The LMPC has been shown to stabilize constrained nonlinear systems while decreasing the cost incurred by trajectories with every successive iteration. However these guarantees require solving a MINLP for computing the MPC which may be prohibitive for real-time control. In [113], the authors convexify the terminal set and the cost function and show that for linear systems, the stability and iterative cost improvement properties of the LMPC are preserved while ensuring that system operates within constraints. In this chapter, we propose a strategy to reduce the computational burden of LMPC for a class of nonlinear systems by replacing the discrete terminal set, cost with continuous, convex ones while still maintaining safety and performance guarantees. Moreover, these quantities are computed without any local approximations of the nonlinear dynamics.

We consider discrete-time nonlinear systems for which the state and input can be reconstructed using certain system output sequences, which are defined as *lifted outputs*. These outputs sequences are constructed using *flat* outputs [56] which have also been used in [9] to construct dynamic feedback linearizing inputs for discrete-time systems. Existing work on constrained control for such systems require a carefully designed reference trajectory which is then tracked using MPC with a linear model obtained either by a first order approximation [43] or by feedback linearization [135, 55, 71, 6, 4]. In both cases, there are no formal guarantees of closed-loop system stability and constraint satisfaction, except in [6] and [4]. [6] proposes a model-free, data-driven approach based on Willem's fundamental lemma for Robust MPC by using input-output data of the feedback-linearized system. However, the formulation can not enforce state constraints for a recursively feasible Robust MPC scheme. In [4] a model-based hierarchical approach is considered: a Robust MPC scheme with the feedback-linearized dynamics provides a reference trajectory, and a low level tracking controller for original nonlinear system is designed to track the reference trajectory. The approach in [4] does not address input constraints, and the terminal set is chosen as the desired *goal set*.

Contributions

1. We iteratively construct convex control invariant sets and CLFs in the space of lifted outputs using historical trajectory data for flat nonlinear systems. These constructions capture the global, nonlinear dynamics of the system using data.

2. We propose an iterative Robust LMPC strategy for nonlinear systems with uncertainty that iteratively collects data for learning (i) the control invariant sets and CLFs, and (ii) the model uncertainty using ideas from Chapter 2

3. Our iterative approach rigorously provides closed-loop guarantees of *i*) constraint satisfaction, *ii*) convergence to a desired set, *iii*) non-decreasing closed-loop system performance across iterations. This is validated via numerical simulations for optimizing an autonomous racing trajectory over a chicane.

4.2 Problem Formulation

Consider a nonlinear discrete-time system given by the dynamics

$$x_{t+1} = f(x_t, u_t) + d_t,\ d_t \in D(x_t, u_t) \tag{4.1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system state and input respectively at time t, and $f(\cdot,\cdot)$ is a known, continuous function. As in Chapter 2, the disturbance d_t is assumed to belong to compact set $D(x_t, u_t)$, where $D(\cdot)$ is a set-valued map, $D : \mathbb{R}^{n+m} \to 2^{\mathbb{R}^n}$. The map $D(\cdot)$ is *unknown* and represents unmodelled dynamics. We assume that this map satisfies the incremental property stated by the following assumption.

Assumption 4.1 *The unknown set-valued map $D(\cdot)$ satisfies the following quadratic constraints, for any $q = (x, u), q' = (x', u')$ in $\mathbb{R}^{n+m}$:*

$$\begin{bmatrix} 1 \\ q - q' \\ d - d' \end{bmatrix}^\top Q \begin{bmatrix} 1 \\ q - q' \\ d - d' \end{bmatrix} \geq 0,\ \forall d \in D(q), \forall d' \in D(q'), \quad \forall Q \in \boldsymbol{\mathcal{Q}}$$

where $\boldsymbol{\mathcal{Q}} = \{Q^{(1)}, \ldots, Q^{(n_Q)}\}$ is a known, finite set of symmetric matrices. Additionally, $D(z)$ is compact for all $z \in \mathbb{R}^{n+m}$.

Also define the *nominal* nonlinear discrete-time system,

$$\bar{x}_{t+1} = f(\bar{x}_t, \bar{u}_t) \tag{4.2}$$

where $\bar{x}_t \in \mathbb{R}^n$, $\bar{u}_t \in \mathbb{R}^m$, are the nominal system state and input at time t. The output $\bar{y}_t$ is a difference flat output [56], and is used to construct the *lifted* output for the nominal system (4.2) as discussed next.

Definition 4.1 *Let $\bar{y}_t = h(\bar{x}_t)$ with $h : \mathbb{R}^n \to \mathbb{R}^m$ be the output of system* (4.2)*. If $\exists R \in \mathbb{N}$ and a function $\mathcal{F} : \mathbb{R}^{m \times R+1} \to \mathbb{R}^n \times \mathbb{R}^m$, such that any state/input pair $(\bar{x}_t, \bar{u}_t)$ can be uniquely reconstructed from a sequence of outputs $\bar{y}_t, \ldots, \bar{y}_{t+R}$ as*

$$(\bar{x}_t, \bar{u}_t) = \mathcal{F}([\bar{y}_t, \bar{y}_{t+1}, \ldots, \bar{y}_{t+R}]), \tag{4.3}$$

then the lifted output is the matrix

$$\bar{\mathbf{Y}}_t = [\bar{y}_t, \ldots, \bar{y}_{t+R}] \in \mathbb{R}^{m \times R+1}. \tag{4.4}$$

Remark 4.1 *For linear discrete-time systems, the existence of the lifted output is equivalent to the system being controllable and strongly observable with the output $y_t = Cx_t$ [140]. For nonlinear systems, flat outputs exist for systems that admit a dynamic feedback linearizing compensator [9].*

We formally assume the existence of the lifted output for our nominal system along with some additional structure on the map $\mathcal{F}(\cdot)$ next.

Assumption 4.2 *We are given an output function $\bar{y}_t = h(\bar{x}_t)$ with corresponding lifted output $\bar{\mathbf{Y}}_t$ for the nominal system* (4.2)*. Moreover, the map $\mathcal{F}(\cdot)$ in* (4.3) *also satisfies the following properties:*

(A) $\mathcal{F}(\cdot)$ is continuous, and requires R and $R+1$ outputs for identifying the nominal state and nominal input respectively, i.e.,

$$\bar{x}_t = \mathcal{F}_x([\bar{y}_t, \bar{y}_{t+1}, \ldots, \bar{y}_{t+R-1}]) \tag{4.5}$$
$$\bar{u}_t = \mathcal{F}_u([\bar{y}_t, \bar{y}_{t+1}, \ldots, \bar{y}_{t+R}]) \tag{4.6}$$

(A) Let $\mathcal{F}^i : \mathbb{R}^{m \times R+1} \to \mathbb{R}$ be the ith component of the map $\mathcal{F} : \mathbb{R}^{m \times R+1} \to \mathbb{R}^n \times \mathbb{R}^m$ where $i = 1, \ldots, n+m$. For each $i \in \{1, \ldots, n+m\}$, there exist functions $\mathcal{F}^{i,\cup}$, $\mathcal{F}^{i,\cap}$ such that $\mathcal{F}^{i,\cap}(\bar{\mathbf{Y}}) \leq \mathcal{F}^i(\bar{\mathbf{Y}}) \leq \mathcal{F}^{i,\cup}(\bar{\mathbf{Y}})$ where $\mathcal{F}^{i,\cap}$ is quasiconcave and $\mathcal{F}^{i,\cup}$ is quasiconvex, i.e.,

$$\begin{aligned}
&\forall \bar{\mathbf{Y}}^1, \bar{\mathbf{Y}}^2 \in \mathbb{R}^{m \times R+1}, \forall t \in [0,1] : \\
&\min\{\mathcal{F}^{i,\cap}(\bar{\mathbf{Y}}^1), \mathcal{F}^{i,\cap}(\bar{\mathbf{Y}}^2)\} \leq \mathcal{F}^{i,\cap}(t\bar{\mathbf{Y}}^1 + (1-t)\bar{\mathbf{Y}}^2), \\
&\mathcal{F}^{i,\cup}(t\bar{\mathbf{Y}}^1 + (1-t)\bar{\mathbf{Y}}^2) \leq \max\{\mathcal{F}^{i,\cup}(\bar{\mathbf{Y}}^1), \mathcal{F}^{i,\cup}(\bar{\mathbf{Y}}^2)\}.
\end{aligned}$$

The additional structure imposed by Assumption 4.2 is used for constructing invariant sets for (4.1), (4.2) using historical data, which will be clarified in Section .

Example 4.1 *Consider the kinematic bicycle, described by the Euler-discretized dynamics*

$$\begin{aligned}
X_{t+1} &= X_t + dt(v_t \cos(\theta_t)) \\
Y_{t+1} &= Y_t + dt(v_t \sin(\theta_t)) \\
\theta_{t+1} &= \theta_t + dt(\frac{v_t}{L_f} \tan^{-1}(\frac{L_r}{L_f + L_r} \tan(\delta_t)))
\end{aligned}$$

with states $\bar{x}_t = [X_t, Y_t, \theta_t]$ and controls $\bar{u}_t = [v_t, \delta_t$. For the output $\bar{y}_t = [X_t, Y_t]$, the states are reconstructed as $[X_t, Y_t] = \bar{y}_t$, $\theta_t = \tan^{-1}(\frac{[0\ 1](\bar{y}_{t+1} - \bar{y}_t)}{[1\ 0](\bar{y}_{t+1} - \bar{y}_t)})$ and the inputs are reconstructed

as $v_t = \frac{\|\bar{y}_{t+1}-\bar{y}_t\|}{dt}$, $\delta_t = \tan^{-1}(\frac{L_f+L_r}{L_r}\tan(\frac{L_f(\theta_{t+1}-\theta_t)}{dtv_t}))$ *(which involves* $\bar{y}_t, \bar{y}_{t+1}, \bar{y}_{t+2}$*), and so* $R = 2$.

For Assumption 4.2((A))*, the states can be reconstructed from two consecutive outputs alone, i.e.,* $\mathcal{F}_x(\bar{y}_t, \bar{y}_{t+1}) = [\bar{y}_t, \tan^{-1}(\frac{[0\ 1](\bar{y}_{t+1}-\bar{y}_t)}{[1\ 0](\bar{y}_{t+1}-\bar{y}_t)})]$.

For Assumption 4.2((A))*, the bounding functions corresponding to the states are*

$$[\mathcal{F}^{1,\cap}(\bar{\mathbf{Y}}_t), \mathcal{F}^{2,\cap}(\bar{\mathbf{Y}}_t)] = [\mathcal{F}^{1,\cup}(\bar{\mathbf{Y}}_t), \mathcal{F}^{2,\cup}(\bar{\mathbf{Y}}_t)] = \bar{y}_t$$
$$\mathcal{F}^{3,\cap}(\bar{\mathbf{Y}}_t) = \mathcal{F}^{3,\cup}(\bar{\mathbf{Y}}_t) = \tan^{-1}(\frac{[0\ 1](\bar{y}_{t+1}-\bar{y}_t)}{[1\ 0](\bar{y}_{t+1}-\bar{y}_t)})$$

if $[1\ 0](\bar{y}_{t+1}-\bar{y}_t) > 0$ *(*$\because$ *composition of quasi-linear function* $\frac{[0\ 1](\bar{y}_{t+1}-\bar{y}_t)}{[1\ 0](\bar{y}_{t+1}-\bar{y}_t)}$ *with monotonically increasing function* $\tan^{-1}(\cdot)$ *is both quasi-convex and quasi-concave). For the bounding functions of the inputs, we have for example,*

$$\mathcal{F}^{4,\cap}(\bar{\mathbf{Y}}_t) = 0, \mathcal{F}^{4,\cup}(\bar{\mathbf{Y}}_t) = \frac{\|\bar{y}_{t+1}-\bar{y}_t\|}{dt}$$
$$\mathcal{F}^{5,\cap}(\bar{\mathbf{Y}}_t) = -\frac{\pi}{2}, \mathcal{F}^{5,\cup}(\bar{\mathbf{Y}}_t) = \frac{\pi}{2}$$

Remark 4.2 *Assumption 4.2*((A)) *is naturally satisfied by flat, simple mechanical systems [94] (where the system geometry/kinematics are affected by the control inputs via integrators). The bounding functions* $\mathcal{F}^{i,\cap}(\cdot), \mathcal{F}^{i,\cup}(\cdot)$ *in Assumption 4.2*((A)) *can be constructed by exploiting system constraints, and the required properties can be verified via first & second order conditions or composition rules for quasiconvex functions [23]. If* $\mathcal{F}^i(\cdot)$ *is both quasiconcave and quasiconvex already, then the bounding functions are simply* $\mathcal{F}^{i,\cap}(\cdot) = \mathcal{F}^{i,\cup}(\cdot) = \mathcal{F}^i(\cdot)$.

System Constraints and Goal Set

In this work, we assume that system (4.1) is subject to state and input constraints given by box sets $\{z|\ lb \leq z \leq ub\}$. We denote the goal set $\mathcal{X}_G = \{x \in \mathbb{R}^n | A^g x \leq b^g\}$ as the set within constraints into which we would like to steer the trajectories of (4.1). We suppose that this set is control invariant for (4.1) i.e., $\forall x_t \in \mathcal{X}_G, \exists u_t \in \mathcal{U} \Rightarrow x_{t+1} \in \mathcal{X}_G$. We formalize this as the assumption below.

Assumption 4.3 *The state constraints* $\mathcal{X}$ *and input constraints* $\mathcal{U}$ *are given by,*

$$\mathcal{X} = \{x \in \mathbb{R}^n | lb_x \leq x \leq ub_x\},$$
$$\mathcal{U} = \{u \in \mathbb{R}^m | lb_u \leq u \leq ub_u\}.$$

for some $lb_x, ub_x \in \mathbb{R}^n$ *and* $lb_u, ub_u \in \mathbb{R}^m$*. The goal set* $\mathcal{X}_G = \{x \in \mathbb{R}^n | A^g x \leq b^g\} \subset \mathcal{X}$ *is robustly control invariant for* (4.1)*,*

$$x_t \in \mathcal{X}_G \implies \exists u_t \in \mathcal{U} : x_{t+1} \in \mathcal{X}_G\ \forall d_t \in D(x_t, u_t)$$

In subsequent sections, we show that the box constraints are required to exploit the element-wise bounds on $\mathcal{F}(\cdot)$ given in Assumption 4.2((A)). This helps in using system trajectory data to construct lifted outputs that map to nominal states and inputs within constraints.

Robust MPC for Nonlinear Systems

Consider the following infinite horizon robust optimal control problem for (4.1) starting from state $x_0 = x_S$:

$$\begin{aligned} J^*_{0\to\infty}(x_S) = \min_{u_0, u_1(\cdot), \ldots} \quad & \sum_{k\geq 0} c(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) + d_k,\ \forall k \geq 0 \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U}, \forall d_k \in D(x_k, u_k)\ \forall k \geq 0 \\ & x_0 = x_S. \end{aligned} \tag{4.7}$$

where the stage cost $c(\cdot)$ is continuous and chosen such that

$$\begin{aligned} c(\bar{x}, \bar{u}) = 0\ \forall (\bar{x}, \bar{u}) \in \mathcal{X}_G \times \mathcal{U}, \\ c(\bar{x}, \bar{u}) > 0\ \forall (\bar{x}, \bar{u}) \in \mathbb{R}^{n+m} \backslash \mathcal{X}_G \times \mathcal{U}. \end{aligned}$$

Observe that due to continuity and positive definiteness of stage cost $c(\cdot)$, a trajectory corresponding to the optimizer of (4.7) must necessarily have its state converge to $\mathcal{X}_G$.

We aim to synthesize a state-feedback policy that approximates the solution to the infinite-horizon (and infinite-dimensional) problem (4.7) such that it captures its most desirable properties: (i) constraint satisfaction (feasibility) and (ii) asymptotic convergence (bounded cost). To tackle the infinite-dimensional nature of the problem, we propose to use a rigid/homethetic tube-based [75] Robust MPC formulation which solves finite-horizon versions of (4.7) at each time step.

Define the error system with state $e_t = x_t - \bar{x}_t$ and dynamics

$$\begin{aligned} e_{t+1} &= f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t,\ d_t \in D(e_t + \bar{x}_t, u_t) \\ f_e(e_t, \bar{x}_t, \bar{u}_t) &= f(e_t + \bar{x}_t, \bar{u}_t + \kappa(e_t)) - f(\bar{x}_t, \bar{u}_t) \end{aligned} \tag{4.8}$$

where u_t is given as the sum of $\bar{u}_t$ and error feedback policy $\kappa(e_t)$. For a fixed policy $\kappa(e_t)$, a robustly positively invariant set $\mathcal{E}$ for the error dynamics (4.8) satisfies,

$$\begin{aligned} &\forall x_t \in \mathcal{X}, \forall u_t \in \mathcal{U}: \\ &e_t \in \mathcal{E} \Rightarrow e_{t+1} \in \mathcal{E}\ \forall d_t \in D(x_t, u_t). \end{aligned} \tag{4.9}$$

The nominal input $\bar{u}_t$ is obtained by solving the following finite-horizon optimal control problem for the nominal system,

$$\begin{aligned} \bar{J}(x_t) = \min_{\bar{\mathbf{u}}_\mathbf{t}, \bar{\mathbf{x}}_\mathbf{t}} \quad & P(\bar{x}_{t+N|t}) + \sum_{k=t}^{t+N-1} c(\bar{x}_{k|t}, \bar{u}_{k|t}) \\ \text{s.t.} \quad & \bar{x}_{k+1|t} = f(\bar{x}_{k|t}, \bar{u}_{k|t}), \\ & \bar{x}_{k|t} \in \bar{\mathcal{X}}, \bar{u}_{k|t} \in \bar{\mathcal{U}}, \ \forall k \in \mathbb{I}_t^{t+N-1}, \\ & \bar{x}_{t+N|t} \in \bar{\mathcal{X}}_N, \\ & x_t - \bar{x}_{t|t} \in \mathcal{E} \end{aligned} \tag{4.10}$$

where $\bar{\mathbf{u}}_\mathbf{t} = [\bar{u}_{t|t}, \ldots, \bar{u}_{t+N-1|t}]$, $\bar{\mathbf{x}}_\mathbf{t} = [\bar{x}_{t|t}, \ldots, \bar{x}_{t+N|t}]$ are decision variables, and x_t is the current state. The optimal solution of (4.10) provides the nominal control input as

$$\bar{u}_t^{MPC}(x_t) = \bar{u}_{t|t}^\star \tag{4.11}$$

and the resulting feedback controller for system (4.1) is

$$u_t = \pi(x_t) = \bar{u}_t^{MPC}(x_t) + \kappa(x_t - x_{t|t}^*) \tag{4.12}$$

The sets $\bar{\mathcal{X}} \subset \mathcal{X} \ominus \mathcal{E}$ and $\bar{\mathcal{U}} \subset \mathcal{U} \ominus \kappa(\mathcal{E})$ are tightened nominal state and input constraints, where $\kappa(\mathcal{E}) = \{u \in \mathbb{R}^m | \exists e \in \mathcal{E} : \kappa(e) = u\}$ to ensure that if $\bar{x}_t \in \bar{\mathcal{X}}, \bar{u}_t \in \bar{\mathcal{U}}, e_t \in \mathcal{E}$, then $x_t \in \mathcal{X}, u_t \in \mathcal{U}$. The terminal cost $P(\cdot)$ and terminal set $\bar{\mathcal{X}}_N$ are designed such that the oproblem (4.10) has a feasible solution $\forall t \geq 0$ for system (4.1) in closed-loop with the control (4.12), yielding a feasible and stabilizing solution to the infinite-horizon problem (4.7). Observe that if the optimal cost $J_{t\to\infty}^*(x_t) = \sum_{k\geq t} c(x_k^*, u_k^*)$ was known $\forall t \geq 0$ over the state space, setting $P(\bar{x}_{t+N|t}) = J_{t+N\to\infty}^*(\bar{x}_{t+N|t})$ solves (4.7) without requiring a terminal constraint $\bar{\mathcal{X}}_N$ in (4.10). The other extreme case is setting $\bar{\mathcal{X}}_N = \mathcal{X}_G \ominus \mathcal{E}$ in (4.10) which would yield a stable and feasible solution without requiring a terminal cost $P(\cdot)$. That being said, computing $J_{t\to\infty}^*(\cdot)$ exactly is possible only in trivial cases and setting $\bar{\mathcal{X}}_N = \mathcal{X}_G \ominus \mathcal{E}$ may lead to an infeasible optimization problem if $\mathcal{X}_G \ominus \mathcal{E}$ is not $N-$step reachable from a nominal state $\bar{x}_0$ such that $x_S - \bar{x}_0 \in \mathcal{E}$. The goal is to design $\bar{\mathcal{X}}_N$ and $P(\cdot)$ so that (4.10) is feasible for all $t \geq 0$ while capturing the convergence properties of the infinite-horizon robust optimal control problem.

Overview of Proposed Approach

The remainder of this chapter is organized as follows:

1. First, we design the terminal set $\bar{\mathcal{X}}_N$ by constructing a convex set using lifted output data $\{[\bar{y}_t, .., \bar{y}_{t+R-1}]\}_{t\geq 0}$, and taking its image under the map $\mathcal{F}_x(\cdot)$ from Assumption 4.2((A)). We provide a constructive proof in Section 4.3 for showing that this set is control invariant for the nominal system (4.2): Definition 4.1 and Assumption 4.2((A)) together guarantee the existence of a control input to keep the state inside

the set $\bar{\mathcal{X}}_N$, and Assumption 4.2((A)) ensures that this input is within the input constraints $\mathcal{U}$. The terminal cost is constructed using Barycentric interpolation in the space $[\bar{y}_t, .., \bar{y}_{t+R-1}]$ and verified to be a Control Lypunov Function (CLF).

2. Second, we use uncertainty quantification technique from Chapter 2 to construct outer-approximations $\hat{\mathcal{D}}$ of the set

$$D(\mathcal{X}, \mathcal{U}) = \bigcup_{(x,u)\in\mathcal{X}\times\mathcal{U}} D(x,u), \tag{4.13}$$

using trajectory data $\{(x_t, u_t)\}_{t\geq 0}$ from the real system (4.1). This is used to construct the error invariant $\mathcal{E}$ and define the tightened constraints $\bar{\mathcal{X}}, \bar{\mathcal{U}}$.

3. Finally, we present an iterative procedure for collecting trajectory data that robustly satisfies constraints using a Robust MPC policy. This data is used for learning (i) the terminal control invariant $\bar{\mathcal{X}}_N$ and cost $P(\dot{)}$, and (ii) the uncertainty estimate $\hat{D}$.

In Section 4.6, the resulting trajectories of system (4.1) in closed-loop with the Robust MPC policy are shown to satisfy constraints robustly, and with non-increasing trajectory costs across iterations.

4.3 Convex Control Invariant Sets and Control Lyapunov Functions from Trajectory Data

In this section, we demonstrate how to exploit Assumption 4.2 to construct a control invariant set within system constraints and a Control Lyapunov function using trajectory data of the nominal system (4.2). To proceed, we stipulate conditions that the trajectory data must satisfy.

Define the following sets in the state and input space,

$$\bar{\mathcal{S}}_x = \left\{ \bar{x} \in \mathbb{R}^n \middle| \begin{matrix} \exists \bar{\mathbf{y}} \in \mathbb{R}^{m\times R}, \bar{x} = \mathcal{F}_x(\bar{\mathbf{y}}) \\ lb_x \leq \mathcal{F}_x^{\cap}(\bar{\mathbf{y}}), \mathcal{F}_x^{\cup}(\bar{\mathbf{y}}) \leq ub_x \end{matrix} \right\} \tag{4.14}$$

$$\bar{\mathcal{S}}_u = \left\{ \bar{u} \in \mathbb{R}^m \middle| \begin{matrix} \exists \bar{\mathbf{Y}} \in \mathbb{R}^{m\times R+1}, \bar{u} = \mathcal{F}_u(\bar{\mathbf{Y}}) \\ lb_u \leq \mathcal{F}_u^{\cap}(\bar{\mathbf{Y}}), \mathcal{F}_u^{\cup}(\bar{\mathbf{Y}}) \leq ub_u \end{matrix} \right\}, \tag{4.15}$$

where $\mathcal{F}_x^{\cup}(\cdot) = [\mathcal{F}^{1,\cup}(\cdot), \ldots, \mathcal{F}^{n,\cup}(\cdot)]$, $\mathcal{F}_u^{\cup}(\cdot) = [\mathcal{F}^{n+1,\cup}(\cdot), \ldots, \mathcal{F}^{n+m,\cup}(\cdot)]$ and $\mathcal{F}_x^{\cap}(\cdot)$, $\mathcal{F}_u^{\cap}(\cdot)$ are defined similarly. Observe that the inequalities in the definition ensure that $\bar{\mathcal{S}}_x \subseteq \mathcal{X}$, $\bar{\mathcal{S}}_u \subseteq \mathcal{U}$ because $\mathcal{F}^{\cap}(\bar{\mathbf{Y}}) \leq \mathcal{F}(\bar{\mathbf{Y}}) = (\bar{x}, \bar{u}) \leq \mathcal{F}^{\cup}(\bar{\mathbf{Y}})$ from Assumption 4.2((A)). Additionally, the sets $\bar{\mathcal{S}}_x, \bar{\mathcal{S}}_u$ also enjoy the following property which will be crucial for constructing the invariant sets within constraints from data.

Lemma 4.1 *Let* $\{\bar{\mathbf{Y}}^1, \ldots, \bar{\mathbf{Y}}^p\}$ *be a set of lifted-outputs such that* $\mathcal{F}(\bar{\mathbf{Y}}^k) \in \bar{\mathcal{S}}_x \times \bar{\mathcal{S}}_u \ \forall k = 1, \ldots, p$. *Then under Assumptions 4.2((A)) and 4.3, it holds that* $\mathcal{F}(\bar{\mathbf{Y}}) = (\bar{x}, \bar{u}) \in \mathcal{X} \times \mathcal{U}$ *for any* $\bar{\mathbf{Y}} \in cvx(\{\bar{\mathbf{Y}}^1, \ldots, \bar{\mathbf{Y}}^p\})$.

Now suppose that we are given trajectory data $\{(\bar{x}_t, \bar{u}_t)\}_{t\geq 0}$ of the nominal system (4.2) such that

$$(\bar{x}_t, \bar{u}_t) \in \bar{\mathcal{S}}_x \times \bar{\mathcal{S}}_u, \forall t \geq 0, \ \text{dist}_{\mathcal{X}_G}(\bar{x}_t) \to 0. \tag{4.16}$$

Assemble output sequences $\bar{\mathbf{y}}_t = [\bar{y}_t, \ldots, \bar{y}_{t+R-1}] \ \forall t \geq 0$ and define the *Convex Output Safe Set* $\bar{\mathcal{CS}}_{\mathbf{y}}$ as the polytope

$$\bar{\mathcal{CS}}_{\mathbf{y}} = \text{cvx}(\bigcup_{t\geq 0} \bar{\mathbf{y}}_t). \tag{4.17}$$

By Assumption 4.2((A)), note that each $\bar{\mathbf{y}}_t$ uniquely identifies the nominal state $\bar{x}_t$ via the map (4.5) as $\mathcal{F}_x(\bar{\mathbf{y}}_t) = \bar{x}_t$, and the lifted-output $\bar{\mathbf{Y}}_t = [\bar{\mathbf{y}}_t, \bar{y}_{t+R}]$ uniquely identifies the nominal input $\bar{u}_t$ via the map (4.6) as $\mathcal{F}_u(\bar{\mathbf{Y}}_t) = \bar{u}_t$. We now show that the set $\bar{\mathcal{CS}}_{\mathbf{y}}$ is in fact, control invariant for the time-shift dynamics (4.18) and maps to nominal states and inputs within constraints (as depicted in Figure 4.1).

$$\begin{aligned} \bar{\mathbf{y}}_{t+1} &= [\bar{y}_{t+1}, \bar{y}_{t+2}, \ldots, \bar{y}_{t+R}] \\ &= \delta([\bar{y}_t, \bar{y}_{t+1}, \ldots, \bar{y}_{t+R-1}], \bar{y}_{t+R}) = \delta(\bar{\mathbf{y}}_t, \bar{y}_{t+R}) \end{aligned} \tag{4.18}$$

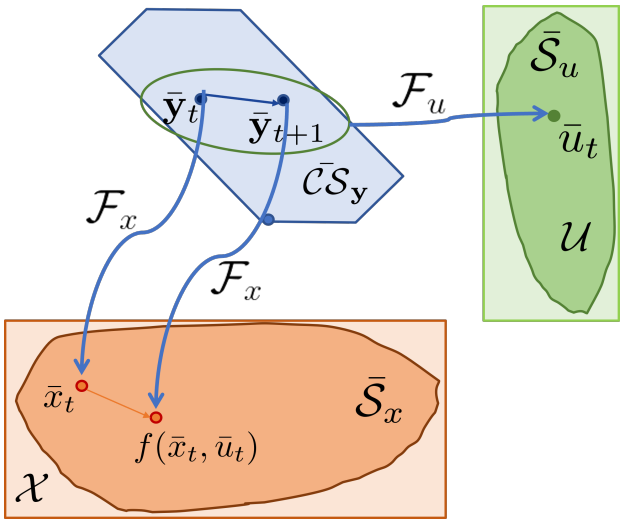


Figure 4.1: Illustration of the claim in Proposition 4.1.

Proposition 4.1 *Under Assumptions 4.2 and 4.3, the set* $\bar{\mathcal{CS}}_{\mathbf{y}}$ (4.17) *is control invariant for the forward-time shift dynamics* (4.18)*, i.e.,*

$$\forall \bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}_{\mathbf{y}}, \exists \bar{y}_{t+R} \ : \ \bar{\mathbf{y}}_{t+1} = \delta(\bar{\mathbf{y}}_t, \bar{y}_{t+R}) \in \bar{\mathcal{CS}}_{\mathbf{y}}, \tag{4.19}$$

and the corresponding nominal state and input are within constraints,

$$\begin{gathered} \bar{x}_t = \mathcal{F}_x(\bar{\mathbf{y}}_t) \in \mathcal{X}, \, \bar{u}_t = \mathcal{F}_u(\bar{\mathbf{y}}_t, \bar{y}_{t+R}) \in \mathcal{U}, \\ \mathcal{F}_x(\bar{\mathbf{y}}_{t+1}) = f(\bar{x}_t, \bar{u}_t) \in \mathcal{X}. \end{gathered} \tag{4.20}$$

Corollary 4.1 *A control invariant set in the state space can be implicitly defined using* $\bar{\mathcal{CS}}_{\mathbf{y}}$ *as*

$$\bar{\mathcal{X}}_N := \{\bar{x} \in \mathbb{R}^n \, | \exists \bar{\mathbf{y}} \in \bar{\mathcal{CS}}_{\mathbf{y}}, \ \mathcal{F}_x(\bar{\mathbf{y}}) = \bar{x}\}. \tag{4.21}$$

The set $\bar{\mathcal{CS}}_{\mathbf{y}}$ can be recursively enlarged using output sequences $\bar{\mathbf{y}}'_t$ of another trajectory $\{(\bar{x}'_t, \bar{u}'_t)\}_{t\geq 0}$ satisfying (4.16), as $\bar{\mathcal{CS}}_{\mathbf{y}} \leftarrow \text{cvx}(\bar{\mathcal{CS}}_{\mathbf{y}} \cup (\bigcup_{t\geq 0} \bar{\mathbf{y}}'_t))$.

Remark 4.3 *Note that the set* (4.21) *is constructed without any local linear approximations of the nominal dynamics* (4.2)*, and uses the nonlinear dynamics implicitly via the data* $\{\bar{\mathbf{y}}_t\}_{t\geq 0}$ *and map* $\mathcal{F}(\cdot)$*.*

Example 4.2 *Consider the 2-state nonlinear system*

$$\begin{bmatrix} \bar{x}_{t+1}[0] \\ \bar{x}_{t+1}[1] \end{bmatrix} = \begin{bmatrix} \bar{x}_t[0] + dt(\bar{x}_t[0] - \bar{x}_t^3[1]) \\ \bar{x}_t[1] + dt(\bar{x}_t^2[0] - \bar{u}_t) \end{bmatrix}$$

with flat output $\bar{y}_t = \bar{x}_t[0]$*, lifted-output* $\bar{\mathbf{Y}}_t = [\bar{y}_t, \bar{y}_{t+1}, \bar{y}_{t+2}]$ *and the map* $\mathcal{F}_x(\cdot)$ *from Assumption 4.2*((A)) *as* $\mathcal{F}_x(\bar{\mathbf{y}}_t) = [\bar{y}_t, (\frac{1}{dt}((1+dt)\bar{y}_t - \bar{y}_{t+1}))^{\frac{1}{3}}]$*. We collect two trajectories for this system converging to distinct equilibria and construct the safe set* $\bar{\mathcal{CS}}_{\mathbf{y}}$ (4.17) *and corresponding set* $\mathcal{X}_f$ *using* (4.21)*. To depict control invariance, we start from a state in* $\mathcal{X}_f$ *and choose input* u_t *using* (4.20) *by finding* $\bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}_{\mathbf{y}}$ *such that* $\mathcal{F}_x(\bar{\mathbf{y}}_t) = \bar{x}_t$*.*

Now we proceed to construct a convex Control Lyapunov Function $\bar{Q}(\cdot)$ on the set $\bar{\mathcal{CS}}_{\mathbf{y}}$ using trajectories of the nominal system that satisfy (4.16). To do so, we require a stage cost function $c(\dot{)}$ to penalize deviations of the system trajectory from $\mathcal{X}_G$. Construct the following box sets, $\bar{\mathcal{X}}_G = \{x | \tilde{lb}_x^g \leq x \leq \tilde{ub}_x^g\} \subset \mathcal{X}_G$, and $\mathcal{U}_{nom} = \{u | \tilde{lb}_u^{nom} \leq u \leq \tilde{ub}_u^{nom}\} \subset \mathcal{U}$ to define the convex set in the lifted output space,

$$\mathcal{Y}_G = \left\{ \mathbf{Y} \middle| \begin{matrix} \mathbf{Y} = [\mathbf{y}, y], \ \tilde{lb}_x^g \leq \mathcal{F}_x^{\cap}(\mathbf{y}), \mathcal{F}_x^{\cup}(\mathbf{y}) \leq \tilde{ub}_x^g, \\ \tilde{lb}_u^{nom} \leq \mathcal{F}_u^{\cap}(\mathbf{Y}), \mathcal{F}_u^{\cup}(\mathbf{Y}) \leq \tilde{ub}_u^{nom} \end{matrix} \right\}. \tag{4.22}$$

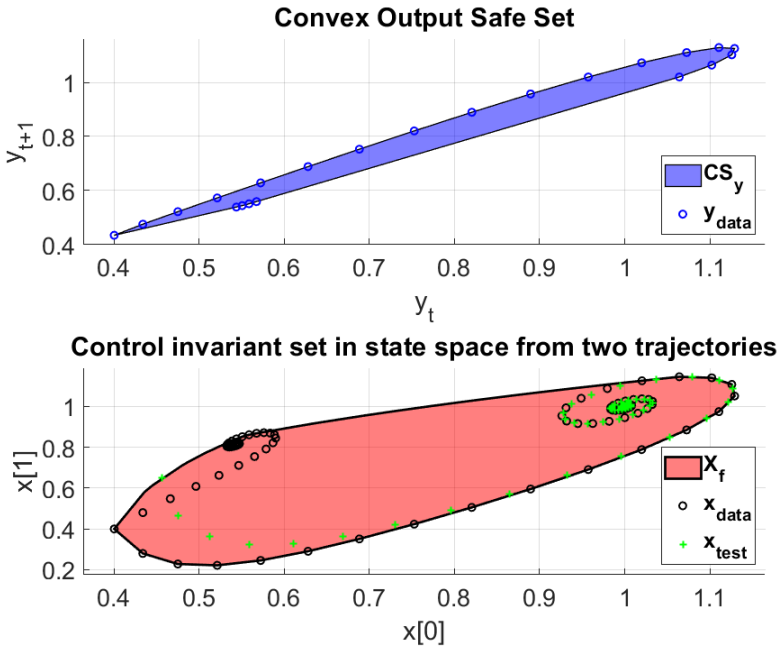


Figure 4.2: Control invariant set construction in example 4.2 from two trajectories (black). Notice that the test trajectory (green) starts outside of the convex hull of the collected trajectory data.

Observe that $\bar{\mathbf{Y}} \in \mathcal{Y}_G \Rightarrow \mathcal{F}(\bar{\mathbf{Y}}) = (\bar{x}, \bar{u}) \in \mathcal{X}_G \times \mathcal{U}_{nom}$ by Assumption 4.2((A)). Now let $c(\cdot)$ be a convex, continuous function satisfying

$$c(\bar{\mathbf{Y}}) = 0 \; \forall \bar{\mathbf{Y}} \in \mathcal{Y}_G, \; c(\bar{\mathbf{Y}}) \succ 0 \; \forall \bar{\mathbf{Y}} \in \mathbb{R}^{m \times R+1} \backslash \mathcal{Y}_G \tag{4.23}$$

for penalizing deviations from the set $\mathcal{Y}_G$. Define the *cost-to-go* at time t corresponding to $\bar{\mathbf{y}}_t$,

$$\mathcal{C}_t = \sum_{k \geq t} c(\bar{\mathbf{Y}}_k). \tag{4.24}$$

We construct the function $\bar{Q}(\cdot)$ on $\bar{\mathcal{CS}}_{\mathbf{y}}$ as follows. Denote $\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})$ as the set of vertices of the polytope $\bar{\mathcal{CS}}_{\mathbf{y}}$, and let $\bar{\mathbf{y}}_{t_i} \in \text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})$ for $i = 1, \ldots, |\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|$, with $\{(\bar{x}_k, \bar{u}_k), \bar{\mathbf{Y}}_k\}_{k \geq t_i}$ being the corresponding state, input and lifted-output trajectory satisfying (4.16). Then for any $\bar{\mathbf{y}} \in \bar{\mathcal{CS}}_{\mathbf{y}}$, we define $\bar{Q}(\bar{\mathbf{y}})$ via linear interpolation:

$$\begin{aligned} \bar{Q}(\bar{\mathbf{y}}) = \min_{\substack{\lambda_i \in [0,1], \\ \forall i = 1, \ldots, |\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|}} & \sum_{i=1}^{|\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|} \lambda_i \mathcal{C}_{t_i} \\ \text{s.t.} & \sum_{i=1}^{|\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|} \lambda_i \begin{bmatrix} \bar{\mathbf{y}}_{t_i} \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{y}} \\ 1 \end{bmatrix}. \end{aligned} \tag{4.25}$$

To ensure that $\bar{Q}(\cdot)$ takes finite values on $\bar{\mathcal{CS}}_{\mathbf{y}}$, the trajectory $\{(\bar{x}_t, \bar{u}_t)\}_{t\geq 0}$ must also satisfy $\mathcal{C}_0 < \infty$, formalized in the following assumption.

Assumption 4.4 *We are provided with a nominal state-input trajectory $\{(\bar{x}_t, \bar{u}_t)\}_{t\geq 0}$ of system (4.2) satisfying*

1. $(\bar{x}_t, \bar{u}_t) \in \bar{\mathcal{S}}_x \times \bar{\mathcal{S}}_u, \forall t \geq 0$, $dist_{\mathcal{X}_G}(\bar{x}_t) \to 0$,
2. $\mathcal{C}_0 < \infty$,

for defining $\bar{\mathcal{CS}}_{\mathbf{y}}$ as (4.17) and $\bar{Q}(\cdot)$ as (4.25).

The function $\text{dist}_S(x) = \min_{y\in S} = ||x - y||_2$ defines the distance of x from the set S. The following proposition identifies CLF-like characteristics of the function (4.25) on the set $\bar{\mathcal{CS}}_{\mathbf{y}}$ which we will use for convergence analysis in Section 4.6.

Proposition 4 *Given Assumptions 4.2 and 4.4, the cost function $\bar{Q}(\cdot)$ satisfies the following properties:*

1. *$\bar{Q}(\cdot)$ is convex, non-negative, and equal to 0 only on the set given by*
$$cvx\left(\{\bar{\mathbf{y}}_t | \bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}_{\mathbf{y}}, \mathcal{F}_x(\bar{\mathbf{y}}_t) \in \mathcal{X}_G\}\right)$$
2. *$\bar{Q}(\bar{\mathbf{y}}_{t+1}) - \bar{Q}(\bar{\mathbf{y}}_t) \leq -c(\bar{\mathbf{Y}}_t)$, $\forall \bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}_{\mathbf{y}}$ where $\bar{\mathbf{y}}_{t+1} = \delta(\bar{\mathbf{y}}_t, \bar{y}_{t+R})$ as in (4.18) and $\bar{\mathbf{Y}}_t = [\bar{\mathbf{y}}_t, \bar{y}_{t+R}]$.*

The above proposition shows that $\bar{Q}(\cdot)$ is a CLF for the dynamics $\bar{\mathbf{y}}_{t+1} = \delta(\bar{\mathbf{y}}_t, \bar{y}_{t+R})$ with input $\bar{y}_{t+R}$ on the set $\mathcal{CS}_{\mathbf{y}}$.

Corollary 4.2 *A Control Lyapunov Function for the nominal system (4.2) is given by*

$$P(\bar{x}) := \{C \in \mathbb{R} | \exists \bar{\mathbf{y}} \in \bar{\mathcal{CS}}_{\mathbf{y}}, \mathcal{F}_x(\bar{\mathbf{y}}) = \bar{x}, \bar{Q}(\bar{\mathbf{y}}) = C\}, \tag{4.26}$$

for which, $P(\bar{x}_{t+1}) - P(\bar{x}_t) \leq -c(\bar{\mathbf{Y}}_t)$, where $(\bar{x}_t, \bar{u}_t) = \mathcal{F}(\bar{\mathbf{Y}}_t)$ and $\bar{x}_{t+1} = f(\bar{x}_t, \bar{u}_t)$.

4.4 Robustification of Constraints

For constructing the tightened constraints $\bar{\mathcal{X}}, \bar{\mathcal{U}}$, we first require the error invariant set $\mathcal{E}$. Suppose we are given trajectory data $\{x_t, u_t, d_t\}_{t\geq 0}$ of the system (4.1). Using Assumption 4.1 and Theorem 2.1 from Chapter 2, we can obtain the ellipsoidal set $\hat{\mathcal{D}} = E^D(\mathcal{X}, \mathcal{U})$ such that $\hat{\mathcal{D}} \supset D(\mathcal{X}, \mathcal{U})$. We now construct the error invariant set $\mathcal{E}$ given a fixed error policy $\kappa(\cdot)$. Let us conservatively model the true system (4.1) as the following uncertain system

$$\begin{aligned} x_{t+1} &= f(x_t, \bar{u}_t + \kappa(x_t - \bar{x}_t)) + d_t \\ \bar{x}_{t+1} &= f(\bar{x}_t, \bar{u}_t) \end{aligned} \tag{4.27}$$

where d_t is the process noise with support $\hat{\mathcal{D}}$. Define the corresponding error system with error state $e_t = x_t - \bar{x}_t$ and uncertain dynamics

$$e_{t+1} = f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t \tag{4.28}$$

where $f_e(\cdot)$ is defined as in (4.8). Let $\mathcal{E}$ be a Robust Positive Invariant (RPI) set for (4.28):

$$\begin{aligned}&\forall \bar{x}_t \in \bar{\mathcal{X}}, \forall \bar{u}_t \in \bar{\mathcal{U}} : \\ &e_t \in \mathcal{E} \Rightarrow e_{t+1} = f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t \in \mathcal{E},\ \forall d_t \in \hat{\mathcal{D}}.\end{aligned}$$

Constructing RPI $\mathcal{E}$ and error policy $\kappa(\cdot)$ for nonlinear systems is difficult in general. However under additional assumptions on $f(\cdot)$ (cf. smoothness, Lipchitz continuity, incremental stabilizability), it is common in the nonlinear MPC literature to fix a policy $\kappa(e_t)$ and compute $\mathcal{E}$, or compute both jointly [108, 141, 126]. In view of this, we make the following assumption to construct the error invariant for the *actual* error dynamics (4.8) in Proposition 4.2.

Assumption 4.5 *Given disturbance support $\hat{\mathcal{D}} \supset \mathcal{D}$ and a known, fixed linear policy $\kappa(e_t) = Ke_t$, the RPI set $\mathcal{E}$ for* (4.28) *can be computed such that*

$$\begin{aligned}&\forall \bar{x}_t \in \bar{\mathcal{X}}, \forall \bar{u}_t \in \bar{\mathcal{U}} : \\ &e_t \in \mathcal{E} \Rightarrow e_{t+1} = f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t \in \mathcal{E},\ \forall d_t \in \hat{\mathcal{D}}.\end{aligned}$$

Proposition 4.2 *Let Assumption 4.5 hold and $\mathcal{E}$ be a Robust Positive Invariant set for* (4.28) *with $\kappa(e_t) = Ke_t$ for some disturbance support $\hat{\mathcal{D}} \supset \mathcal{D}$. Then for the true error system given by* (4.8) *with error state $e_t = x_t - \bar{x}_t$, we have*

$$\begin{aligned}&\forall \bar{x}_t \in \mathcal{X} \ominus \mathcal{E}, \forall \bar{u}_t \in \mathcal{U} \ominus K\mathcal{E} : \\ &e_t \in \mathcal{E} \Rightarrow e_{t+1} = f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t \in \mathcal{E}, \forall d_t \in D(e_t + \bar{x}_t, u_t).\end{aligned}$$

Remark 4.4 *In theory, the difference flatness of $f(\cdot)$ can be used for constructing the RPI. All difference flat systems are linearizable by a dynamic feedback control [9], i.e., there exists a continuous map $T(\cdot)$ and extended state and input space $(z, v) \in \mathbb{R}^{n'+m}$, with $n' > n$ such that $T(x, u) = (z, v)$, where the dynamics of (z, v) are linear. Then, mapping the uncertainty support $\mathcal{D}$ from (x, u) dynamics to the (z, v) dynamics is possible (although non-trivial) by over-approximations of T. However, this approach will further increase conservatism in general.*

Given the error invariant $\mathcal{E}$ and policy $\kappa(e_t) = Ke_t$, we construct the tightened constraints with the properties:

(i) $\bar{\mathcal{X}} \subset \bar{\mathcal{S}}_x$ and $\bar{\mathcal{U}} \subset \bar{\mathcal{S}}_u$ so that the invariant set $\bar{\mathcal{X}}_N$ and terminal cost $P(\cdot)$ can be constructed from trajectory data of the nominal system.

(ii) $\bar{\mathcal{X}} \subset \mathcal{X} \ominus \mathcal{E}$ and $\bar{\mathcal{U}} \subset \mathcal{U} \ominus \kappa(\mathcal{E})$ so that $\bar{x}_t \in \bar{\mathcal{X}}$, $\bar{u}_t \in \bar{\mathcal{U}}$, $e_t \in \mathcal{E} \Rightarrow x_t \in \mathcal{X}$, $u_t \in \mathcal{U}$ for system (4.1).

We construct these tightened sets in two steps. In the first step, we solve the following nonlinear program

$$\begin{aligned} \max_{\alpha_x \geq 1, \alpha_u \geq 1, v_x, v_u} \quad & \alpha_x + \alpha_u \\ \text{s.t} \quad & \alpha_x \mathcal{X} + v_x \subseteq \bar{\mathcal{S}}_x, \\ & \alpha_u \mathcal{U} + v_u \subseteq \bar{\mathcal{S}}_u \end{aligned} \tag{4.29}$$

to compute polytopic inner-approximations of $\bar{\mathcal{S}}_x, \bar{\mathcal{S}}_u$ as the intermediate state and input box sets

$$\begin{aligned} \hat{\mathcal{X}} &= \alpha_x^\star \mathcal{X} + v_x^\star, \\ \hat{\mathcal{U}} &= \alpha_u^\star \mathcal{U} + v_u^\star. \end{aligned} \tag{4.30}$$

Finally, the tightened state and input constraints are defined as

$$\begin{aligned} \bar{\mathcal{X}} &= \hat{\mathcal{X}} \ominus \mathcal{E}, \\ \bar{\mathcal{U}} &= \hat{\mathcal{U}} \ominus K\mathcal{E}. \end{aligned} \tag{4.31}$$

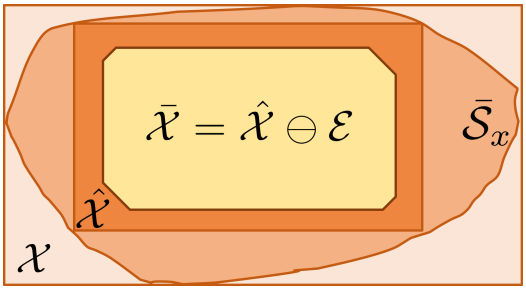


Figure 4.3: Construction of tightened state constraints $\bar{\mathcal{X}}$ via (4.29), and (4.31). The set $\bar{\mathcal{S}}_x \subset \mathcal{X}$ is defined so that $lb_x \leq \mathcal{F}_x^{\cap}(\bar{\mathbf{Y}}) \leq \mathcal{F}_x(\bar{\mathbf{Y}}) = \bar{x} \leq \mathcal{F}_x^{\cup}(\bar{\mathbf{Y}}) \leq ub_x$, required for Lemma 4.1

Remark 4.5 *To enforce $\alpha_x \mathcal{X} + v_x \subseteq \bar{\mathcal{S}}_x$ in* (4.29)*, the $n-1$ dimensional facets of $\alpha_x \mathcal{X} + v_x$ are gridded and each grid point is constrained to lie within $\bar{\mathcal{S}}_x$. which is non-convex but simply connected (which can be shown by exploiting the continuity, surjectivity of $\mathcal{F}(\cdot)$ and convexity of the set $\{\bar{\mathbf{y}} | lb_x^j \leq \mathcal{F}_x^{\cup}(\bar{\mathbf{y}}), \mathcal{F}_x^{\cap}(\bar{\mathbf{y}}) \leq ub_x^j\}$ [93, Chapter 9]). The constraints for $\hat{\mathcal{U}}$ are enforced similarly.*

4.5 Iterative Robust MPC Design

Now we combine the elements from sections 4.3 and 4.4 to propose an iterative Robust MPC algorithm for

1. driving the system (4.1) to the goal set $\mathcal{X}_G$ while robustly satisfying constraints,
2. improving trajectory costs iteratively,
3. collecting trajectory data for learning the control invariant set $\bar{\mathcal{CS}}_{\mathbf{y}}$, the CLF $\bar{Q}(\cdot)$, and the uncertainty support $\hat{\mathcal{D}}$.

An iteration as defined as a rollout of system (4.1) starting from a fixed state x_S with some policy $\pi(\cdot)$ such that system state and input remain within constraints, and the system state is asymptotically steered to $\mathcal{X}_G$. Formally, at iteration j:

$$\begin{aligned} \forall t \geq 0: \quad & x_{t+1}^j = f(x_t^j, u_t^j) + d_t^j,\ d_t^j \in D(x_t^j, u_t^j) \\ & x_0^j = x_S,\ x_t^j \in \mathcal{X},\ u_t^j = \pi^j(x_t) \in \mathcal{U}, \\ & \text{dist}_{\mathcal{X}_G}(x_t^j) \to 0 \end{aligned} \tag{4.32}$$

We denote $\pi^j(\cdot)$ as the policy for iteration j and x_t^j, u_t^j are the state and input of (4.1) respectively at time t. The quantities for the nominal system (4.2) are similarly denoted as $\bar{x}_t^j, \bar{u}_t^j, \bar{y}_t^j$.

At iteration j, we use trajectory data $\{\{(x_t^i, u_t^i)\}_{t\geq 0}\}_{i=0}^{j-1}$ to construct outer-approximation $\hat{\mathcal{D}}^j$ of the disturbance support $D(\mathcal{X}, \mathcal{U})$ by solving (2.11) offline, which is used for computing the error invariant $\mathcal{E}^j$. Then the tightened state and input constraints $\bar{\mathcal{X}}^j, \bar{\mathcal{U}}^j$ are constructed using (4.31) with $\mathcal{E}^j$. The terminal constraint is defined as in (4.21) but with $\bar{\mathcal{CS}}_{\mathbf{y}}^{j-1} = \text{cvx}(\{\bigcup_{t\geq 0} \bar{\mathbf{y}}_t\}_{i=0}^{j-1})$. The stage cost $c(\cdot)$ is chosen as in (4.23), and the terminal cost $\bar{Q}^{j-1}(\cdot)$ (4.26) is defined on $\bar{\mathcal{CS}}_{\mathbf{y}}^{j-1}$. Like the forward-shift operator (4.18), we define the backward-time shift operator,

$$\begin{aligned} \bar{\mathbf{Y}}_t &= [\bar{y}_t, \ldots, \bar{y}_{t+R}] \\ &= \delta^-([\bar{y}_{t+1}, \bar{y}_{t+1}, \ldots, \bar{y}_{t+R+1}], \bar{y}_t) \\ &= \delta^-(\bar{\mathbf{Y}}_{t+1}, \bar{y}_t) \end{aligned} \tag{4.33}$$

Employing this definition, the optimization problem for Robust Output-Lifted LMPC is

given by the following:

$$
\begin{aligned}
J_t^j(x_t^j) = \min_{\substack{\bar{\boldsymbol{u}}_t^j, \bar{\boldsymbol{x}}_t^j, \\ \bar{\mathbf{y}}_{t+N|t}^j}} \quad & \bar{Q}^{j-1}(\bar{\mathbf{y}}_{t+N|t}^j) + \sum_{k=t}^{t+N-1} c(\bar{\mathbf{Y}}_{k|t}^j) \\
\text{s.t.} \quad & \bar{x}_{k+1|t}^j = f(\bar{x}_{k|t}^j, \bar{u}_{k|t}^j), \\
& \bar{x}_{k|t}^j \in \bar{\mathcal{X}}^j, \bar{u}_{k|t}^j \in \bar{\mathcal{U}}^j \ \forall k \in \mathbb{I}_t^{t+N-1}, \\
& \bar{\mathbf{Y}}_{k|t}^j = \delta^-(\bar{\mathbf{Y}}_{k+1|t}^j, h(\bar{x}_{k|t}^j)) \ \forall k \in \mathbb{I}_t^{t+N-2}, \\
& \bar{x}_{t+N|t}^j = \mathcal{F}_x(\bar{\mathbf{y}}_{t+N|t}^j), \ \bar{\mathbf{y}}_{t+N|t}^j \in \bar{\mathcal{CS}}_{\mathbf{y}}^{j-1}, \\
& \bar{\mathbf{Y}}_{t+N-1|t}^j = [h(\bar{x}_{t+N-1|t}^j), \bar{\mathbf{y}}_{t+N|t}^j], \\
& x_t^j - \bar{x}_{t|t}^j \in \mathcal{E}^j, \bar{x}_{t|t}^j = \bar{x}_{t|t-1}^{j\star}
\end{aligned}
\tag{4.34}
$$

where $\boldsymbol{u}_t^j = [\bar{u}_{t|t}^j, \ldots, \bar{u}_{t+N-1|t}^j]$, $\boldsymbol{x}_t^j = [\bar{x}_{t|t}^j, \ldots, \bar{x}_{t+N|t}^j]$ and x_t^j is the state of the system at time t. The control policy is obtained by solving (4.34) online and using the optimal solution $\bar{u}_t^j = \bar{u}_{t|t}^{j\star}, \bar{x}_t^j = \bar{x}_{t|t}^{j\star}$, as

$$
u_t^j = \pi^j(x_t^j) = \bar{u}_t^j + K(x_t^j - \bar{x}_t^j) \tag{4.35}
$$

The optimal nominal state-input trajectory $\{(\bar{x}_{t|t}^{j\star}, \bar{u}_{t|t}^{j\star})\}_{t\geq 0}$ is used for constructing the set $\bar{\mathcal{CS}}_{\mathbf{y}}^j$ and function $\bar{Q}^j(\cdot)$ for iteration $j+1$. We summarize the iterative policy synthesis in Algorithm 2. To initialize the iteration, we require a slight modification of Assumption 4.4.

Assumption 4.6 (Modification of Assumption 4.4) *We are provided with a state-input trajectory $\{(x_t^0, u_t^0)\}_{t\geq 0}$ of system* (4.1) *satisfying* (4.32) *and a nominal state-input trajectory $\{(\bar{x}_t^0, \bar{u}_t^0)\}_{t\geq 0}$ satisfying*

1. $(\bar{x}_t^0, \bar{u}_t^0) \in \bar{\mathcal{X}}^1 \times \bar{\mathcal{U}}^1, \forall t \geq 0$, $dist_{\mathcal{X}_G}(\bar{x}_t) \to 0$,
2. $\mathcal{C}_0 < \infty$,

for defining $\bar{\mathcal{CS}}_{\mathbf{y}}^0$ as (4.17) *and $\bar{Q}^0(\cdot)$ as* (4.25).

Remark 4.6 *The nominal state $\bar{x}_t^j = x_{t|t}^{j\star}$ for* (4.34) *at time t is obtained from the solution of* (4.34) *at time $t-1$. Consequently, the solution to* (4.34) *is the same for any $x_t^j \in \mathcal{E}^j \oplus x_{t|t}^{j\star}$. To*

Algorithm 2: Iterative Robust Output-lifted LMPC

Input : $\{(x_t^0, u_t^0)\}_{t\geq 0}$, $\{(\bar{x}_t^0, \bar{u}_t^0)\}_{t\geq 0}$ satisfying Assumption 4.4,
Error policy $\kappa(e_t) = Ke_t$, max_iters

Output: $\{\{(x_t^j, u_t^j)\}_{t\geq 0}\}_{j=1}^{\text{max\_iters}}$ satisfying (4.32)

Procedure `Robust_Output-lifted_LMPC`**:**

- $\bar{\mathcal{CS}}_{\mathbf{y}}^0 \leftarrow \emptyset$
- $\hat{\mathcal{X}}, \hat{\mathcal{U}} \leftarrow$ Use (4.3) after solving (4.29)
- **for** $j = 1$ **to** *max_iters* **do**
  - ***(Offline)***
  - **Step 1**: Construct $\hat{\mathcal{D}}^j$ by solving (2.11) and compute $\mathcal{E}^j$ by Assumption 4.5
  - **Step 2**: Construct $\bar{\mathcal{X}}^j, \bar{\mathcal{U}}^j$ using (4.31)
  - **Step 3**: Construct $\bar{\mathcal{CS}}_{\mathbf{y}}^j$ like in (4.17) and $\bar{Q}^j(\cdot)$ from (4.25)
  - ***(Online)***
  - $x_0^j \leftarrow x_S, t \leftarrow 0$
  - **while** $x_t^j \notin \mathcal{X}_G$ **do**
    - Solve (4.34) and store $\bar{x}_t^j \leftarrow \bar{x}_{t|t}^{j\star}, \bar{u}_{t-1}^j \leftarrow \bar{u}_{t-1|t}^{j\star}$
    - Apply (4.35) and store $u_t^j \leftarrow \pi_t^j(x_t^j)$
    - Measure x_{t+1}^j
    - $t \leftarrow t + 1$
  - **end while**
- **end for**

return $\{\{(x_t^j, u_t^j)\}_{t\geq 0}\}_{j=1}^{\textbf{max\_iters}}$

incorporate feedback from x_t^j, the constraints $x_t^j - \bar{x}_{t|t}^j \in \mathcal{E}^j$, $\bar{x}_{t|t}^j = \bar{x}_{t|t-1}^{j\star}$ can be replaced with $x_t^j - \bar{x}_{t|t}^j \in \mathcal{E}^j$, $\bar{x}_{t|t}^j = f(\bar{x}_{t-1|t-1}^{j\star}, \bar{u}_{t-1|t}^j), \bar{u}_{t-1|t}^j \in \bar{\mathcal{U}}^j$, where the nominal state $\bar{x}_t^j = x_{t|t}^{j\star}$ and nominal input $\bar{u}_{t-1}^j = \bar{u}_{t-1|t}^{j\star}$ are re-computed to generate a nominal state-input trajectory for constructing the control invariant $\bar{\mathcal{CS}}_{\mathbf{y}}^j$.

4.6 Properties of Proposed Strategy

In this section, we establish the closed-loop properties of the system trajectories with the proposed Robust Output-lifted LMPC, and examine the system performance across iterations.

Theorem 4.1 establishes the recursive feasibility of optimization problem (4.34) for system (4.1) in closed-loop with the LMPC policy (4.35). We show this by leveraging the recursive definition of $\bar{\mathcal{CS}}_{\mathbf{y}}^j$ and the result of Proposition 4.1.

Theorem 4.1 *[**Recursive Feasibility**] Given Assumptions 4.1-4.6, the optimization problem* (4.34) *is feasible for the system* (4.1) *in closed-loop with the policy* (4.35) $\forall t \geq 0$ *and for all iterations* $j \geq 1$. *Consequently,* $x_t^j \in \mathcal{X}$, $u_t^j \in \mathcal{U}$ $\forall t \geq 0$, $\forall j \geq 1$.

We also establish convergence of the closed-loop state trajectories of (4.1) to the set $\mathcal{X}_G$. First, we show that if $\lim_{t\to\infty} \text{dist}_{\mathcal{Y}_G}(\bar{\mathbf{Y}}_t^j) = 0$ then $\lim_{t\to\infty} \text{dist}_{\mathcal{X}_G \ominus \mathcal{E}^1}(\bar{x}_t^j) = 0$. We use this result to finally show that $\lim_{t\to\infty} \text{dist}_{\mathcal{X}_G}(x_t^j) = 0$ in the proof of Theorem 4.2.

Lemma 4.2 *Given Assumption 4.2, if the trajectory of lifted outputs* $\{\bar{\mathbf{Y}}_t\}_{t\geq 0}$ *for system* (4.2) *converges to the set* $\mathcal{Y}_G$, *then the nominal state trajectory* $\{\bar{x}_t\}_{t\geq 0}$ *converges to* $\mathcal{X}_G \ominus \mathcal{E}^1$, *i.e.,*

$$\lim_{t\to\infty} dist_{\mathcal{Y}_G}(\bar{\mathbf{Y}}_t^i) = 0 \Rightarrow \lim_{t\to\infty} dist_{\mathcal{X}_G \ominus \mathcal{E}^1}(\bar{x}_t) = 0$$

Theorem 4.2 *[**Convergence**] Given Assumptions 4.1-4.6, for any iteration* $j \geq 1$, *the state trajectory of system* (4.1) *in closed-loop with control* (4.35) *converges to the set* $\mathcal{X}_G$, *i.e.,*

$$\lim_{t\to\infty} dist_{\mathcal{X}_G}(x_t^j) = 0.$$

We conclude our theoretical analysis of the proposed Robust MPC (4.35) with the following theorem. We state and prove that the closed-loop costs of system trajectories in closed-loop with the LMPC do not increase with iterations if the system starts from the same state, i.e., $x_0^j = x_S$ $\forall j \geq 0$.

Theorem 4.3 *[**Performance Improvement**] Given Assumptions 2.1-4.4, the cost of the trajectories of system* (4.1) *in closed-loop with control* (4.35) *does not increase with iterations,*

$$j_2 > j_1 \Rightarrow J_{0\to\infty}^{j_2}(x_S) \leq J_{0\to\infty}^{j_1}(x_S)$$

where $J_{0\to\infty}^j(x_S) = \mathcal{C}_0^j$, *the cost of the jth iteration.*

4.7 Numerical Example: Kinematic Bicycle in Frenet Frame

In this section, we demonstrate our approach for constrained optimal control of a kinematic bicycle in the Frenet frame. The code for this example is hosted at `https://github.com/shn66/ROLMPC`.

Problem Formulation

We solve a constrained optimal control problem for driving a kinematic bicycle over a chicane into a goal set $\mathcal{X}_G$. The dynamics $f(\cdot)$ of the bicycle are described in the Frenet frame,

$$\begin{aligned} s_{t+1} &= s_t + \frac{v_t \cos(e_{\psi t})}{1 - e_{y,t} C(s_t)} \mathrm{dt} + d_{1,t} \\ e_{y,t+1} &= e_{y,t} + v_t \sin(e_{\psi,t}) \mathrm{dt} + d_{2,t} \\ e_{\psi,t+1} &= e_{\psi,t} + v_t \left(\frac{\tan(\delta_t)}{L_{RF}} - \frac{\cos(e_{\psi t}) C(s_t)}{1 - e_{y,t} C(s_t)} \right) \mathrm{dt} + d_{3,t} \end{aligned}$$

where time-step $\mathrm{dt} = 0.2s$, and state $x_t = [s_t, e_{y,t}, e_{\psi,t}]$ consists of the longitudinal abcissa s_t, the lateral offset $e_{y,t}$ and the heading alignment error $e_{\psi,t}$ w.r.t the road center-line. The disturbance $d_t = [d_{1,t}, d_{2,t}, d_{3,t}]$ captures bounded process noise, errors from discretization and conversion between the Euclidean and Frenet frame. The inputs $u_t = [v_t, \delta_t]$ are the speed v_t of the rear axle and steering angle δ_t of the front axle. $L_{RF} = 4m$ is the wheelbase of the vehicle, and it is assumed that the centre of gravity is on the rear axle. The center-line is given by a chicane with curvature $C(s_t) = \frac{1}{10\pi} \tan^{-1}(100 - \frac{1}{2} s_t^2)$. The constraints are given by the box sets as in Assumption 4.3:

$$\begin{aligned} \mathcal{X} &= \left\{ [s, e_y, e_\psi] \middle| s \in [-2, 60], e_y \in [-4.5, 4.5], e_\psi \in [-\frac{\pi}{3}, \frac{\pi}{3}] \right\} \\ \mathcal{U} &= \left\{ [v, \delta] \middle| \; v \in [0, 18], \delta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \right\}, \end{aligned}$$

and the goal set is $\mathcal{X}_G = \{[s, e_y, e_\psi] \in \mathcal{X} | s \geq 40\}$.

We use our Robust Output-lifted LMPC to iteratively approximate the solution of the following optimal control problem

$$\begin{aligned} \min_{\{\pi_t(\cdot)\}_{t \geq 0}} \quad & \sum_{t=0}^{\infty} \max\{40 - s_t, 0\} \\ \text{s.t.} \quad & x_{t+1} = f(x_t, \pi_t(x_t)) + d_t, \\ & x_t \in \mathcal{X}, \; \pi_t(x_t) \in \mathcal{U}, \\ & x_0 = x_S \end{aligned} \tag{4.36}$$

for the kinematic bicycle starting from $x_S = [0, 1, 0]$. To apply Algorithm 2, we verify that Assumptions 2.1, 4.2 and 4.5 are satisfied. First, we describe the lifted output $\tilde{\mathbf{Y}}$ and associated maps $\mathcal{F}_x(\cdot), \mathcal{F}_u(\cdot)$ for the kinematic bicycle, and obtain the bounding functions $\mathcal{F}^{\cup}(\cdot), \mathcal{F}^{\cap}(\cdot)$ for verifying Assumption 4.2. Second, we obtain an uncertainty description (as in Assumption 2.1) for the additive disturbance d_t using data. To verify Assumption 4.5, we describe a procedure for constructing the error invariant $\mathcal{E}$ for the error dynamics (4.8), and choosing a fixed linear policy $\kappa(e_t) = Ke_t$. The three steps are detailed below.

Lifted Outputs

The difference flat output for the nominal kinematic bicycle model are given by $\bar{s}_t, \bar{e}_{y,t}$. The lifted output and associated maps for the nominal system are given by

$$\bar{y}_t = [\bar{y}_{1,t}\ \bar{y}_{2,t}]^\top = [\bar{s}_t\ \bar{e}_{y,t}]^\top,\ \bar{\mathbf{Y}}_t = [\bar{y}_t, \bar{y}_{t+1}, \bar{y}_{t+2}]$$
$$\mathcal{F}_x(\bar{y}_t, \bar{y}_{t+1}) = \begin{bmatrix} \bar{y}_t \\ \tan^{-1}\left(\frac{\bar{y}_{2,t+1} - \bar{y}_{2,t}}{(1-\bar{y}_{2,t}C(\bar{y}_{1,t}))(\bar{y}_{1,t+1} - \bar{y}_{1,t})}\right) \end{bmatrix}$$
$$\mathcal{F}_u(\bar{y}_t, \bar{y}_{t+1}, \bar{y}_{t+2}) = [\bar{v}_t\ \bar{\delta}_t]^\top$$

$$\bar{v}_t = \frac{1}{dt}\left\| \begin{bmatrix} (1-\bar{y}_{2,t}C(\bar{y}_{1,t}))(\bar{y}_{1,t+1} - \bar{y}_{1,t}) \\ \bar{y}_{2,t+1} - \bar{y}_{2,t} \end{bmatrix} \right\|_2$$
$$\bar{\delta}_t = \tan^{-1}\Big([0\ 0\ \frac{L_{RF}}{dt\bar{v}_t}](\mathcal{F}_x(\bar{y}_{t+1}, \bar{y}_{t+2}) - \mathcal{F}_x(\bar{y}_t, \bar{y}_{t+1}))$$
$$(\bar{y}_{1,t+1} - \bar{y}_{1,t})C(\bar{y}_{1,t})\Big)$$

Next, we propose bounding functions $\mathcal{F}^\cup(\cdot), \mathcal{F}^\cap(\cdot)$ such that they are quasiconvex and quasiconcave respectively, with $\mathcal{F}^\cap(\bar{\mathbf{Y}}) \leq \mathcal{F}(\bar{\mathbf{Y}}) \leq \mathcal{F}^\cup(\bar{\mathbf{Y}})$, as required by Assumption 4.2((A)). For the nominal positions $\bar{s}_t, \bar{e}_{y,t}$, the bounding functions are trivially given by $\mathcal{F}_x^1(\cdot), \mathcal{F}_x^2(\cdot)$ because linear functions are both quasiconvex and quasiconcave.

$$[\mathcal{F}^{1,\cap}(\bar{\mathbf{Y}}_t), \mathcal{F}^{2,\cap}(\bar{\mathbf{Y}}_t)] = [\mathcal{F}^{1,\cup}(\bar{\mathbf{Y}}_t), \mathcal{F}^{2,\cup}(\bar{\mathbf{Y}}_t)] = \bar{y}_t$$

For the bounding functions corresponding to $\bar{e}_{\psi,t}$, we use the system constraints to bound $(1 - \bar{y}_{2,t}C(\bar{y}_{1,t})) \in [\frac{31}{40}, \frac{49}{40}]$, to construct bounding functions as:

$$\mathcal{F}^{3,\cap}(\bar{\mathbf{Y}}_t) = \tan^{-1}\left(\min\{\frac{(\bar{y}_{2,t+1} - \bar{y}_{2,t})}{\frac{49}{40}(\bar{y}_{1,t+1} - \bar{y}_{1,t})}, \frac{(\bar{y}_{2,t+1} - \bar{y}_{2,t})}{\frac{31}{40}(\bar{y}_{1,t+1} - \bar{y}_{1,t})}\}\right)$$
$$\mathcal{F}^{3,\cup}(\bar{\mathbf{Y}}_t) = \tan^{-1}\left(\max\{\frac{(\bar{y}_{2,t+1} - \bar{y}_{2,t})}{\frac{49}{40}(\bar{y}_{1,t+1} - \bar{y}_{1,t})}, \frac{(\bar{y}_{2,t+1} - \bar{y}_{2,t})}{\frac{31}{40}(\bar{y}_{1,t+1} - \bar{y}_{1,t})}\}\right)$$

where $\mathcal{F}^{3,\cup}(\cdot), \mathcal{F}^{3,\cap}(\cdot)$ can be verified to be quasiconvex and quasiconcave respectively by using composition rules of quasilinear functions. Similarly for the inputs, we get:

$$\mathcal{F}^{4,\cap}(\bar{\mathbf{Y}}_t) = 0, \mathcal{F}^{4,\cup}(\bar{\mathbf{Y}}_t) = \frac{1}{dt}\left\| \begin{bmatrix} \frac{49}{40}(\bar{y}_{1,t+1} - \bar{y}_{1,t}) \\ \bar{y}_{2,t+1} - \bar{y}_{2,t} \end{bmatrix} \right\|_2$$
$$\mathcal{F}^{5,\cap}(\bar{\mathbf{Y}}_t) = -\frac{\pi}{2}, \mathcal{F}^{5,\cup}(\bar{\mathbf{Y}}_t) = \frac{\pi}{2}$$

Uncertainty Modeling

The disturbance d_t is assumed to lie in a state dependent set $D(x)$, modelled as $D(x) = \{d|\exists w : d = d(x) + w, ||w||_2 \leq \gamma\}$, where the function $d(x)$ is unknown, but assumed to be $L-$Lipschitz. The conditions on $D(x)$ for Assumption 2.1 are satisfied as shown in Example 2.1. The constants L, γ are estimated as follows:

1. Sample system transitions to obtain data-set $\mathcal{T} = \{(x_i, d_i)\}_{i=1}^{T}$
2. The true Lipschitz constant L and bound γ are one of the minimizers of the following semi-infinite, multi-objective optimization problem:
$$\min \left\{ (\tilde{L}, \tilde{\gamma}) \in \mathbb{R}^2_+ \middle| \begin{array}{l} \|\bar{d} - \bar{d}'\|_2 \leq \left|\tilde{L} \left\|x - x'\right\|_2 + 2\tilde{\gamma}\right|, \\ \forall \bar{d} \in D(x), \bar{d}' \in D(x'), \\ \forall x, x' \in \mathbb{R}^n \end{array} \right\}$$
3. We solve for L, γ for the scalarized objective $\tilde{L} + \tilde{\gamma}$, and approximate the semi-infinite optimization via the scenario approach using the data-set $\mathcal{T}$ to obtain the LP:
$$\begin{aligned} \min_{\tilde{L}>0, \tilde{\gamma}>0} \ & \tilde{L} + \tilde{\gamma} \\ \text{s.t } & \|d_i - d_j\|_2 \leq \tilde{L} \left\|x_i - x_j\right\|_2 + 2\tilde{\gamma}, \\ & \forall (x_i, d_i), (x_j, d_j) \in \mathcal{T} \end{aligned}$$
The approximated constants obtained after solving the linear program were $\hat{L} = 0.0329$, $\hat{\gamma} = 0.1640$. Using [5, Corollary 6] and linearity of the semi-infinite constraints, it can be shown that sampled constraints provide an *inner approximation* of the actual feasible set for the semi-infinite problem with high-confidence. Thus, the event $E := \{\hat{L} \geq L, \hat{\gamma} \geq \gamma\}$ holds with high probability, and the statements of Theorems 2.1–4.3 hold conditioned on E (which is conventional as noted in Remark 2.1).

Error Invariant and Error Policy

To construct the error invariant and the error policy for Assumption 4.5, we linearise the bicycle dynamics about $x_{ref} = (s, e_y, e_\psi) = (\sqrt{200}, 0, 0)$ $u_{ref} = (v, \delta) = (10, 0)$, and obtain bounds on the higher-order terms using the system constraints to give the linearized dynamics $x_{t+1} = f(x_{ref}, u_{ref}) + A(x_t - x_{ref}) + B(u_t - u_{ref}) + n_t + d_t$, where $n_t \in \mathcal{W}^l$ is the linearisation error and $d_t \in \hat{\mathcal{D}}$ corresponds to the error due to unmodelled dynamics. Similarly, the nominal dynamics are given as $\bar{x}_{t+1} = f(x_{ref}, u_{ref}) + A(\bar{x}_t - x_{ref}) + B(\bar{u}_t - u_{ref}) + \bar{n}_t$, and so, the error dynamics are $e_{t+1} = Ae_t + B\kappa(e_t) + d_t + [I_n \ -I_n][n_t^\top \ \bar{n}_t^\top]^\top$. For the combined disturbance $\bar{\mathcal{D}} = \hat{\mathcal{D}} \oplus [I_n \ -I_n](\mathcal{W}^l \times \mathcal{W}^l)$, the RPI is computed by fixing $\kappa(e_t) = K_{LQR}e_t$ and setting $\mathcal{E}_{RPI} = \bigoplus_{i=0}^{\infty}(A + BK_{LQR})^i\bar{\mathcal{D}}$. The cost matrices Q, R for the LQR policy are tuned such that $\mathcal{E}_{RPI} \subset \mathcal{X}$ and $K_{LQR}\mathcal{E}_{RPI} \subset \mathcal{U}$.

Results

We implement the proposed Robust Output-lifted LMPC strategy as described in Algorithm 2 for optimal control of the kinematic bicycle from Section 4.7. The results emphasize the following aspects of our approach:

Iterative Learning

The trajectory data across iterations $i = 0, .., j-1$ is used for constructing the disturbance support $\hat{\mathcal{D}}^j$, the terminal set $\bar{\mathcal{X}}_N^j$ and trajectory cost estimate via the terminal cost $P^j(\cdot)$. In the following plots, we observe that with each successive iteration, the disturbance supports shrink, the terminal sets enlarge and the trajectory costs decrease.

- **Disturbance Bound:** For Step 1 of Algorithm 2 at iteration j, we use the approach in Theorem 2.1 with system trajectory data $\{\{x_t^i, u_t^i\}_{t\geq 0}\}_{i=0}^{j-1}$ to obtain the outer-approximation of the disturbance support $\hat{\mathcal{D}}^j$ as a product of intervals, $\hat{\mathcal{D}}_1^j \times \hat{\mathcal{D}}_2^j \times \hat{\mathcal{D}}_3^j \subset \mathbb{R}^3$. We show the constructed disturbance support for iterations $j = 1, 10, 15, 20, 25$ in Figure 4.4. Notice that the disturbance supports shrink as more trajectory data as collected, as enforced by the SDP (2.11).

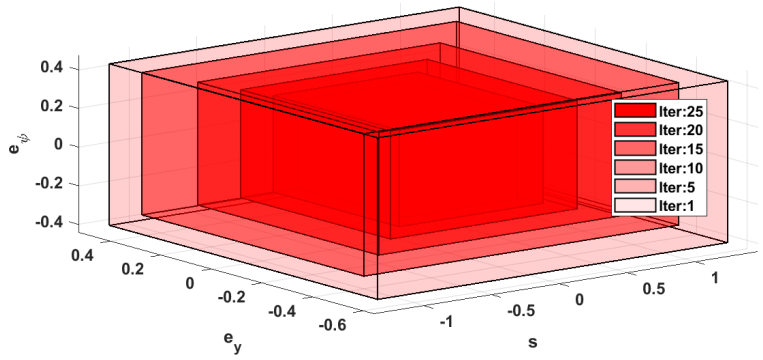
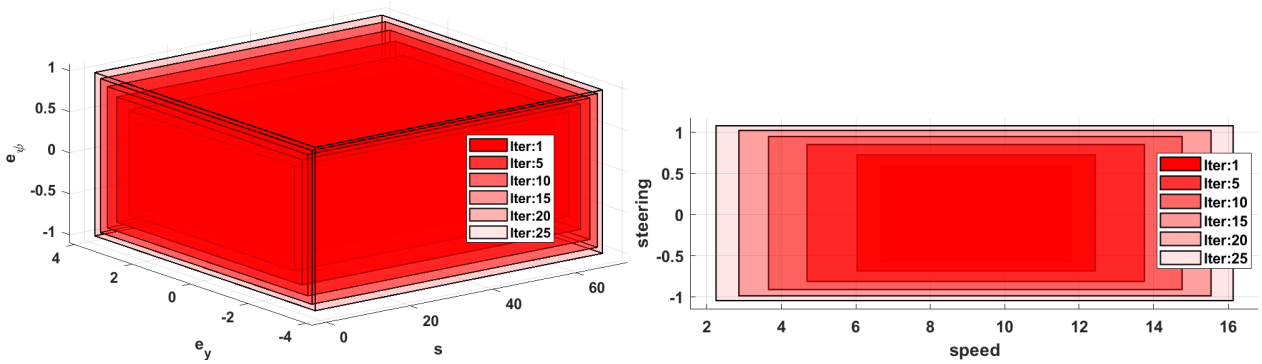


Figure 4.4: Disturbance support estimates across iterations. As more data is collected, the support estimates shrink.

- **Tightened Constraints:** For Step 2 of Algorithm 2 at iteration j, the tightened constraints $\bar{\mathcal{X}}^j, \bar{\mathcal{U}}^j$ are obtained by solving NLP (4.29) and using (4.3), (4.31) in Section 4.4. This requires the error invariant $\mathcal{E}^j$ and error policy $\kappa(e_t) = Ke_t$ for the error dynamics (4.8) with disturbance support $\hat{\mathcal{D}}^j$ computed in Step 1. The error invariant and error policy are computed as described in Section 4.7. We show the tightened constraints for iterations $j = 1, 10, 15, 20, 25$ in Figure 4.5. Notice that the tightened state and input constraints increase in size with increasing iterations.



(a) Tightened state constraints across iterations. (b) Tightened input constraints across iterations.

Figure 4.5: The tightened constraints increase in size across iterations, as the model uncertainty is learned.

- **Terminal Set:** The terminal set at iteration j is constructed from $\bar{\mathcal{CS}}^j_{\mathbf{y}}$ as $\bar{\mathcal{X}}^j_N = \{\bar{x} | \exists \bar{\mathbf{y}} \in \bar{\mathcal{CS}}^j_{\mathbf{y}}, \mathcal{F}_x(\bar{\mathbf{y}}) = \bar{x}\}$. To visualize this set, 1) we sample points in $\bar{\mathcal{CS}}^j_{\mathbf{y}}$, 2) map them onto the state space via $\mathcal{F}_x(\cdot)$ and 3) use Matlab's *alphaShape* function to fit a surface over the projected points (note that the image of convex set $\bar{\mathcal{CS}}^j_{\mathbf{y}}$ under continuous, surjective function $\mathcal{F}_x(\cdot)$ can be shown to be path-connected [93, Chapter 9]). The resulting terminal set approximation is shown in Figure 4.6 using trajectory data up to iteration $j = 25$, in both Frenet and global coordinates. Note that these continuous sets were constructed without any local linear approximations, or a pre-computed reference, and utilise the complete nonlinear dynamics of (4.1) *implicitly* via trajectory data $\{\bar{x}_t, \bar{u}_t\}_{t \geq 0}$ and the map $\mathcal{F}_x(\cdot)$.
- **Trajectory Costs:** We plot the trajectory costs of the closed-loop trajectories across all the iterations in Figure 4.7, and see that the trajectory costs decrease with each iteration, validating the claim of Theorem 4.3.

Robust Constraint Satisfaction

The tightened constraints within the Robust MPC formulation ensure that the closed-loop system trajectories satisfy the constraints robustly, despite the uncertainty in the dynamics.

We plot the closed-loop state and input trajectories of the kinematic bicycle in global coordinates in Figures 4.8, 4.9a, 4.9c across the iterations. Iteration 0 corresponds to the first trajectory with which our algorithm was initialized. At iteration 25, we see that the path of the closed-loop trajectory is significantly tighter than that of iteration 0 in Figure 4.8. From the Figures 4.9a, 4.9b, 4.9c, we see that the actual and nominal speed profiles, and the steering commands are within constraints. The tightened constraints

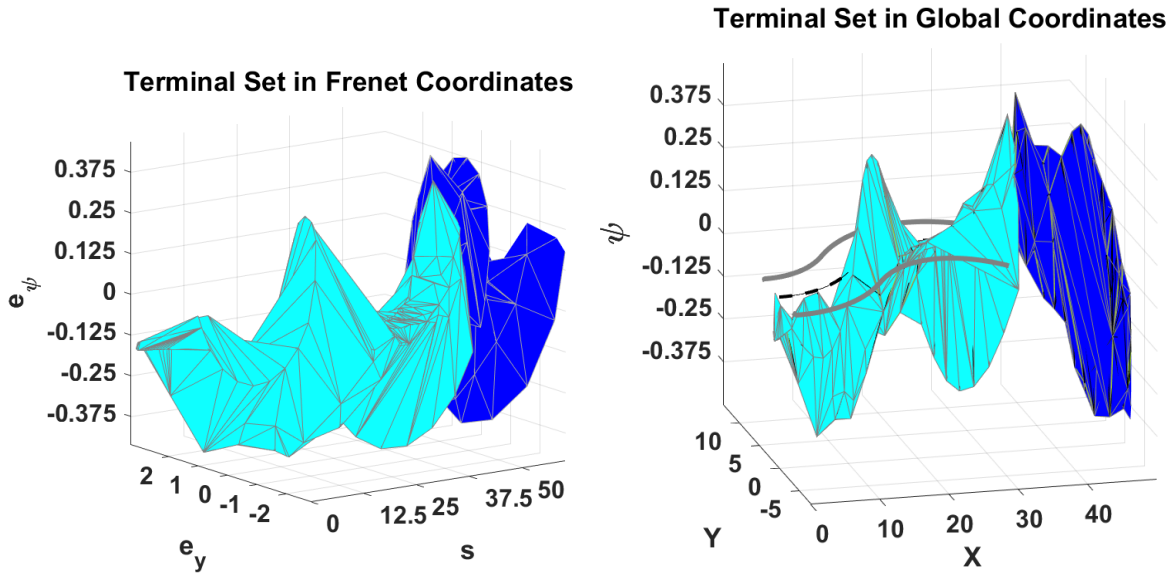


Figure 4.6: Terminal sets in state space, in Frenet and global coordinates constructed from nominal system trajectory data up to iteration $j = 20$. The dark blue regions denote states in $\mathcal{X}_G$.

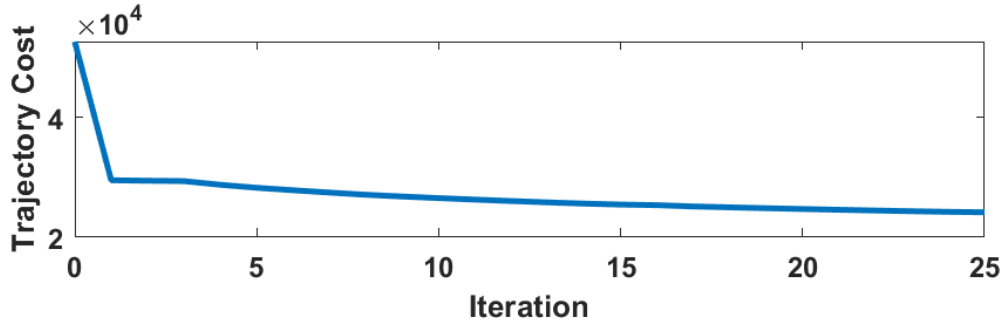


Figure 4.7: Closed-loop trajectory costs decrease across iterations.

for the nominal speed 4.9b are shown for $j = 25$. Also notice in Figure 4.9b that the trajectory in iteration 25 reaches $\mathcal{X}_G$ the fastest.

Computational Tractability

The proposed Convex Output Safe set $\bar{\mathcal{CS}}^j_{\mathbf{y}}$ is a convex set, as opposed to the discrete Safe set construction $\mathcal{SS}^j$ in [114]. This improves computation efficiency for solving the optimization problem 4.34 without sacrificing performance guarantees.

We compare the average solve times for our approach and the LMPC from [114] to

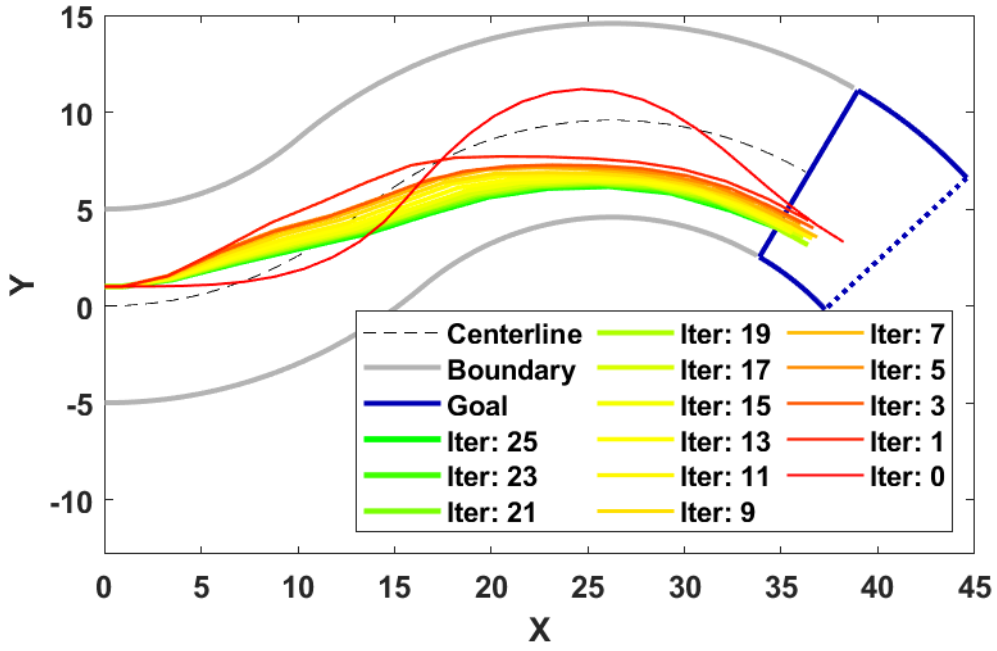
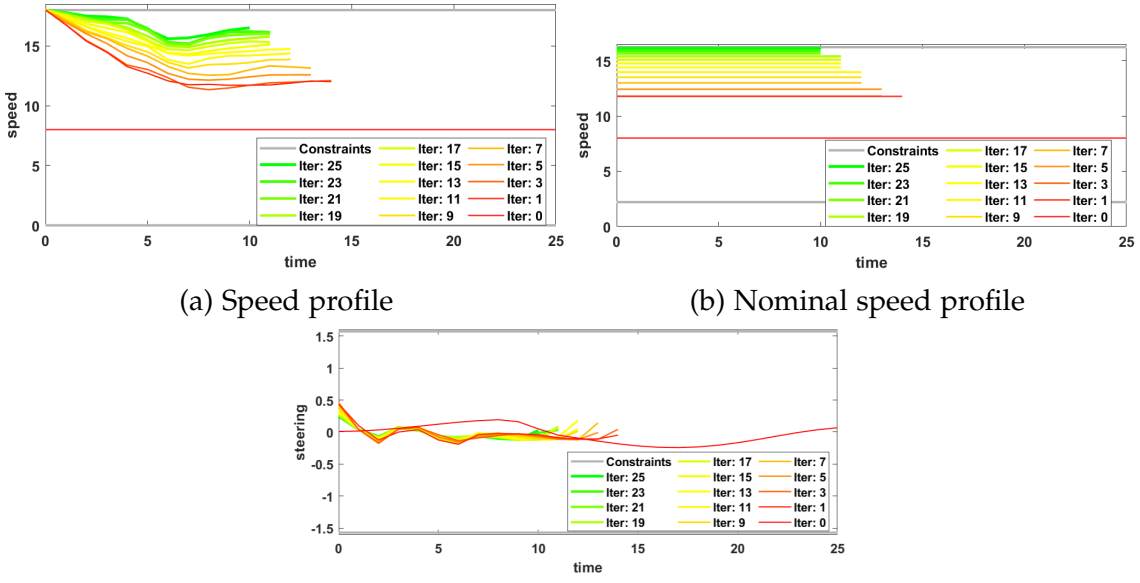


Figure 4.8: State trajectories across iterations.



(a) Speed profile

(b) Nominal speed profile

(c) Steering profile

Figure 4.9: Input trajectories across iterations.

demonstrate the benefit of using the continuous safe set $\bar{\mathcal{CS}}^j_{\mathrm{y}}$ over the discrete safe set

$\mathcal{SS}^j$. The former leads to solving a nonlinear program (which is solved using IPOPT) and the latter requires solving a mixed-integer nonlinear program (which is solved using BONMIN). In Figure 4.10, we see that the solve times increase with iterations because of the growing size of the safe sets, but our approach is markedly more efficient (with solve times $\leq 10^{-0.5}s \approx 0.3s$).

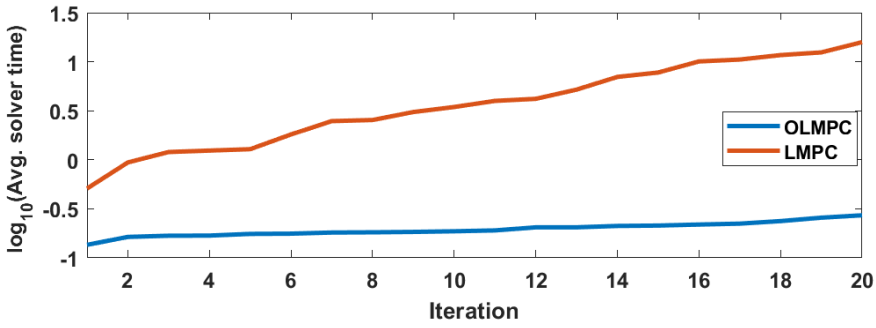


Figure 4.10: Avg. solve times for Output-lifted LMPC and LMPC for iterations $j = 1$ to $j = 20$ in $\log_{10}$ scale.

4.8 Conclusion

We proposed a formulation of Robust LMPC for systems with lifted outputs performing iterative tasks. We showed that using Assumption 4.2 on the lifted otputs, we can iteratively construct continuous, convex control invariant terminal sets and CLF terminal costs for nonlinear systems using historical lifted output data. Furthermore, our approach uses trajectory data to quantify and iteratively decrease model uncertainty, and construct tightened state and input constraints for the Robust MPC design. The proposed Robust Output-lifted LMPC scheme is recursively feasible, convergent, and iteratively improves system performance while guaranteeing robust constraint satisfaction.

4.9 Appendix

Proof of Lemma 4.1

Before proving the statement, we first prove the following auxiliary property that is granted by Assumption 4.2((A)):

$$[\mathcal{F}^{\cap}(\bar{\mathbf{Y}}), \mathcal{F}^{\cup}(\bar{\mathbf{Y}})] \subseteq [\min_{k=1,..,p} \mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k), \max_{k=1,..,p} \mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k)]$$

for $\bar{\mathbf{Y}} \in \text{cvx}(\{\bar{\mathbf{Y}}^1, \ldots, \bar{\mathbf{Y}}^p\})$, where the intervals and min, max are defined elment-wise. We proceed using induction on p, the number of points in the set. For $p = 2$, the property follows trivially by Assumption 4.2((A)). Suppose the property is true for $p-1$, i.e.,

$$\begin{aligned}&[\mathcal{F}^{\cap}(\bar{\mathbf{Y}}'), \mathcal{F}^{\cup}(\bar{\mathbf{Y}}')] \subseteq \\ &[\min_{k=1,..,p-1} \mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k), \max_{k=1,..,p-1} \mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k)]\end{aligned}$$

for any $\bar{\mathbf{Y}}' \in \text{cvx}(\{\bar{\mathbf{Y}}^1, \ldots, \bar{\mathbf{Y}}^{p-1}\})$. Adding an additional point $\bar{\mathbf{Y}}^p$ in the set, let $\bar{\mathbf{Y}} = \lambda \bar{\mathbf{Y}}^p + (1-\lambda)\bar{\mathbf{Y}}'$ for some $\lambda \in [0,1]$. Using the property for $p = 2$, we have

$$\begin{aligned}&[\mathcal{F}^{\cap}(\bar{\mathbf{Y}}), \mathcal{F}^{\cup}(\bar{\mathbf{Y}})] \subseteq \\ &[\min(\mathcal{F}^{\cap}(\bar{\mathbf{Y}}^p), \mathcal{F}^{\cap}(\bar{\mathbf{Y}}')), \max(\mathcal{F}^{\cup}(\bar{\mathbf{Y}}^p), \mathcal{F}^{\cup}(\bar{\mathbf{Y}}'))]\end{aligned}$$

Using the truth of property for $p-1$, we therefore write

$$\begin{aligned}&\min_{k=1,..,p} \mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k) \leq \min(\mathcal{F}^{\cap}(\bar{\mathbf{Y}}^p), \mathcal{F}^{\cap}(\bar{\mathbf{Y}}')), \\ &\max(\mathcal{F}^{\cup}(\bar{\mathbf{Y}}^p), \mathcal{F}^{\cup}(\bar{\mathbf{Y}}')) \leq \max_{k=1,..,p} \mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k) \\ \Rightarrow &[\mathcal{F}^{\cap}(\bar{\mathbf{Y}}), \mathcal{F}^{\cup}(\bar{\mathbf{Y}})] \subseteq [\min_{k=1,\ldots,p} \mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k), \max_{k=1,..,p} \mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k)] \ (\star)\end{aligned}$$

where the min, max for vectors are computed element-wise. The property thus holds true for p as well and induction helps us conclude that this holds for any $p \geq 1$.

Now we prove constraint satisfaction. We have $\forall k = 1, .., p$, $\mathcal{F}(\bar{\mathbf{Y}}^k) = (\bar{x}^k, \bar{u}^k) \in \bar{\mathcal{X}}^j \times \bar{\mathcal{U}}^j$. Since $\bar{\mathcal{X}}^j \times \bar{\mathcal{U}}^j \subset \bar{\mathcal{S}}_x \times \bar{\mathcal{S}}_u$ by construction, this implies that $\mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k)$, $\mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k) \in \hat{\mathcal{X}} \times \hat{\mathcal{U}}$ $\forall k = 1, \ldots, p$[1]. Additionally $\hat{\mathcal{X}} \times \hat{\mathcal{U}}$ is a box constraint, so we have

$$[\min_{k=1,\ldots,p} \mathcal{F}^{\cap}(\bar{\mathbf{Y}}^k), \max_{k=1,\ldots,p} \mathcal{F}^{\cup}(\bar{\mathbf{Y}}^k)] \subseteq \hat{\mathcal{X}}^j \times \hat{\mathcal{U}}^j.$$

Finally by using result $(\star)$ and $\mathcal{F}^{\cap}(\bar{\mathbf{Y}}) \leq \mathcal{F}(\bar{\mathbf{Y}}) \leq \mathcal{F}^{\cup}(\bar{\mathbf{Y}})$, we have $\mathcal{F}(\bar{\mathbf{Y}}) \in \bar{\mathcal{X}}^j \times \bar{\mathcal{U}}^j$ for any $\bar{\mathbf{Y}} \in \text{cvx}(\{\bar{\mathbf{Y}}^1, \ldots, \bar{\mathbf{Y}}^p\})$. ■

Proof of Proposition 4.1

By the definition of $\bar{\mathcal{CS}}_{\mathbf{y}}$, we have for $\bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}_{\mathbf{y}}$ that

$$\begin{aligned}&\begin{bmatrix}\bar{\mathbf{y}}_t \\ 1\end{bmatrix} = \sum_{i=1}^{|\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|} \lambda_i \begin{bmatrix}\bar{\mathbf{y}}_{t_i} \\ 1\end{bmatrix}, \\ &\bar{\mathbf{y}}_{t_i} \in \text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}}),\ \lambda_i \in [0,1],\ \forall i = 1, \ldots, |\text{vert}(\bar{\mathcal{CS}}_{\mathbf{y}})|\end{aligned} \tag{4.37}$$

[1]Technically, $\exists \bar{\mathbf{Y}}'^k : \mathcal{F}(\bar{\mathbf{Y}}'^k) = (\bar{x}^k, \bar{u}^k), \mathcal{F}^{\cup}(\bar{\mathbf{Y}}'^k), \mathcal{F}^{\cap}(\bar{\mathbf{Y}}'^k) \in \hat{\mathcal{X}}^j \times \hat{\mathcal{U}}^j$ but due to the uniqueness of $(\bar{x}^k, \bar{u}^k)$ granted by Definition 4.1, we can use $\bar{\mathbf{Y}}^k$ instead of $\bar{\mathbf{Y}}'^k$ w.l.o.g.

where each $\bar{\mathbf{y}}_{t_i}$ maps to a feasible state, i.e., $\mathcal{F}_x(\bar{\mathbf{y}}_{t_i}) = \bar{x}_{t_i} \in \bar{\mathcal{S}}_x \subseteq \mathcal{X}$. Invoking Lemma 4.2 gives us,

$$\mathcal{F}_x(\bar{\mathbf{y}}_t) = \bar{x}_t \in \bar{\mathcal{S}}_x \subset \mathcal{X} \tag{4.38}$$

We use the lifted-output $\bar{\mathbf{Y}}_{t_i}$ and map $\mathcal{F}_u(\cdot)$ to reconstruct nominal inputs

$$\bar{u}_{t_i} = \mathcal{F}_u([\bar{y}_{t_i}, \bar{y}_{t_i+1}, \dots, \bar{y}_{t_i+R}]) = \mathcal{F}_u([\bar{\mathbf{y}}_{t_i}, \bar{y}_{t_i+R}])$$

and note that $\bar{u}_{t_i} \in \bar{\mathcal{S}}_u \subset \mathcal{U}$. Consider the following control input

$$\begin{aligned} \bar{u}_t &= \mathcal{F}_u\Big(\sum_{i=1}^{|\text{vert}(\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}})|} \lambda_i [\bar{\mathbf{y}}_{t_i}, \bar{y}_{t_i+R}]\Big) \\ &= \mathcal{F}_u([\bar{\mathbf{y}}_t, \bar{y}_{t+R}]) \end{aligned} \tag{4.39}$$

where $\bar{y}_{t+R} = \sum_{i=1}^{|\text{vert}(\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}})|} \lambda_i \bar{y}_{t_i+R}$. Invoking Lemma 4.2 again proves $\bar{u}_t \in \mathcal{U}$.

By construction, notice that $\bar{\mathbf{y}}_{t_i} \in \bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}} \Rightarrow \bar{\mathbf{y}}_{t_i+1} \in \bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}}$. Thus, we can establish the invariance of $\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}}$ with respect to the time-shift dynamics $\delta(\cdot)$ as follows:

$$\begin{aligned} \bar{\mathbf{y}}_{t+1} &= \delta([\bar{\mathbf{y}}_t, \bar{y}_{t+R}]) \\ &= \delta\Big(\sum_{i=1}^{|\text{vert}(\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}})|} \lambda_i [\bar{\mathbf{y}}_{t_i}, \bar{y}_{t_i+R}]\Big) \\ &= \sum_{i=1}^{|\text{vert}(\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}})|} \lambda_i \bar{\mathbf{y}}_{t_i+1} \Rightarrow \bar{\mathbf{y}}_{t+1} \in \bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}}. \end{aligned} \tag{4.40}$$

Finally, we show that $\mathcal{F}_x(\bar{\mathbf{y}}_{t+1})$ is precisely $\bar{x}_{t+1} = f(\bar{x}_t, \bar{u}_t)$ which also lies within constraints. Let $\bar{u}_2, \dots, \bar{u}_{R-1} \in \mathbb{R}^m$ be the remaining inputs that generate $[\bar{y}_t, \bar{\mathbf{y}}_{t+1}] \in \mathbb{R}^{m \times R+1}$, i.e.,

$$\begin{aligned} [\bar{y}_t, \bar{\mathbf{y}}_{t+1}] =& [h(\bar{x}_t), h(f(\bar{x}_t, \bar{u}_t)), h(f^{(2)}(\bar{x}_t, \bar{u}_t, \bar{u}_2)), \dots, \\ & h(f^{(R-1)}(\bar{x}_t, \bar{u}_t, \dots, \bar{u}_{R-1}))] \in \mathbb{R}^{m \times R+1} \end{aligned} \tag{4.41}$$

where $f^{(k)}(\bar{x}_t, \bar{u}_t, \dots, \bar{u}_k) = \underbrace{f(\dots(f}_{k \text{ times}}(\bar{x}_t, \bar{u}_t), \dots \bar{u}_k)$. Using the map (4.5) to construct the nominal state, we can write

$$\begin{aligned} \mathcal{F}_x(\bar{\mathbf{y}}_{t+1}) =& \mathcal{F}_x([h(f(\bar{x}_t, \bar{u}_t)), h(f^{(2)}(\bar{x}_t, \bar{u}_t, \bar{u}_2)), \dots, \\ & h(f^{(R-1)}(\bar{x}_t, \bar{u}_t, \dots, \bar{u}_{R-1}))]) \\ =& f(\bar{x}_t, \bar{u}_t) = \bar{x}_{t+1} \end{aligned}$$

where the last equality is true because of the unique correspondence from $[\bar{y}_t, \dots, \bar{y}_{t+R-1}]$ $= [h(\bar{x}_t), \dots, h(f^{(R-1)}(\bar{x}_t, \bar{u}_t, \dots, \bar{u}_{t+R-1}))]$ to $\bar{x}_t$ (Definition 4.1). Invoking Lemma 4.2 again using sequences $\bar{\mathbf{y}}_{t_i+1}, \forall i = 1, \dots, |\text{vert}(\bar{\mathcal{C}\mathcal{S}}_{\mathbf{y}})|$ gives us

$$f(\bar{x}_t, \bar{u}_t) \in \mathcal{X}. \tag{4.42}$$

■

Proof of Proposition 4

1) First note that $\bar{\mathbf{y}} \in \bar{\mathcal{CS}}^j_{\mathbf{y}}$ implies that the optimization problem implicit in the definition (4.25) of $\bar{Q}^j(\cdot)$ is feasible. Also see that since the feasible set is compact (countable product of compact sets is compact by Tychonoff's theorem) and the objective is continuous (linear, in fact, and bounded because of Theorem), a minimizer exists by Weierstrass' theorem for every $\bar{\mathbf{y}} \in \bar{\mathcal{CS}}^j_{\mathbf{y}}$. Thus for any $\bar{\mathbf{y}} \in \bar{\mathcal{CS}}^j_{\mathbf{y}}$, we can write $\bar{Q}^j(\bar{\mathbf{y}}) = \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \mathcal{C}_k^i$ where the $\lambda_k^{\star i}$s satisfy the constraints in (4.25). The definition of $\mathcal{C}_k^i$ in (4.24) and positive definiteness of $c(\cdot)$ by (4.23) imply that $\bar{Q}^j(\mathbf{y}) \succ 0 \ \forall \mathbf{y} \in \mathcal{CS}^j_{\mathbf{y}} \backslash \mathcal{Y}_G$. For any $\bar{\mathbf{y}} \in \mathcal{Y}^j_G$, we have $\bar{\mathbf{y}} = \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^i \bar{\mathbf{y}}_k^i$ with $\lambda_k^i > 0$ only for $\bar{\mathbf{y}}_k^i \in \mathcal{Y}_G$, implying $\bar{Q}^j(\bar{\mathbf{y}}) \leq 0$. Thus, $\bar{Q}^j(\mathbf{y}) = 0 \ \forall \bar{\mathbf{y}} \in \mathcal{Y}^j_G$. We finish the proof for the first part by observing that for $\bar{\mathbf{y}} \in \mathcal{Y}_G \backslash \mathcal{Y}^j_G$, there exists no combination of multipliers such that $\lambda_k^i > 0$ only for $\bar{\mathbf{y}}_k^i \in \mathcal{Y}_G$, and since $\mathcal{C}_k^i > 0$ for $\bar{\mathbf{y}}_k^i \notin \mathcal{Y}_G$, we must have $\bar{Q}^j(\bar{\mathbf{y}}) > 0$.

2) For any $\bar{\mathbf{y}}_t \in \bar{\mathcal{CS}}^j_{\mathbf{y}}$, let $\bar{Q}^j(\bar{\mathbf{y}}_t) = \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \mathcal{C}_k^i$ with $\lambda_k^{\star i}$ satisfying the constraints in (4.25). Observing the linearity of the forward-time shift operator $\delta(\cdot,\cdot)$, we have

$$
\begin{aligned}
\bar{\mathbf{y}}_{t+1} &= \delta(\bar{\mathbf{y}}_t, \bar{y}_{t+R}) \\
&= \delta\Big(\sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \bar{\mathbf{y}}_k^i, \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \bar{y}_{k+R}^i\Big) \\
&= \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \delta(\bar{\mathbf{y}}_k^i, \bar{y}_{k+R}^i) \\
&= \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \bar{\mathbf{y}}_{k+1}^i .a
\end{aligned}
$$

Thus the same $\lambda_k^{\star i}$s are also feasible for (4.25) at $\mathbf{y}_{t+1}$ and we have

$$
\begin{aligned}
\bar{Q}^j(\bar{\mathbf{y}}_{t+1}) - \bar{Q}^j(\bar{\mathbf{y}}_t) &\leq \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} (\mathcal{C}_{k+1}^i - \mathcal{C}_k^i) \\
&= \sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} (-c(\bar{\mathbf{Y}}_k^i)) \\
&\leq -c\Big(\sum_{i=0}^{j-1} \sum_{k\geq 0} \lambda_k^{\star i} \bar{\mathbf{Y}}_k^i\Big) \\
&= -c(\bar{\mathbf{Y}}_t)
\end{aligned}
$$

The second to last inequality comes from the convexity of $c(\cdot)$. This completes the proof of the second part of the proposition. ■

Proof of Proposition 4.2

Since $\mathcal{E}$ is RPI for (4.28) with $\kappa(e_t) = Ke_t$ and disturbance support $\hat{\mathcal{D}}$, we have

$$\begin{aligned}&\forall \bar{x}_t \in \mathcal{X} \ominus \mathcal{E}, \forall \bar{u}_t \in \mathcal{U} \ominus K\mathcal{E} : \\ &e_t \in \mathcal{E} \Rightarrow e_{t+1} = f_e(e_t, \bar{x}_t, \bar{u}_t) + d_t \in \mathcal{E},\ \forall d_t \in \hat{\mathcal{D}}.\end{aligned}$$

Since $x_t = \bar{x}_t + e_t \in \mathcal{X}$ and $u_t = \bar{u}_t + Ke_t \in \mathcal{U}$ for any $e_t \in \mathcal{E}$, $\bar{x}_t \in \mathcal{X} \ominus \mathcal{E}$, $\bar{u}_t \in \mathcal{U} \ominus K\mathcal{E}$, we have $d_t \in D(e_t + \bar{x}_t, u_t) \subset \mathcal{D}$ from (4.13). The desired result then follows because $\mathcal{D} \subset \hat{\mathcal{D}}$ and the definition of the Pontryagin difference operator $\ominus$. ■

Proof of Theorem 4.1

For any iteration $j \geq 1$, suppose that the problem (4.34) is feasible at time $t \geq 1$. Let the state-input trajectory corresponding to the optimal solution of (4.34) be

$$\{\bar{x}^{j\star}_{t|t}, \bar{u}^{j\star}_{t|t}, \bar{x}^{j\star}_{t+1|t}, \bar{u}^{j\star}_{t+1|t}, \ldots, \bar{x}^{j\star}_{t+N|t}, \bar{\mathbf{y}}^{j\star}_{t+N|t}\}. \tag{4.43}$$

Applying the control $u^j_t = \bar{u}^{j\star}_{t|t} + K(x^j_t - \bar{x}^{j\star}_{t|t})$ to system (4.1) yields x^j_{t+1} such that $x^j_{t+1} - \bar{x}^{j\star}_{t+1|t} \in \mathcal{E}^j,\ \forall d_t \in D(x^j_t, u^j_t)$ because $x^j_t - \bar{x}^{j\star}_{t|t} \in \mathcal{E}^j$ ($\because$ Proposition 4.2), where $\bar{x}^{j\star}_{t+1|t} = f(\bar{x}^j_t, \bar{u}^{j\star}_{t|t})$ and $\bar{x}^j_t = \bar{x}^{j\star}_{t|t}$. We also have

$$\bar{x}^{j\star}_{t+N|t} = \mathcal{F}_x(\bar{\mathbf{y}}^{j\star}_{t+N|t}),\ \bar{\mathbf{y}}^{j\star}_{t+N|t} \in \bar{\mathcal{CS}}^j_{\mathbf{y}}.$$

From Proposition 4.1, we have $\bar{x}^{j\star}_{t+N|t} \in \bar{\mathcal{X}}^j$, and $\exists \bar{y}'$ such that $\bar{\mathbf{y}}' = \delta(\bar{\mathbf{y}}^{j\star}_{t+N|t}, \bar{y}') \in \bar{\mathcal{CS}}^j_{\mathbf{y}}$, $\bar{u}' = \mathcal{F}_u([\bar{\mathbf{y}}^{j\star}_{t+N|t}, \bar{y}']) \in \bar{\mathcal{U}}^j$ and $\bar{x}' = \mathcal{F}_x(\bar{\mathbf{y}}') = f(\bar{x}^{j\star}_{t+N|t}, \tilde{u}) \in \bar{\mathcal{X}}^j$. Now consider the following state-input trajectory

$$\{x^{j\star}_{t+1|t}, \bar{u}^{j\star}_{t+1|t}, \ldots, \bar{x}^{j\star}_{t+N|t}, \bar{u}', \bar{x}', \bar{\mathbf{y}}'\} \tag{4.44}$$

and see that this is feasible for problem (4.34) at time $t+1$.
We have shown that feasibility of the LMPC problem (4.34) at time $t \geq 1$ implies feasibility of the LMPC problem (4.34) at time $t+1$. For $t = 0$ and any $j \geq 1$, we have $x^j_t = x^{j-1}_t = x_S$ and $\bar{\mathcal{X}}^j \supset \bar{\mathcal{X}}^{j-1}$, $\bar{\mathcal{U}}^j \supset \bar{\mathcal{U}}^{j-1}$, $\bar{\mathcal{CS}}^j_{\mathbf{y}} \supset \bar{\mathcal{CS}}^{j-1}_{\mathbf{y}}$. Thus, the solution to (4.34) from iteration $j-1$ at $t = 0$ is feasible for iteration j, with the initialization for $j = 1$ given by Assumption 4.4. Induction on time t proves the persistent feasibility of (4.34) $\forall t \geq 0, \forall j \geq 1$.

Thus, $x^j_t - \bar{x}^j_t \in \mathcal{E}^j$, $\bar{x}^j_t \in \bar{\mathcal{X}}^j \subset \mathcal{X} \ominus \mathcal{E}^j$, $\bar{u}^j_t \in \bar{\mathcal{U}}^j \subset \mathcal{U} \ominus K\mathcal{E}^j\ \forall t \geq 0, \forall j \geq 1$. Since $(\mathcal{X} \ominus \mathcal{E}^j) \oplus \mathcal{E}^j \subset \mathcal{X}, (\mathcal{U} \ominus K\mathcal{E}^j) \oplus K\mathcal{E}^j \subset \mathcal{U}$, we have $x^j_t \in \mathcal{X}, u^j_t \in \mathcal{U}$. ■

Proof of Lemma 4.2

Define the set $\mathcal{Y}_G^x = \{\mathbf{y} | [\mathbf{y}, y] \in \mathcal{Y}_G\}$, which is just a linear projection of $\mathcal{Y}_G$ to obtain the first R outputs, and see that $\lim_{t\to\infty} \text{dist}_{\mathcal{Y}_G}(\bar{\mathbf{Y}}_t) = 0 \Rightarrow \lim_{t\to\infty} \text{dist}_{\mathcal{Y}_G^x}(\bar{\mathbf{y}}_t) = 0$ because of the continuity of the projection. Then from the fact that the image of the flat map (4.5) is unique (Definition 4.1) and the continuity of $\mathcal{F}_x(\cdot)$ (Assumption 4.2(A)), we have

$$\begin{aligned}\lim_{t\to\infty} \text{dist}_{\mathcal{X}_G \ominus \mathcal{E}^1}(\bar{x}_t) &= \lim_{t\to\infty} \text{dist}_{\mathcal{X}_G \ominus \mathcal{E}^1}(\mathcal{F}_x(\bar{\mathbf{y}}_t)) \\ &= \lim_{t\to\infty} \text{dist}_{\mathcal{F}_x^{-1}(\mathcal{X}_G \ominus \mathcal{E}^1)}(\bar{\mathbf{y}}_t).\end{aligned}$$

From the definition of $\mathcal{Y}_G, \mathcal{Y}_G^x$, we know that $\bar{\mathbf{y}} \in \mathcal{Y}_G^x \Rightarrow \mathcal{F}_x(\bar{\mathbf{y}}) = \bar{x} \in \mathcal{X}_G \ominus \text{Box}(\mathcal{E}^1) \subset \mathcal{X} \ominus \mathcal{E}^1$. Thus, $\mathcal{F}_x(\mathcal{Y}_G^x) \subseteq \mathcal{X}_G \ominus \mathcal{E}^1 \Rightarrow \mathcal{Y}_G^x \subseteq \mathcal{F}_x^{-1}(\mathcal{X}_G \ominus \mathcal{E}^1)$ and so $\lim_{t\to\infty} \text{dist}_{\mathcal{Y}_G^x}(\bar{\mathbf{y}}_t) = 0 \Rightarrow \lim_{t\to\infty} \text{dist}_{\mathcal{F}_x^{-1}(\mathcal{X}_G \ominus \mathcal{E}^1)}(\bar{\mathbf{y}}_t) = 0$.

$$\therefore \lim_{t\to\infty} \text{dist}_{\mathcal{X}_G \ominus \mathcal{E}^1}(\bar{x}_t) = 0$$

. ■

Proof of Theorem 4.2

We adopt the same notation as the proof of theorem 4.1. Using the feasibility of (4.44) for the problem (4.34) at time $t+1$ and the fact that $J_{t+1}^j(x_{t+1}^j)$ is the optimal cost at time $t+1$, we get

$$\begin{aligned}J_{t+1}^j(x_{t+1}^j) &\leq \bar{Q}^j(\mathbf{y}') + c(\bar{\mathbf{Y}}_{N|t}^\star) + \sum_{k=1}^{N-1} c(\bar{\mathbf{Y}}_{k|t}^\star) \\ &\leq \bar{Q}^j(\bar{\mathbf{y}}_{N|t}^\star) + \sum_{k=1}^{N-1} c(\bar{\mathbf{Y}}_{k|t}^\star) \\ &= J_t^j(x_t^j) - c(\bar{\mathbf{Y}}_{t|t}^\star). \end{aligned} \tag{4.45}$$

The feasibility of the problem (4.34) (guaranteed by Theorem 4.1) and positive definiteness of $c(\cdot)$ imply that the sequence $\{J_t^j(x_t^j)\}_{t\geq 0}$ is non-increasing. Moreover, positive definiteness of $\bar{Q}^j(\cdot)$ (by Proposition 4) further implies that the sequence is lower bounded by 0. Thus the sequence converges and taking limits on both sides of (4.45) gives

$$0 \leq \lim_{t\to\infty} -c(\bar{\mathbf{Y}}_t^j) \leq 0 \Rightarrow \lim_{t\to\infty} c(\bar{\mathbf{Y}}_t^j) = 0$$

By continuity of $c(\cdot)$, we have that $\lim_{t\to\infty} c(\bar{\mathbf{Y}}_t^j) = 0 \Leftrightarrow \lim_{t\to\infty} \text{dist}_{c^{-1}(0)}(\bar{\mathbf{Y}}_t^j) = 0$. From (4.23), we know $c^{-1}(0) = \mathcal{Y}_G$ and thus, $\lim_{t\to\infty} \text{dist}_{\mathcal{Y}_G}(\bar{\mathbf{Y}}_t^j) = 0$ and consequently

$\lim_{t\to\infty} \text{dist}_{\mathcal{X}_G \ominus \mathcal{E}^1}(\bar{x}_t^j) = 0$ by lemma 4.2. By Proposition 4.2, we also have $x_t^j - \bar{x}_t^j \in \mathcal{E}^j \subseteq \mathcal{E}^1$. Using the facts $(\mathcal{X}_G \ominus \mathcal{E}^1) \oplus \mathcal{E}^1 \subseteq \mathcal{X}_G$ and $\text{dist}_S(\cdot) \geq 0$, we have

$$\begin{aligned}
\lim_{t\to\infty} \text{dist}_{\mathcal{X}_G}(x_t^j) &= \lim_{t\to\infty} \text{dist}_{\mathcal{X}_G}(\bar{x}_t^j + x_t^j - \bar{x}_t^j) \\
&\leq \lim_{t\to\infty} \text{dist}_{(\mathcal{X}_G \ominus \mathcal{E}^1)\oplus\mathcal{E}^1}(\bar{x}_t^j + x_t^j - \bar{x}_t^j) \\
&= 0 \\
\Rightarrow \lim_{t\to\infty} \text{dist}_{\mathcal{X}_G}(x_t^j) &= 0
\end{aligned}$$

■

Proof of Theorem 4.3

The cost of the trajectory in iteration $j-1$ is given by

$$\begin{aligned}
J_{0\to\infty}^{j-1}(x_S) &= \sum_{t\geq 0} c(\bar{\mathbf{Y}}_t^{j-1}) \\
&= \sum_{t=0}^{N-1} c(\bar{\mathbf{Y}}_t^{j-1}) + \mathcal{C}_N^{j-1} \\
&\geq \sum_{t=0}^{N-1} c(\bar{\mathbf{Y}}_t^{j-1}) + \bar{Q}^j(\bar{\mathbf{y}}_N^{j-1}) \\
&\geq J_0^j(x_S)
\end{aligned}$$

The second to last inequality comes from the definition of $\bar{Q}^j(\cdot)$ in (4.25) while the last inequality comes from optimality of problem (4.34) in the jth iteration starting from $x_0^j = x_S$.
Now we use inequality (4.45) repeatedly to derive

$$\begin{aligned}
J_0^j(x_S) &\geq c(\bar{\mathbf{Y}}_0^j) + J_1^j(x_1^j) \\
&\geq c(\bar{\mathbf{Y}}_0^j) + c(\bar{\mathbf{Y}}_1^j) + J_2^j(x_2^j) \\
&\geq \lim_{t\to\infty} (\sum_{k=0}^{t-1} c(\bar{\mathbf{Y}}_t^j) + J_t^j(x_t^j)) \\
&\geq \lim_{t\to\infty} \sum_{k=0}^{t-1} c(\bar{\mathbf{Y}}_t^j) \\
&= J_{0\to\infty}^j(x_S)
\end{aligned}$$

Thus,

$$J_{0\to\infty}^{j-1}(x_S) \geq J_{0\to N}^j(x_S) \geq J_{0\to\infty}^j(x_S)$$

The desired statement easily follows from above. ■

Chapter 5

Optimization-based Collision Avoidance in Dynamic, Uncertain Environments

5.1 Overview

In this chapter, we develop an MPC algorithm for motion planning in the presence of uncertain predictions of the surrounding agents/obstacles represented by general convex sets. Our main focus is to design a planner that can solve the planning problem 1) efficiently (measured as time to compute the solution, critical for real-time deployment), 2) reliably (measured as low rate of infeasibility, critical for fewer interventions of backup planners) and 3) compute high-quality solutions (measured in terms of mobility, comfort and conservatism).

Contributions

- We build on the dual perspective collision avoidance from [144] to find separating hyperplanes between convex sets in the presence of three different uncertainty descriptions: 1) Arbitrary distributions with polytopic support, 2) Gaussian distributions, and 3) Arbitrary distributions with first two moments known.
- We propose a Sequential Convex Programming (SCP) approach for MPC formulation, that jointly optimizes over the separating hyperplanes for collision avoidance and parameterized feedback policies over the AV's and obstacles' states.
- We present a systematic evaluation of our framework along axes of (i) mobility, (ii) comfort, (iii) conservatism and (iv) computational efficiency at a simulated traffic intersection.

5.2 Problem Formulation

Dynamics and Geometry of AV and Obstacles

We model the AV as a linear time-varying discrete-time model

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + E_t w_t \\ p_t &= C x_t + c_t \end{aligned} \tag{5.1}$$

where $x_t \in \mathbb{R}^{n_x}, u_t \in \mathbb{R}^{n_u}, w_t \in \mathbb{R}^{n_x}$ are the state, input and process noise respectively and A_t, B_t, E_t, C, c_t are the system matrices at time t. The vector $p_t \in \mathbb{R}^n$ describes the position of the autonomous agent in a global, Cartesian coordinate system. Given the rotation matrix R_t describing the orientation of the AV (with respect to the global coordinate system) at time t, define the space occupied by the autonomous agent as the set

$$\mathbb{S}_t(x_t) = \{z \in \mathbb{R}^n \mid z = R_t y + p_t, Gy \preceq_{\mathcal{K}} g\} \tag{5.2}$$

where $G \in \mathbb{R}^{l \times n}, g \in \mathbb{R}^l$ and $\mathcal{K} \subset \mathbb{R}^l$ is a closed convex cone with non-empty interior. The set $\{y | Gy \preceq_{\mathcal{K}} g\}$ is non-empty, convex and compact, and describes the space occupied by the un-oriented agent at the origin, and can denote various shapes for an appropriate choice of $\mathcal{K}$ (e.g., a polytope when $\mathcal{K}$ is the positive orthant or an ellipsoid when $\mathcal{K}$ is the second-order cone).

Now suppose that there are M obstacles, each described by the affine time-varying discrete-time dynamics

$$\begin{aligned} o^i_{t+1} &= T^i_t o^i_t + q^i_t + F^i_t n^i_t, \\ p^i_t &= C o^i_t \qquad\qquad\qquad \forall i \in \mathbb{I}_1^M \end{aligned} \tag{5.3}$$

where $o^i_t \in \mathbb{R}^{n_x}, p^i_t \in \mathbb{R}^n, n^i_t \in \mathbb{R}^{n_x}$ are the state, position and process noise respectively and T^i_t, q^i_t, F^i_t, C are the system matrices of the i^{th} obstacle at time t. Now suppose that the orientation R^i_t of the i^{th} obstacle at time t is given, and define the space occupied by the obstacle as the set

$$\mathbb{S}^i_t(o^i_t) = \{z \in \mathbb{R}^n \mid z = R^i_t y + p^i_t, G^i y \preceq_{\mathcal{K}} g^i\} \tag{5.4}$$

where the non-empty, convex and compact set $\{y | G^i y \preceq_{\mathcal{K}} g^i\}$ describes the un-oriented shape of the obstacle at the origin. We also introduce the notation $o_t = [o^{1\top}_t \ldots, o^{M\top}_t]^\top$, $n_t = [n^{1\top}_t \ldots, n^{M\top}_t]^\top$ to denote the stacked obstacle state and process noise vectors at time t, and $T_t = \text{blkdiag}(T^1_t, \ldots, T^M_t), F_t = \text{blkdiag}(F^1_t, \ldots, F^M_t), q_t = [q^{1\top}_t \ldots q^{M\top}_t]^\top$ to define the combined dynamics of the obstacles as $o_{t+1} = T_t o_t + q_t + F_t n_t$.

Uncertainty Description

The presence of process noises w_t and n_t in the dynamics of the controlled agent and the obstacles adds uncertainty in the prediction of their state trajectories. In this paper, we consider three different descriptions of the distributions of the process noise as follows.

- **D1**: The process noise $[w_t^\top, n_t^\top]^\top$ are i.i.d $\forall t \geq 0$ and have compact support,

$$[w_t^\top \; n_t^\top]^\top \in \mathcal{D} = \{d \mid ||\Gamma d||_\infty \leq \gamma\}$$

  for $\gamma > 0$ and non-singular Γ.

- **D2**: The process noise $[w_t^\top, n_t^\top]^\top$ are i.i.d $\forall t \geq 0$ and are given by the Gaussian distribution

$$[w_t^\top \; n_t^\top]^\top \sim \mathcal{N}(0, \Sigma).$$

- **D3**: The process noise $[w_t^\top, n_t^\top]^\top$ are i.i.d $\forall t \geq 0$ and are given by an unknown distribution with known mean and covariance,

$$\mathbb{E}\left([w_t^\top \; n_t^\top]^\top\right) = 0,$$
$$\mathbb{E}\left(([w_t^\top \; n_t^\top]^\top)([w_t^\top \; n_t^\top])\right) = \Sigma.$$

Model Predictive Control Formulation

We aim to compute the MPC policy $u_t^{MPC}(x_t, o_t)$ for the AV such that it avoids collisions with the obstacles, i.e., $\mathbb{S}_t(x_t) \cap \mathbb{S}_t^i(o_t^i) = \emptyset \; \forall i \in \mathbb{I}_1^M$, while respecting polytopic state-input constraints given by $\mathcal{XU} = \{(x, u) \mid F_j^x x + F_j^u u \leq f_j, \; \forall j \in \mathbb{I}_1^J\}$, where $F_j^x \in \mathbb{R}^{1 \times n_x}, F_j^u \in \mathbb{R}^{1 \times n_u}, f_j \in \mathbb{R} \; \forall j \in \mathbb{I}_1^J$. To compute the MPC, we solve the following finite-horizon constrained optimal control problem.

$$\begin{aligned}
&\mathbf{OPT}_t(\mathbf{D} \in \{\mathbf{D1}, \mathbf{D2}, \mathbf{D3}\}): \\
&\min_{\boldsymbol{\theta}_t} \quad J_t(\mathbf{x}_t, \mathbf{u}_t) && \text{(5.5a)} \\
&\text{s.t.} \quad x_{k+1|t} = A_k x_{k|t} + B_k u_{k|t} + E_k w_{k|t} && \text{(5.5b)} \\
&\qquad o_{k+1|t} = T_k o_{k|t} + q_k + F_k n_{k|t} && \text{(5.5c)} \\
&\qquad (\mathbf{w}_t, \mathbf{n}_t) \text{ given by } \mathbf{D} && \text{(5.5d)} \\
&\qquad (\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t) \in \mathcal{C}(\mathbf{D}) && \text{(5.5e)} \\
&\qquad \mathbf{u}_t = \Pi_{\boldsymbol{\theta}_t}(\mathbf{x}_t, \mathbf{o}_t) && \text{(5.5f)} \\
&\qquad x_{t|t} = x_t, \; o_{t|t} = o_t && \text{(5.5g)} \\
&\qquad \forall k \in \mathbb{I}_t^{t+N-1}
\end{aligned}$$

where $\mathbf{x}_t = [x_{t|t}^\top, \ldots, x_{t+N|t}^\top]^\top$ ($\mathbf{o}_t$ similarly) and $\mathbf{u}_t = [u_{t|t}^\top, \ldots, u_{t+N-1|t}^\top]^\top$ ($\mathbf{w}_t, \mathbf{n}_t$ similarly). The objective (5.5a) penalizes deviation of the agent's trajectory from a desired reference. The obstacle avoidance constraints, and state-input constraints along the prediction horizon are summarised as $\mathcal{C}(\mathbf{D})$ in (5.5e), and depend on the uncertainty description assumed in (5.5d). In (5.5f), the control inputs $\mathbf{u}_t$ along the prediction horizon are given by a parameterized policy class that depends on predictions of the agent's and

obstacles' trajectories. We solve our MPC problem in batch form by explicitly substituting for the equality constraints (5.5b), (5.5c) and optimize over the policy parameters $\boldsymbol{\theta}_t$.

5.3 Collision Avoidance for Dynamic Obstacles with Uncertain Predictions

In this section, we detail our MPC formulation for avoiding collisions with uncertain, dynamically moving obstacles. First, we present our choice of policy parameterization in (5.5f). Second, we present a continuous reformulation of the set intersection problem $\mathbb{S}_k(x_{k|t}) \cap \mathbb{S}_k^i(o_{k|t}^i) = \emptyset \ \forall i \in \mathbb{I}_1^M$, which we use as our model for collision avoidance. Then we derive deterministic reformulations of the collision avoidance constraints and state-input constraints for each uncertainty description, and finally consolidate all these components to present our MPC design.

Policy Parameterization

We use parameterised feedback policies $\Pi_{\boldsymbol{\theta}_t}(\mathbf{x}_t, \mathbf{o}_t)$ for the control actions $\mathbf{u}_t$ (as in (5.5f)) along the prediction horizon. Consider the following policy for the input at time k,

$$u_{k|t} = h_{k|t} + \sum_{l=t}^{k-1} M_{l,k|t} w_{l|t} + K_{k|t}(o_{k|t} - \bar{o}_{k|t}) \tag{5.6}$$

which uses state feedback for the obstacles' states but affine disturbance feedback for feedback over the agent's states (cf. [54] for equivalence of state feedback and disturbance feedback, [12] for a recent application). The nominal states $\bar{o}_{k|t}$ of the obstacles are obtained as $\bar{o}_{k+1|t} = T_k \bar{o}_{k|t} + q_k \ \forall k \in \mathbb{I}_t^{t+N-1}$, with $\bar{o}_{t|t} = o_t$.

In Appendix 5.6, we define the matrices $\mathbf{A}_t, \mathbf{B}_t, \mathbf{E}_t$ given by (5.23),(5.25) to get the agent's trajectory, $\mathbf{x}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{E}_t \mathbf{w}_t$. Similarly, the matrices $\mathbf{T}_t, \mathbf{q}_t, \mathbf{F}_t$ given by (5.24),(5.25), give the obstacles' trajectory as $\mathbf{o}_t = \mathbf{T}_t o_t + \mathbf{q}_t + \mathbf{R}_t \mathbf{n}_t$. Defining $\mathbf{h}_t, \mathbf{M}_t, \mathbf{K}_t$ given by (5.21),(5.22), the control policies $\Pi_{\boldsymbol{\theta}_t}(\mathbf{x}_t, \mathbf{o}_t)$ along the prediction horizon are $\mathbf{u}_t = \mathbf{h}_t + \mathbf{M}_t \mathbf{w}_t + \mathbf{K}_t \mathbf{F}_t \mathbf{n}_t$, parameterized by $\boldsymbol{\theta}_t = (\mathbf{h}_t, \mathbf{M}_t, \mathbf{K}_t)$.

Note that although $\mathbf{o}_t$ doesn't necessarily depend on $\mathbf{x}_t$ ($\because \mathbf{n}_t$ may be independent from $\mathbf{w}_t$), the policies $\Pi_{\boldsymbol{\theta}_t}(\mathbf{x}_t, \mathbf{o}_t)$ modify the distribution of $\mathbf{x}_t$ in response to $\mathbf{o}_t$. Solving (5.5) over open-loop sequences (i.e., $\Pi_{\boldsymbol{\theta}_t}(\mathbf{x}_t, \mathbf{o}_t) = \mathbf{h}_t$) can be conservative because the agent-obstacle trajectories $(\mathbf{x}_t, \mathbf{o}_t)$ from a single control sequence $\mathbf{u}_t = \mathbf{h}_t$ must satisfy all the constraints regardless of the realizations of $\mathbf{w}_t, \mathbf{n}_t$.

Collision Avoidance Constraint Reformulation by Dualization

The obstacle avoidance constraint between the agent and the i^{th} obstacle at the k^{th} time step along the prediction horizon is given by $\mathbb{S}_k(x_{k|t}) \cap \mathbb{S}_k^i(o_{k|t}^i) = \emptyset$. This can be equivalently expressed as $\text{dist}(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i)) > 0$[1] where $\text{dist}(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i))$ is the solution of the convex optimization problem

$$\begin{aligned}\text{dist}(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i)) &= \min_{z_1 \in \mathbb{S}_k(x_{k|t}), z_2 \in \mathbb{S}_k^i(o_{k|t}^i)} ||z_1 - z_2||_2 \\ &= \min_{z_1, z_2} \quad ||z_1 - z_2||_2 \\ &\quad \text{s.t} \quad G R_k^\top (z_1 - p_{k|t}) \preceq_{\mathcal{K}} g, \\ &\qquad G^i R_k^{i\top} (z_2 - p_{k|t}^i) \preceq_{\mathcal{K}} g^i. \end{aligned} \tag{5.7}$$

In the following proposition, we use the above formulation (5.7) and Lagrange duality to express the set intersection problem $\mathbb{S}_k(x_{k|t}) \cap \mathbb{S}_k^i(o_{k|t}^i) = \emptyset$ as a convex feasibility problem.

Proposition 5 *Given the state and orientation of the agent $x_{k|t}, R_k$, and state and orientation of the i^{th} obstacle $o_{k|t}^i, R_k^i$ at the k^{th} prediction time step, we have*

$$\begin{aligned} & dist(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i)) > 0 \\ \Leftrightarrow & \exists \lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^* : \; -\lambda_{k|t}^{i\top}(G R_k^\top (p_{k|t} - p_{k|t}^i) + g) - \nu_{k|t}^{i\top} g^i > 0, \\ & \|\lambda_{k|t}^{i\top} G R_k^\top\|_2 \leq 1, \lambda_{k|t}^{i\top} G R_k^\top = -\nu_{k|t}^{i\top} G^i R_{k|t}^{i\top}. \end{aligned} \tag{5.8}$$

The feasibility of (5.8) gives a separating hyperplane with normal vector $\mu = -\lambda_{k|t}^{i\top} G R_k^\top$ for the sets $\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i)$. For any $z_1 \in \mathbb{S}_k(x_{k|t})$, $z_2 \in \mathbb{S}_k^i(o_{k|t}^i)$ and $\lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^*$ feasible for (5.8), we have $\lambda_{k|t}^{i\top}(g - G R_k^\top (z_1 - p_{k|t})) \geq 0$, $\nu_{k|t}^{i\top}(g^i - G^i R_k^{i\top}(z_1 - p_{k|t}^i)) \geq 0$. Adding these two inequalities, we get $\mu^\top z_1 - \mu^\top z_2 \geq -\lambda_{k|t}^{i\top}(G R_k^\top (p_{k|t} - p_{k|t}^i) + g) - \nu_{k|t}^{i\top} g^i > 0$.

Next, we reformulate (5.8) to address the non-determinism arising from the uncertainty in positions $p_{k|t} = C x_{k|t} + c_k$, $p_{k|t}^i = C o_{k|t}^i$ along the prediction horizon due to $\mathbf{w}_t, \mathbf{n}_t$.

Deterministic Constraint Reformulation

To specify the constraints (5.5e) for our MPC optimization problem, we derive deterministic reformulations for the obstacle avoidance constraints (5.8) and the state-input constraints $\mathcal{XU} = \{(x, u) \mid F_j^x x + F_j^u u \leq f_j \; \forall j \in \mathbb{I}_1^J\}$ for the state predictions $\mathbf{x}_t$ in closed-loop with (5.6), for each uncertainty description: **D1**, **D2** and **D3**.

[1] In practice, we replace > 0 with $\geq d_{min}$ for some small $d_{min} > 0$.

We introduce the constant matrices $S_k^x, S_k^u, S_k^{o,i}$ such that $S_k^x \mathbf{x}_t = x_{k|t}$, $S_k^u \mathbf{u}_t = u_{k|t}$ and $S_k^{o,i} \mathbf{o}_t = o_{k|t}^i$. Let $\mathbf{P}$ be a permutation matrix such that $[\mathbf{w}_t^\top \ \mathbf{n}_t^\top]^\top = \mathbf{P}\mathbf{v}_t$ where $\mathbf{v}_t = [w_{t|t}^\top \ n_{t|t}^\top \dots w_{t+N-1|t}^\top \ n_{t+N-1|t}^\top]^\top$. Also, define $\lambda_{k|t} = [\lambda_{k|t}^{1\top} \dots, \lambda_{k|t}^{M\top}]^\top$, $\boldsymbol{\lambda}_t = [\lambda_{t+1|t}^\top \dots, \lambda_{t+N|t}^\top]^\top$ (similarly, $\nu_{k|t}$, $\boldsymbol{\nu}_t$). For a given sequence of noise realisations $(\mathbf{w}_t, \mathbf{n}_t)$, define the set of feasible joint realizations of the agent and the obstacles $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t)$ in the lifted-space $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t, \boldsymbol{\lambda}_t, \boldsymbol{\nu}_t)$ as

$$\mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t) = \left\{ \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \middle| \begin{array}{l} \forall k \in \mathbb{I}_t^{t+N-1}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J : \\ \lambda_{k+1|t}^i, \nu_{k+1|t}^i \in \mathcal{K}^*, \|\lambda_{k+1|t}^{i\top} G R_{k+1}^\top\|_2 \leq 1, \\ \lambda_{k+1|t}^{i\top} G R_{k+1}^\top = -\nu_{k+1|t}^{i\top} G^i R_{k+1}^{i\top}, \\ \lambda_{k|t}^{i\top} (G R_k^\top (C(S_k^x \mathbf{x}_t - S_k^{o,i} \mathbf{o}_t) + c_t) + g) \\ \quad < -\nu_{k|t}^{i\top} g_i, \\ F_j^x S_k^x \mathbf{x}_t + F_j^u S_k^u \mathbf{u}_t \leq f_j, \\ \mathbf{x}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{E}_t \mathbf{w}_t, \\ \mathbf{o}_t = \mathbf{T}_t o_t + \mathbf{q}_t + \mathbf{F}_t \mathbf{n}_t \end{array} \right\}. \tag{5.9}$$

We now express the reformulations for the considered uncertainty descriptions using this set.

Robust Formulation for Uncertainty Description D1

We seek to tighten the obstacle avoidance constraints, and state-input constraints to find $\mathbf{u}_t$ such that the tuple $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t)$ satisfies the aforementioned constraints for all realisations of $[w_{k|t}^\top \ n_{k|t}^\top]^\top \in \mathcal{D}$, $\forall k \in \mathbb{I}_t^{t+N-1}$. We can write this formally as

$$\mathcal{C}(\mathbf{D1}) = \bigcap_{\forall \mathbf{v}_t \in \mathcal{D}^N} \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t) \tag{5.10}$$

where $\mathcal{D}^N = \{\mathbf{d} | \ \|\mathbf{\Gamma} \mathbf{d}\|_\infty \leq \gamma\}$, $\mathbf{\Gamma} = I_N \otimes \Gamma$.

Chance Constraint Formulation for Uncertainty Description D2

For uncertainty description **D2**, we have that $[w_t^\top, n_t^\top]^\top \sim \mathcal{N}(0, \Sigma)$, i.i.d. $\forall t \geq 0$. Since the uncertainties now have unbounded support, we adopt a chance constrained formulation, where for some $0 < \epsilon << 1$, we find $\mathbf{u}_t$ such that the tuple $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t, \boldsymbol{\lambda}_t, \boldsymbol{\nu}_t)$ satisfies the obstacle avoidance constraints (5.8) and state-input constraints with probability greater than $1 - \epsilon$, given that $[w_{k|t}^\top \ n_{k|t}^\top]^\top \sim \mathcal{N}(\mu, \Sigma)$, $\forall k \in \mathbb{I}_t^{t+N-1}$. Formally, we

write this set as

$$\mathcal{C}(\mathbf{D2}) = \left\{ \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \middle| \mathbb{P}\left(\begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \in \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t) \right) \geq 1 - \epsilon \right\} \tag{5.11}$$

where the probability measure $\mathbb{P}(\cdot)$ is over $\mathbf{v}_t$, and constructed as the product of measures of N i.i.d Gaussian distributions $\mathcal{N}(0, \Sigma)$.

Distributionally Robust Formulation for Uncertainty Description D3

For uncertainty description **D3**, we have that $[w_t^\top, n_t^\top]^\top$ are i.i.d. $\forall t \geq 0$ and have known mean and covariance, $\mathbb{E}([w_t^\top, n_t^\top]^\top) = 0$, $\mathbb{E}([w_t^\top, n_t^\top]^\top [w_t^\top, n_t^\top]) = \Sigma$. Denote the mean and covariance of the stacked random variables $\mathbf{v}_t$ as $\mathbf{0} = [0^\top \ldots, 0^\top]^\top$, $\boldsymbol{\Sigma} = I_N \otimes \Sigma$. Now define the *ambiguity set* [107] as

$$\mathcal{P} = \{\text{Probability distributions with } \mathbb{E}(\mathbf{v}_t) = \mathbf{0}, \mathbb{E}(\mathbf{v}_t \mathbf{v}_t^\top) = \boldsymbol{\Sigma}\}.$$

We adopt a distributionally robust chance constrained formulation, where for some $0 < \epsilon << 1$, we find $\mathbf{u}_t$ such that the tuple $(\mathbf{x}_t, \mathbf{u}_t, \mathbf{o}_t, \boldsymbol{\lambda}_t, \boldsymbol{\nu}_t)$ satisfies the obstacle avoidance constraints (5.8) and state-input constraints with probability greater than $1 - \epsilon$, for all probability distributions in $\mathcal{P}$. Formally, we write this set as

$$\mathcal{C}(\mathbf{U3}) = \left\{ \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \middle| \inf_{\substack{P \in \mathcal{P} \\ \mathbf{v}_t \sim P}} \mathbb{P}\left(\begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \in \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t) \right) \geq 1 - \epsilon \right\}. \tag{5.12}$$

The next theorem provides deterministic reformulations of the constraint sets presented above, and establishes the feasible set of (5.5) in terms of the policy parameters $\boldsymbol{\theta}_t = (\mathbf{h}_t, \mathbf{M}_t, \mathbf{K}_t)$ and Lagrange multipliers $\boldsymbol{\lambda}_t, \boldsymbol{\nu}_t$.

Theorem 5.1 *For the agent* (5.1) *in closed-loop with policy* (5.6) *and obstacles modelled by* (5.3)*, define the following functions* $\forall k \in \mathbb{I}_t^{t+N}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J$ *in the dual variables and policy parameters* $\boldsymbol{\theta}_t = (\mathbf{h}_t, \mathbf{M}_t, \mathbf{K}_t)$*:*

$$Y^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) = -\lambda^{i\top}_{k|t} g - \nu^{i\top}_{k|t} g^i - \lambda^{i\top}_{k|t} G R^\top_k c_t \\ - \lambda^{i\top}_{k|t} G R^\top_k C(S^x_k(\mathbf{A}_t x_t + \mathbf{B}\mathbf{h}_t) - S^{o,i}_k(\mathbf{T}_t o_t + \mathbf{q}_t)), \tag{5.13}$$

$$Z^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) = \lambda^{i\top}_{k|t} G R^\top_k C \begin{bmatrix} (S^x_k(\mathbf{B}_t\mathbf{M}_t + \mathbf{E}_t))^\top \\ \mathbf{F}^\top_t (S^x_k \mathbf{B}_t \mathbf{K}_t - S^{o,i}_k)^\top \end{bmatrix}^\top \tag{5.14}$$

$$\bar{Y}^j_{k|t}(\boldsymbol{\theta}_t) = f_j - F^x_j S^x_k(\mathbf{A}_t x_t + \mathbf{B}\mathbf{h}_t) - F^u_j S^u_k \mathbf{h}_t, \tag{5.15}$$

$$\bar{Z}^j_{k|t}(\boldsymbol{\theta}_t) = \begin{bmatrix} (F^x_j S^x_k(\mathbf{B}_t\mathbf{M}_t + \mathbf{E}_t) + F^u_j S^u_k \mathbf{M}_t)^\top \\ ((F^x_j S^x_k + F^u_j S^u_k)\mathbf{B}_t \mathbf{K}_t \mathbf{F}_t)^\top \end{bmatrix}^\top \tag{5.16}$$

Then deterministic reformulations of the feasible sets of (5.5) *are given as follows:*

1. *For uncertainty description* **D1**, *the feasible set is given by*

$$\mathcal{F}_t(\mathbf{D1}) = \left\{ \begin{pmatrix} \mathbf{h}_t \\ \mathbf{M}_t \\ \mathbf{K}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{pmatrix} \middle| \begin{array}{l} \forall k \in \mathbb{I}_t^{t+N-1}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J : \\ \lambda^i_{k+1|t}, \nu^i_{k+1|t} \in \mathcal{K}^* \| \lambda^{i\top}_{k+1|t} G R^\top_{k+1} \|_2 \le 1, \\ \lambda^{i\top}_{k+1|t} G R^\top_{k+1} = -\nu^{i\top}_{k+1|t} G^i R^{i\top}_{k+1}, \\ Y^i_{k+1|t}(\boldsymbol{\theta}_t, \lambda^i_{k+1|t}, \nu^i_{k+1|t}) > \\ \gamma \| Z^i_{k+1|t}(\boldsymbol{\theta}_t, \lambda^i_{k+1|t}, \nu^i_{k+1|t}) \mathbf{P}\boldsymbol{\Gamma}^{-1} \|_1, \\ \bar{Y}^j_{k|t}(\boldsymbol{\theta}_t) - \gamma \| \bar{Z}^j_{k|t}(\boldsymbol{\theta}_t) \mathbf{P}\boldsymbol{\Gamma}^{-1} \|_1 \ge 0 \end{array} \right\} \tag{5.17}$$

2. *For uncertainty description* **D2**, *the feasible set is inner-approximated by*

$$\mathcal{F}_t(\mathbf{D2}) = \left\{ \begin{pmatrix} \mathbf{h}_t \\ \mathbf{M}_t \\ \mathbf{K}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{pmatrix} \middle| \begin{array}{l} \forall k \in \mathbb{I}_t^{t+N-1}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J : \\ \lambda^i_{k+1|t}, \nu^i_{k+1|t} \in \mathcal{K}^* \| \lambda^{i\top}_{k+1|t} G R^\top_{k+1} \|_2 \le 1, \\ \lambda^{i\top}_{k+1|t} G R^\top_{k+1} = -\nu^{i\top}_{k+1|t} G^i R^{i\top}_{k+1}, \\ Y^i_{k+1|t}(\boldsymbol{\theta}_t, \lambda^i_{k+1|t}, \nu^i_{k+1|t}) > \\ \gamma_{ca} \| Z^i_{k+1|t}(\boldsymbol{\theta}_t, \lambda^i_{k+1|t}, \nu^i_{k+1|t}) \mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}} \|_2, \\ \bar{Y}^j_{k|t}(\boldsymbol{\theta}_t) \ge \gamma_{xu} \| \bar{Z}^j_{k|t}(\boldsymbol{\theta}_t) \mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}} \|_2 \end{array} \right\} \tag{5.18}$$

   where $\gamma_{ca} = \Phi^{-1}(1 - \frac{\epsilon}{2NM})$ *and* $\gamma_{xu} = \Phi^{-1}(1 - \frac{\epsilon}{2NJ})$, $\Phi^{-1}(\cdot)$ *being the percentile function of the standard normal distribution.*

3. *For uncertainty description* **D3**, *the inner-approximation of the feasible set,* $\mathcal{F}_t(\mathbf{D3})$, *is the same as* $\mathcal{F}_t(\mathbf{D2})$, *except with* $\gamma_{ca} = \sqrt{\frac{2NM-\epsilon}{\epsilon}}$ *and* $\gamma_{xu} = \sqrt{\frac{2NJ-\epsilon}{\epsilon}}$.

The constraint sets characterised in Theorem 5.1 are non-convex because the terms $Y^i_{k|t}(\cdot), Z^i_{k|t}(\cdot)\ \forall k \in \mathbb{I}_{t+1}^{t+N}, i \in \mathbb{I}_1^M$ given by (5.13), (5.14) are bilinear in $\boldsymbol{\lambda}_t, \boldsymbol{\theta}_t$. In the next section, we proceed to derive a convex approximation of the constraint sets and propose a MPC formulation amenable to convex programming.

Sequential Convex Programming MPC Formulation

We linearize the bilinear terms $Y^i_{k|t}(\cdot), Z^i_{k|t}(\cdot)\ \forall k \in \mathbb{I}^{t+N}_{t+1}, i \in \mathbb{I}^M_1$ from (5.13), (5.14) for time t, about the previous solution $\boldsymbol{\theta}^*_{t-1}, \boldsymbol{\lambda}^*_{t-1}, \boldsymbol{\nu}^*_{t-1}$ to get affine functions $\mathcal{L}Y^i_{k|t}(\cdot), \mathcal{L}Z^i_{k|t}(\cdot)$ given by

$$\begin{aligned}\mathcal{L}Y^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) =& Y^i_{k|t}(\boldsymbol{\theta}_t, \lambda^{i*}_{k-1|t-1}, \nu^i_{k|t}) \\ &+ Y^i_{k|t}(\boldsymbol{\theta}^*_{t-1}, \lambda^i_{k|t} - \lambda^{i*}_{k-1|t-1}, \nu^i_{k|t}) \\ \mathcal{L}Z^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) =& Z^i_{k|t}(\boldsymbol{\theta}_t, \lambda^{i*}_{k-1|t-1}, \nu^i_{k|t}) \\ &+ Z^i_{k|t}(\boldsymbol{\theta}^*_{t-1}, \lambda^i_{k|t} - \lambda^{i*}_{k-1|t-1}, \nu^i_{k|t})\end{aligned}$$

When $\gamma > 0$, $\epsilon < \min\{NM, NJ\}$, the constraints

$$\begin{aligned}\mathcal{L}Y^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) &> \gamma \|\mathcal{L}Z^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t})\mathbf{P}\boldsymbol{\Gamma}^{-1}\|_1, \\ \mathcal{L}Y^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) &> \gamma_{ca} \|\mathcal{L}Z^i_{k|t}(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t})\mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_2\end{aligned}$$

are second-order cone (SOC) representable (LP representable in the first case) because the composition of a SOC constraint with an affine map is still an SOC constraint. Substituting these affine functions in the set definitions of $\mathcal{F}_t(\mathbf{D}_1), \mathcal{F}_t(\mathbf{D}_2), \mathcal{F}_t(\mathbf{D}_3)$, the resulting constraint sets $\tilde{\mathcal{F}}_t(\mathbf{D}_1), \tilde{\mathcal{F}}_t(\mathbf{D}_2), \tilde{\mathcal{F}}_t(\mathbf{D}_3)$ are convex. This follows from 1) convexity of the dual cone $\mathcal{K}^*$, 2) convexity of constraints $\|\lambda^{i\top}_{k|t} G R^\top_k\|_2 \leq 1, \lambda^{i\top}_{k|t} G R^\top_k = -\nu^{i\top}_{k|t} G^i R^{i\top}_k$, and 3) convexity of constraints $\bar{Y}^j_{k|t}(\boldsymbol{\theta}_t) \geq \gamma \|\bar{Z}^j_k(\boldsymbol{\theta}_t)\mathbf{P}\boldsymbol{\Gamma}^{-1}\|_1$, $\bar{Y}^j_{k|t}(\boldsymbol{\theta}_t) \geq \gamma_{xu} \|\bar{Z}^j_k(\boldsymbol{\theta}_t)\mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_2$ due to composition of SOC constraint with affine maps $\bar{Y}^j_{k|t}(\cdot), \bar{Z}^j_{k|t}(\cdot)$.

We also define a convex, quadratic cost (7.1a) of the MPC optimization problem (5.5) to penalise deviations of the agent's nominal (certainty-equivalent) state-input trajectories from a given reference trajectory $\mathbf{x}^{\text{ref}}_t = [x^{\text{ref}\top}_t, \ldots, x^{\text{ref}\top}_{t+N}]^\top$, $\mathbf{u}^{\text{ref}}_t = [u^{\text{ref}\top}_t, \ldots, u^{\text{ref}\top}_{t+N-1}]^\top$:

$$C_t(\bar{\mathbf{x}}_t, \bar{\mathbf{u}}_t) = (\mathbf{x}^{\text{ref}}_t - \bar{\mathbf{x}}_t)^\top \mathbf{Q}(\mathbf{x}^{\text{ref}}_t - \bar{\mathbf{x}}_t) + (\mathbf{u}^{\text{ref}}_t - \bar{\mathbf{u}}_t)^\top \mathbf{R}(\mathbf{u}^{\text{ref}}_t - \bar{\mathbf{u}}_t) \tag{5.19}$$

where $\mathbf{Q}, \mathbf{R} \succ 0$, $\bar{\mathbf{x}}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{h}_t$, $\bar{\mathbf{u}}_t = \mathbf{h}_t$. The resulting optimization problem for our MPC for either of the uncertainty descriptions **D1**, **D2** or **D3** is given by the convex optimization problem:

$$\begin{aligned}&\mathbf{OPT}^{CVX}_t(\mathbf{D} \in \{\mathbf{D1}, \mathbf{D2}, \mathbf{D3}\}) : \\ &\min_{\{\mathbf{h}_t, \mathbf{K}_t, \mathbf{M}_t, \boldsymbol{\lambda}_t, \boldsymbol{\nu}_t\}} \quad C_t(\bar{\mathbf{x}}_t, \bar{\mathbf{u}}_t) \\ &\qquad \text{s.t } \ \bar{\mathbf{x}}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{h}_t, \ \bar{\mathbf{u}}_t = \mathbf{h}_t, \\ &\qquad\quad \{\mathbf{h}_t, \mathbf{K}_t, \mathbf{M}_t, \boldsymbol{\lambda}_t, \boldsymbol{\nu}_t\} \in \tilde{\mathcal{F}}_t(\mathbf{D}).\end{aligned} \tag{5.20}$$

When the cone $\mathcal{K}$ is given by the positive orthant (for polytopic shapes) or the second-order cone (for ellipsoidal shapes), the optimization problem (5.20) is given by a second-order cone program which can be efficiently solved. The optimal solution to (5.20) is used to obtain the control action $u^*_{t|t}$ given by (5.6).

Remark 5.1 *The feasible set of* (5.20) *is not a convex inner-approximation of the original problem with* $\mathcal{F}_t(\mathbf{D})$*. However, at the cost of introducing several new variables, a convex-inner approximation can be obtained by enforcing the collision avoidance constraints for all points in the convex relaxation of the bilinear equalities* (5.13), (5.14) *given by McCormick envelopes[89]. An investigation along these lines is left for future research.*

5.4 Results

In this section, we demonstrate our MPC formulation [2] for an unprotected left turn at a traffic intersection in CARLA. We evaluate the MPC formulations for each uncertainty description $\mathbf{D} \in \{\mathbf{D1}, \mathbf{D2}, \mathbf{D3}\}$ to highlight the benefit of the proposed collision avoidance formulation.

Motion Planning Setup

The AV model (5.1) is obtained by linearizing the kinematic bicycle model about a reference trajectory. The reference trajectory is computed offline by solving a nonlinear trajectory optimization problem to track a high-level route provided by the CARLA waypoint API, while accounting for actuation limits and lane constraints. The predictions for the target vehicle are given by Multipath [37] to produce a normally distributed trajectory prediction. The mean and covariance of this distribution are used to define the uncertainty descriptions **D2**, **D3** and obtain the model (5.3). For uncertainty **D1**, Γ, γ are chosen to construct a polytopic outer-approximation of the $1-\epsilon$ confidence set of normal distribution, where $\epsilon = 0.05$. Both vehicle geometries are modeled as polytopes. The MPC optimization problem (5.20) is solved using Gurobi [57] and when an infeasible problem is encountered, a braking control is commanded. The corresponding control action $u^*_{t|t}$ is given as a reference to a low-level control module that sets the vehicle's steering, throttle, and brake inputs. These loops are repeated until all vehicles in the scenario reach their destination.

Policies

We evaluate the following set of policies for each uncertainty description: We compare the following control policies for the agent corresponding to the different uncertainty descriptions: 1) Robust MPC (RMPC) for **D1**, 2) Stochastic MPC (SMPC) for **D2**, and 3) Distributionally Robust MPC (DRMPC) for **D3**.

- Robust MPC (**RMPC**), given by (5.20) with uncertainty **D1**

[2] Experiments were run on a computer with a Intel i9-9900K CPU, 32 GB RAM, and a RTX 2080 Ti GPU.

- Robust MPC Baseline (**RMPC_BL**), given by the collision avoidance formulation from Chapter 3, but with robust constraint tightening for **D1** and fixed risk allocation
- Stochastic MPC (**SMPC**), given by (5.20) with uncertainty **D2**
- Stochastic MPC Baseline (**SMPC_BL**), given by the SMPC from Chapter 3 but with fixed risk allocation
- Distributionally Robust MPC (**DMPC**), given by (5.20) with uncertainty **D3**
- Distributionally Robust MPC Baseline (**DMPC_BL**), given by the collision avoidance formulation from Chapter 3, but with distributionally robust constraint tightening for **D3** and fixed risk allocation

Evaluation Metrics

To evaluate the performance of our approach with respect to the baseline policies, we introduce a set of closed-loop behavior metrics. A desirable planning framework enables high mobility without being overly conservative, allowing the timely completion of the driving task, while maintaining passenger comfort. The computation time should also not be exorbitant to allow for real-time processing of updated scene information.

The following metrics are used to assess these factors:

- **Mobility**: We record the time the EV takes to reach its goal: $\tilde{\mathcal{T}}_{episode}$, normalised by the time taken to reach the same goal without target vehicles.
- **Comfort**: We record (1) the peak lateral acceleration: $\tilde{\mathcal{A}}_{lat}$, normalised by the peak lateral acceleration in the absence of TVs, (2) the average longitudinal jerk: $\bar{\mathcal{J}}_{long}$ and (3) the average lateral jerk: $\bar{\mathcal{J}}_{lat}$. High values are undesirable, linked to sudden braking or steering.
- **Conservatism**: We record (1) the deviation of the closed-loop trajectory from the trajectory obtained without TVs: $\Delta\tau$, (2) the smallest Euclidean distance observed between the EV and TV: $\bar{d}_{min}$ and (3) the feasibility % of the SMPC optimization problem: $\mathcal{F}$. While an increase in (1) and (2) corresponds to higher safety, a large value could indicate an over-conservative planner and resulting low values of (3).
- **Computational Efficiency**: This is the average time taken by the solver: $\bar{\mathcal{T}}_{solve}$; lower is better.

We run each policy for 10 initial conditions by varying the initial speed and distance from the intersection. The computed metrics are presented in Table 5.1.

Table 5.1: Closed-loop performance comparison across all scenarios.

Policy	Mobility	Comfort			Conservatism			Efficiency
	$\tilde{\mathcal{T}}_{episode}$	$\tilde{\mathcal{A}}_{lat}$	$\bar{\mathcal{J}}_{long}$ $(\frac{m}{s^3})$	$\bar{\mathcal{J}}_{lat}$ $(\frac{m}{s^3})$	$\Delta\tau$ (m)	$\bar{d}_{min}$ (m)	$\mathcal{F}$ (%)	$\bar{\mathcal{T}}_{solve}$ (ms)
RMPC_BL	1.166	1.712	4.998	6.016	1.396	10.174	98.59	**30.280**
RMPC	**1.094**	**1.512**	**2.531**	**5.691**	**1.242**	**4.884**	**100.00**	35.949
SMPC_BL	1.099	1.830	3.336	6.824	1.130	5.270	98.74	**33.063**
SMPC	**1.086**	**1.027**	**2.098**	**6.554**	**1.242**	**3.578**	**100.00**	52.319
DMPC_BL	1.370	1.620	3.971	8.595	2.707	11.582	93.09	**43.372**
DMPC	**1.252**	**1.159**	**2.143**	**4.378**	**1.734**	**7.917**	**97.51**	61.400

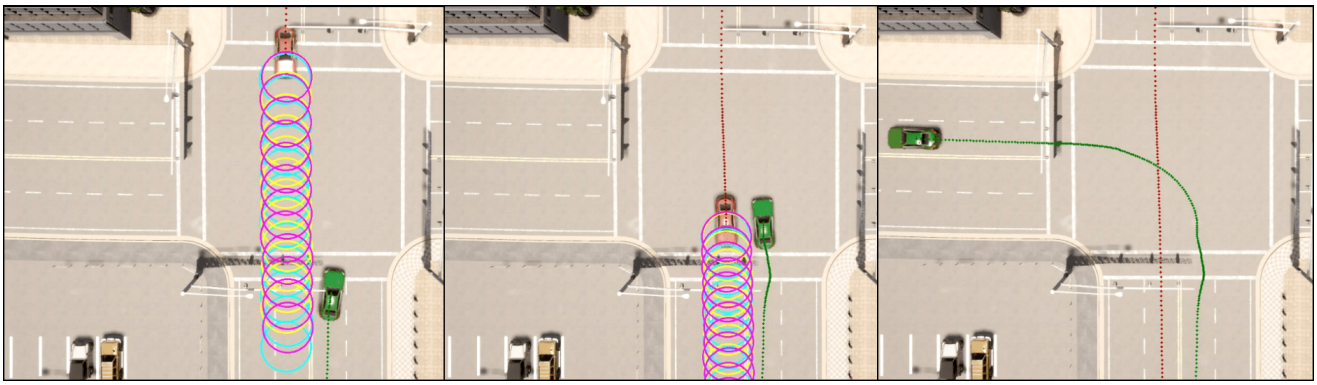
Discussion

We observe that the tighter collision avoidance formulation proposed in this chapter achieves better results than the baseline, all across mobility, comfort, and conservatism. The additional dual variables increase the solve times but are still competitive for real-time control. In general, we see that the distributionally robust approach is more conservative than the stochastic and robust counterparts.

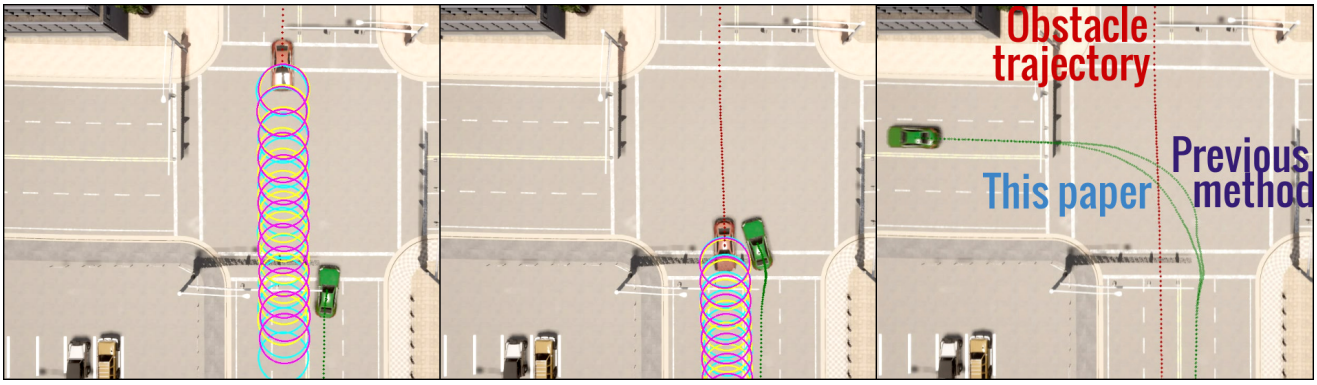
To qualitatively depict the difference in performance, we compare the SMPC from this chapter to the SMPC from chapter 3. In chapter 3, collision is modeled as the intersection of an ellipse (for the obstacle) and a circle (for the agent), and the free space is inner-approximated using an affine constraint. While the latter is robust to deviations of the agent's orientation along the predictions, this approach induces conservative and undesirable maneuvers for collision avoidance. We summarise our findings in Figure 5.1, and observe that the new approach allows for a tighter left-turn in Figure 5.1b.

5.5 Conclusion

We proposed convex MPC formulations for collision avoidance with dynamic obstacles and prediction uncertainties given by 1) Polytopic supports, 2) Gaussian distributions and 3) Arbitrary distributions with known mean and variance, and numerically validated our approach at traffic intersection scenarios. The key idea is to use dual perspective of collision avoidance, and tighten the reformulated dual feasibility problem for different prediction uncertainties. This approach can be applied for agent and obstacle geometries given by general convex sets such as ellipsoids and polytopes.



(a) SMPC from chapter 3, [100]



(b) SMPC from this chapter

Figure 5.1: Closed-loop trajectories of the agent (green) with the SMPC from this chapter versus that of chapter 2. The coloured ellipses are the confidence sets given by the multi-modal predictions for the obstacle (red). The improved collision avoidance formulation allows for a tighter turn. Video: `https://youtu.be/wgqO36a1SU8`

5.6 Appendix

Matrix definitions

$$\mathbf{h}_t = [h_{t|t}^\top \dots h_{t+N-1|t}^\top]^\top, \mathbf{K}_t = \text{blkdiag}\left(K_{t|t}, \dots, K_{t+N-1|t}\right), \tag{5.21}$$

$$\mathbf{M}_t = \begin{bmatrix} O & \dots & \dots & O \\ M_{t,t+1|t} & O & \dots & O \\ \vdots & \vdots & \vdots & \ddots \\ M_{t,t+N-1|t} & \dots & M_{t+N-2,t+N-1|t} & O \end{bmatrix}, \tag{5.22}$$

$$\mathbf{A}_t = \begin{bmatrix} I_{n_x} \\ A_t \\ \vdots \\ \prod_{k=t}^{t+N-1} A_k \end{bmatrix}, \mathbf{B}_t = \begin{bmatrix} O & \dots & \dots & O \\ B_t & O & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} A_k B_t & \dots & \dots & B_{t+N-1} \end{bmatrix}, \tag{5.23}$$

$$\mathbf{T}_t = \begin{bmatrix} I_{n_x} \\ T_t \\ \vdots \\ \prod_{k=t}^{t+N-1} T_k \end{bmatrix}, \mathbf{q}_t = \begin{bmatrix} O \\ q_t \\ \vdots \\ q_{t+N-1} + \sum_{k=t}^{t+N-1} \prod_{l=k+1}^{t+N-1} T_l q_k \end{bmatrix}, \tag{5.24}$$

$$\mathbf{E}_t = \begin{bmatrix} O & \dots & O \\ E_t & \dots & O \\ \vdots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} A_k E_t & \dots & E_{t+N-1} \end{bmatrix}, \mathbf{F}_t = \begin{bmatrix} O & \dots & \dots & O \\ F_t & O & \dots & O \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{k=t+1}^{t+N-1} T_k F_t & \dots & \dots & F_{t+N-1} \end{bmatrix} \tag{5.25}$$

Proof of Proposition 5

Consider the Lagrangian of the optimization problem (5.7), $\mathcal{L}(z_1, z_2, \lambda_{k|t}^i, \nu_{k|t}^i) = ||z_1 - z_2||_2 + \lambda_{k|t}^{i\top}(GR_k^\top(z_1 - p_{k|t}) - g) + \nu_{k|t}^{i\top}(G^i R_k^{i\top}(z_2 - p_{k|t}^i) - g^i)$ where the Lagrange multipliers $\lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^*$. Define the dual objective as

$$\begin{aligned} d(\lambda_{k|t}^i, \nu_{k|t}^i) &= \inf_{z_1,z_2} \mathcal{L}(z_1, z_2, \lambda_{k|t}^i, \nu_{k|t}^i) \\ &= -\lambda_{k|t}^{i\top}(GR_k^\top p_{k|t} + g) - \nu_{k|t}^{i\top}(GR_k^{i\top} p_{k|t}^i + g^i) \\ &\quad - \sup_{z_1,z_2} \left(\begin{bmatrix} -R_k G^\top \lambda_{k|t}^i \\ -R_k^i G^{i\top} \nu_{k|t}^i \end{bmatrix}^\top \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - f(z_1 - z_2) \right) \end{aligned}$$

where $f(\cdot) = ||\cdot||_2$. We use the properties of convex conjugates to obtain the dual objective. The convex conjugate of $f(\cdot)$ is given by $f^*(y) = \sup_x y^\top x - f(x) = \{0 \text{ if } ||y||_2 \leq 1, \infty \text{ otherwise } \}$. Moreover if $h(x) = f(Ax)$, then $h^*(y) = \inf_{A^\top z = y} f^*(z)$. For $y =$

$-[\lambda_{k|t}^{i\top} G R_k^\top \ \nu_{k|t}^{i\top} G^i R_k^{i\top}]^\top$, $A = [I_n \ -I_n]$, $x = [z_1^\top z_2^\top]^\top$, the dual function can now be written as follows,

$$\begin{aligned} d(\lambda_{k|t}^i, \nu_{k|t}^i) &= -\lambda_{k|t}^{i\top}(GR_k^\top p_{k|t} + g) - \nu_{k|t}^{i\top}(GR_k^{i\top} p_{k|t}^i + g^i) - h^*(y) \\ &= -\lambda_{k|t}^{i\top}(GR_k^\top p_{k|t} + g) - \nu_{k|t}^{i\top}(GR_k^{i\top} p_{k|t}^i + g^i) \end{aligned}$$

for $\|\lambda_{k|t}^{i\top} G R_k^\top\| \leq 1, \|\nu_{k|t}^{i\top} G^i R_k^{i\top}\|_2 \leq 1, R_k G^\top \lambda_{k|t}^i + R_{k|t}^i G_{k|t}^{i\top} \nu_{k|t}^i = 0$. The dual optimization problem for (5.7) can now be written as

$$\begin{aligned} \max_{\lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^*} & -\lambda_{k|t}^{i\top}(GR_k^\top p_{k|t} + g) - \nu_{k|t}^{i\top}(GR_k^{i\top} p_{k|t}^i + g^i) \\ \text{s.t } & \|\lambda_{k|t}^{i\top} G R_k^\top\|_2 \leq 1, \|\nu_{k|t}^{i\top} G^i R_k^{i\top}\|_2 \leq 1, \\ & R_k G^\top \lambda_{k|t}^i + R_{k|t}^i G_{k|t}^{i\top} \nu_{k|t}^i = 0. \end{aligned} \tag{5.26}$$

Since the sets $\{e | Ge \preceq_{\mathcal{K}} g\}, \{o | G^i o \preceq_{\mathcal{K}} g^i\}$ are non-empty, the feasible set of (5.7) has a non-empty interior. By Slater's condition, strong duality holds and $\text{dist}(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i))$ is equal the optimal objective of (5.26). Thus, we rewrite $\text{dist}(\mathbb{S}_k(x_{k|t}), \mathbb{S}_k^i(o_{k|t}^i)) > 0$ as

$$\begin{aligned} & \max_{\substack{\lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^*, \|\lambda_{k|t}^{i\top} G R_k^\top\|_2 \leq 1 \\ R_k G^\top \lambda_{k|t}^i = -R_{k|t}^i G_{k|t}^{i\top} \nu_{k|t}^i}} d(\lambda_{k|t}^i, \nu_{k|t}^i) > 0 \\ \Leftrightarrow & \exists \lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^* : -\lambda_{k|t}^{i\top}(GR_k^\top (p_{k|t} - p_{k|t}^i) + g) - \nu_{k|t}^{i\top} g^i > 0, \\ & \|\lambda_{k|t}^{i\top} G R_k^\top\|_2 \leq 1, R_k G^\top \lambda_{k|t}^i = -R_{k|t}^i G_{k|t}^{i\top} \nu_{k|t}^i \end{aligned}$$

■

Proof of Theorem 5.1

For all $k \in \mathbb{I}_t^{t+N-1}$, $i \in \mathbb{I}_1^M$, define the sets

$$\mathcal{S}_{k|t}^i(\mathbf{w}_t, \mathbf{n}_t) = \left\{ \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \\ \mathbf{o}_t \\ \boldsymbol{\lambda}_t \\ \boldsymbol{\nu}_t \end{bmatrix} \middle| \begin{array}{l} \lambda_{k|t}^i, \nu_{k|t}^i \in \mathcal{K}^*, \|\lambda_{k|t}^{i\top} G R_k^\top\|_2 \leq 1, \\ \lambda_{k|t}^{i\top} G R_k^\top = -\nu_{k|t}^{i\top} G^i R_k^{i\top}, \\ \lambda_{k|t}^{i\top}(GR_k^\top (C(S_k^x \mathbf{x}_t - S_k^{o,i} \mathbf{o}_t) + c_t) + g) \\ \quad < -\nu_{k|t}^{i\top} g_i, \\ \mathbf{x}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{E}_t \mathbf{w}_t, \\ \mathbf{o}_t = \mathbf{T}_t o_t + \mathbf{q}_t + \mathbf{F}_t \mathbf{n}_t \end{array} \right\}$$

$$\bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t) = \left\{ (\mathbf{x}_t, \mathbf{u}_t) \middle| \begin{array}{l} F_j^x S_{k+1}^x \mathbf{x}_t + F_j^u S_k^u \mathbf{u}_t \leq f_j, \ \forall j \in \mathbb{I}_1^J \\ \mathbf{x}_t = \mathbf{A}_t x_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{E}_t \mathbf{w}_t \end{array} \right\}$$

and see that

$$\mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t) = \bigcap_{k=t}^{t+N-1} \left(\bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t) \bigcap_{i=1}^{M} \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t) \right) \tag{5.27}$$

We proceed to derive the feasible set for each case as follows.
1) We use (5.27) to express (5.10) as

$$\mathcal{C}(\mathbf{D1}) = \bigcap_{k=t}^{t+N-1} \left(\bigcap_{\mathbf{v}_t \in \mathcal{D}^N} \bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t) \bigcap_{i=1}^{M} \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t) \right)$$

For any $k \in \mathbb{I}_{t+1}^{t+N}$, $i \in \mathbb{I}_1^M$, a feasible point in $\bigcap_{\mathbf{v}_t \in \mathcal{D}^N} \mathcal{S}^i_{k|t}(\mathbf{w}_t, \mathbf{n}_t)$ satisfies the constraints

$$\begin{aligned} &- \lambda^{i\top}_{k|t}(GR_k^\top(C(S_k^x \mathbf{x}_t - S_k^{o,i}\mathbf{o}_t) + c_t) + g) > \nu^{i\top}_{k|t} g_i, \forall \mathbf{v}_t \in \mathcal{D}^N \\ &\lambda^i_{k|t}, \nu^i_{k|t} \in \mathcal{K}^*, \|\lambda^{i\top}_{k|t} G R_k^\top\|_2 \leq 1, \lambda^{i\top}_{k|t} G R_k^\top = -\nu^{i\top}_{k|t} G^i R_k^{i\top}. \end{aligned}$$

Plugging in the closed-loop evolution of $\mathbf{x}_t, \mathbf{o}_t$ using $\mathbf{u}_t = \mathbf{h}_t + \mathbf{M}_t \mathbf{w}_t + \mathbf{K}_t \mathbf{F}_t \mathbf{n}_t$, we can rewrite this using the functions $Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t})$, $Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t})$ as

$$\begin{aligned} &Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) + Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \begin{bmatrix} \mathbf{w}_t \\ \mathbf{n}_t \end{bmatrix} > 0, \ \forall \mathbf{v}_t \in \mathcal{D}^N, \\ &\lambda^i_{k|t}, \nu^i_{k|t} \in \mathcal{K}^*, \|\lambda^{i\top}_{k|t} G R_k^\top\|_2 \leq 1, \lambda^{i\top}_{k|t} G R_k^\top = -\nu^{i\top}_{k|t} G^i R_k^{i\top}. \end{aligned}$$

The first inequality in the last implication can be equivalently expressed without the quantifier $\forall \mathbf{v}_t \in \mathcal{D}^N$ as

$$\begin{aligned} &Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) + \min_{\mathbf{v}_t \in \mathcal{D}^N} Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \mathbf{P} \mathbf{v}_t > 0 \\ \Leftrightarrow &Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) - \max_{\|\tilde{\mathbf{d}}\|_\infty \leq 1} -Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \mathbf{P} \gamma \boldsymbol{\Gamma}^{-1} \tilde{\mathbf{d}} > 0 \\ \Leftrightarrow &Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) - \gamma \| Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \mathbf{P} \boldsymbol{\Gamma}^{-1} \|_1 > 0 \end{aligned}$$

where the last implication is obtained by noting that $\|\cdot\|_1$ is the dual norm of $\|\cdot\|_\infty$. For the state-input constraints, we similarly have for each $j \in \mathbb{I}_1^J$,

$$\begin{aligned} &F_j^x S_{k+1}^x \mathbf{x}_t + F_j^u S_k^u \mathbf{u}_t \leq f_j, \ \forall \mathbf{v}_t \in \mathcal{D}^N \\ \Leftrightarrow &\bar{Y}^j_k(\boldsymbol{\theta}_t) + \bar{Z}^j_k(\boldsymbol{\theta}_t) \begin{bmatrix} \mathbf{w}_t \\ \mathbf{n}_t \end{bmatrix} > 0, \ \forall \mathbf{v}_t \in \mathcal{D}^N, \\ \Leftrightarrow &\bar{Y}^j_k(\boldsymbol{\theta}_t) - \gamma \| \bar{Z}^j_k(\boldsymbol{\theta}_t) \mathbf{P} \boldsymbol{\Gamma}^{-1} \|_1 \geq 0 \end{aligned}$$

The feasible set $\mathcal{F}_t(\mathbf{D1})$ is thus defined by the above inequalities $\forall k \in \mathbb{I}_t^{t+N-1}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J$.
2) Now we proceed to the chance-constrained case. For convenience, define $\mathbf{z}_t =$

$[\mathbf{x}_t^\top \ \mathbf{u}_t^\top \ \mathbf{o}_t^\top \ \boldsymbol{\lambda}_t^\top \ \boldsymbol{\nu}_t]^\top$. The joint constraint $\mathbb{P}(\mathbf{z}_t \in \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t)) \geq 1-\epsilon$ is difficult to reformulate in the given form, so we construct an inner-approximation to this set using individual chance constraints as follows.

$$\begin{aligned}
\mathbb{P}(\mathbf{z}_t \in \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t)) &= \mathbb{P}(\mathbf{z}_t \in \bigcap_{k=t}^{t+N-1} (\bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t) \bigcap_{i=1}^{M} \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t))) \\
\Rightarrow \mathbb{P}(\mathbf{z}_t \notin \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t)) &= \mathbb{P}(\mathbf{z}_t \notin \bigcup_{k=t}^{t+N-1} (\bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t) \bigcup_{i=1}^{M} \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t))) \\
&\leq \sum_{k=t}^{t+N-1} \Big(\mathbb{P}(\mathbf{z}_t \notin \bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t)) + \sum_{i=1}^{M} \mathbb{P}(\mathbf{z}_t \notin \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t)) \Big)
\end{aligned}$$

Thus, if each $\mathbb{P}(\mathbf{z}_t \notin \bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t)) \leq \frac{\epsilon}{2N}$ and $\mathbb{P}(\mathbf{z}_t \notin \mathcal{S}^i_{k+1|t}(\mathbf{w}_t, \mathbf{n}_t)) \leq \frac{\epsilon}{2NM}$, we get $\mathbb{P}(\mathbf{z}_t \in \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t)) = 1 - \mathbb{P}(\mathbf{z}_t \notin \mathcal{S}_t(\mathbf{w}_t, \mathbf{n}_t)) \geq 1-\epsilon$. So for any $k \in \mathbb{I}_{t+1}^{t+N-1}$, we reformulate $\mathbb{P}(\mathbf{z}_t \in \mathcal{S}^i_{k|t}(\mathbf{w}_t, \mathbf{n}_t)) \geq 1 - \frac{\epsilon}{2NM}$ using the result: $\mathbb{P}(a^\top x > b) \geq 1-\epsilon, x \sim \mathcal{N}(\mu_x, \Sigma_x) \Leftrightarrow a^\top \mu_x - \Phi^{-1}(1-\epsilon)\|a^\top \Sigma_x^{\frac{1}{2}}\|_2 > b$.

$$\begin{aligned}
&\mathbb{P}(\mathbf{z}_t \in \mathcal{S}^i_{k|t}(\mathbf{w}_t, \mathbf{n}_t)) \geq 1 - \frac{\epsilon}{2NM}, \\
\Leftrightarrow &\mathbb{P}(Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) + Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \begin{bmatrix} \mathbf{w}_t \\ \mathbf{n}_t \end{bmatrix} > 0) \geq 1-\epsilon, \\
&\lambda^i_{k|t}, \nu^i_{k|t} \in \mathcal{K}^*, \|\lambda^{i\top}_{k|t} G R_k^\top\|_2 \leq 1, \lambda^{i\top}_{k|t} G R_k^\top = -\nu^{i\top}_{k|t} G^i R_k^{i\top}, \\
\Leftrightarrow &Y^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) - \gamma_{ca} \|Z^i_k(\boldsymbol{\theta}_t, \lambda^i_{k|t}, \nu^i_{k|t}) \mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_2 > 0, \\
&\lambda^i_{k|t}, \nu^i_{k|t} \in \mathcal{K}^*, \|\lambda^{i\top}_{k|t} G R_k^\top\|_2 \leq 1, \lambda^{i\top}_{k|t} G R_k^\top = -\nu^{i\top}_{k|t} G^i R_k^{i\top}.
\end{aligned}$$

where $\gamma_{ca} = \Phi^{-1}(1 - \frac{\epsilon}{2NM})$. The joint-chance constraint $\mathbb{P}((\mathbf{x}_t, \mathbf{u}_t) \notin \bar{\mathcal{S}}_{k|t}(\mathbf{w}_t, \mathbf{n}_t)) \leq \frac{\epsilon}{2N}$ is similarly inner-approximated by imposing $\mathbb{P}(F^x_j S^x_{k+1} \mathbf{x}_t + F^u_j S^u_k \mathbf{u}_t > f_j) \leq \frac{\epsilon}{2NJ} \ \forall j \in \mathbb{I}_1^J$, which is given by

$$\bar{Y}^j_k(\boldsymbol{\theta}_t) - \gamma_{xu} \|\bar{Z}^j_k(\boldsymbol{\theta}_t) \mathbf{P}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_2 \geq 0, \ \forall j \in \mathbb{I}_1^J.$$

where $\gamma_{xu} = \Phi^{-1}(1 - \frac{\epsilon}{2NJ})$. Thus, the inner-approximation of the feasible set is given by $\mathcal{F}_t(\mathbf{D2})$, defined by the above inequalities $\forall k \in \mathbb{I}_t^{t+N-1}, \forall i \in \mathbb{I}_1^M, \forall j \in \mathbb{I}_1^J$.

3) For the distributionally robust case, the joint-chance constraints are converted to individual chance constraints in the same way as shown above. However, since only the first two moments of the probability distribution are known, we reformulate the constraint using the result: $\inf_{\substack{P \in \bar{\mathcal{P}} \\ x \sim P}} \mathbb{P}(a^\top x > b) \geq 1-\epsilon \Leftrightarrow a^\top \mu_x - \sqrt{\frac{1-\epsilon}{\epsilon}} \|a^\top \Sigma_x^{\frac{1}{2}}\|_2 > b$, where $\bar{\mathcal{P}}$ is the set of distributions with mean μ_x and variance Σ_x. Proceeding identically as in the chance-constrained case but with $\gamma_{ca} = \sqrt{\frac{2NM-\epsilon}{\epsilon}}$ and $\gamma_{xu} = \sqrt{\frac{2NJ-\epsilon}{\epsilon}}$, we get $\mathcal{F}_t(\mathbf{D3})$.

■

Part III

Efficiency

Chapter 6

Learning for Mixed-Integer Predictive Control with Parametric Sub-optimality Certificates

6.1 Overview

Multi-parametric Mixed-Integer Programming (mp-MIP) is a convenient framework for modelling various non-convex motion planning and constrained optimal control problems [66]. The mixed-integer formulation can model constraints such as collision avoidance [86], mixed-logical specifications [128] and mode transitions for hybrid dynamics [61]. The multi-parametric nature of these mp-MIPs arises from requiring to solve these problems for different initial conditions, obstacles configurations or system constraints—all of which affect the MIP solution. When Model Predictive Control (MPC) [92, 21] is used for such class of problems, an MIP has to be solved in a receding horizon fashion at each time step. However, computing solutions for MIPs is $\mathcal{NP}-$hard and challenging for real-time ($\geq$ 10Hz) applications.

There are two broad approaches towards solving these MIPs online for real-time MPC. The first approach is Explicit MPC [21, 15] which involves offline computation of the solution map of the mp-MIP explicitly as piece-wise functions over partitions of the parameter space, so that online computation is reduced to a look-up. However this approach is best suited for mp-MIPs of moderate size because the complexity of the online look-up and offline storage of partitions, increases rapidly with scale [41]. The second approach for real-time mixed-integer MPC relies on predicting warm-starts or strategies for the mp-MIP by training Machine Learning (ML) models offline on large datasets [87, 149, 18, 34]. The authors of [87, 149] use various supervised learning frameworks to predict the optimal integer variables for the mp-MIP at a given parameter so that the online computation is reduced to solving a convex program. In [18, 34], the authors define the notion of an *optimal strategy* for a mp-MIP as a mapping from pa-

rameters to the complete information required to efficiently recover an optimal solution. For multi-parametric Mixed-Integer Linear/Quadratic Programs (mp-MILPs/MIQPs), an optimal strategy is defined as *a set of integer variables and active constraints at the optimal solution*. Given an optimal strategy, an optimal solution can be recovered by solving a linear system of equations which is computationally inexpensive compared to tree search methods typically used for solving MIPs, such as Branch-and-Bound (BnB). Thus, a prediction model is trained offline to predict the optimal strategy for efficiently solving the MIPs online. However, a common issue that plagues these ML-based approaches is the inability to assess the quality of the predicted warm-start/strategy to guard against poor predictions, which can lead to sub-optimal or infeasible solution predictions. Indeed, prediction models may perform poorly for various reasons: insufficient richness of the model parameterization, distribution shift between the training and test sets, convergence of the training algorithm towards a sub-optimal minimum [142].

Contributions

1. We propose a supervised learning framework for predicting strategies to efficiently solve an MILP, along with a mechanism to measure the sub-optimality of the prediction.

2. Analogous to the duality gap in convex programming, we propose bounding functions for sub-optimality quantification of the predicted strategy by drawing inspiration from Branch-and-Bound. This enables us to efficiently recover the solution of an MILP online from the predicted strategy by solving some LPs online in parallel, and also measure the sub-optimality of the recovered solution.

3. Using ideas from multi-parametric programming, we show the parametric behaviour of our proposed strategy definition for mp-MILPs.

6.2 Problem Formulation

Consider the general formulation for Mixed-Integer MPC (MIMPC) adapted from [21]:

$$\begin{aligned} V^\star(x_t) = \min_{\substack{\boldsymbol{x}_t, \boldsymbol{u}_t, \\ \boldsymbol{\delta}_t, \boldsymbol{z}_t}} \; & ||Px_{t+N|t}||_p + \sum_{k=t}^{t+N} ||Q \begin{bmatrix} x_{k|t} \\ \delta_{k|t} \end{bmatrix} ||_p + ||Ru_{k|t}||_p, \\ \text{s.t. } & x_{k+1|t} = Ax_{k|t} + B_1 u_{k|t} + B_2 \delta_{k|t} + B_3 z_{k|t}, \\ & E_2 \delta_{k|t} + E_3 z_{k|t} \leq E_1 u_{k|t} + E_4 x_{k|t} + E_5, \\ & x_{t|t} = x_t, \; \delta_{k|t} \in \{0,1\}^{n_\delta} \; \forall k = t, .., t+N-1 \end{aligned} \tag{6.1}$$

where x_t is system state at time t, $p = 1$ or ∞, and decision variables $\boldsymbol{x}_t = [x_{t|t}, .., x_{t+N|t}]$, $\boldsymbol{u}_t$, $\boldsymbol{\delta}_t$, $\boldsymbol{z}_t$ (defined similarly) are the states, inputs, binary variables and auxiliary variables respectively. The optimal solution to (6.1) defines the MPC policy as $\pi_{MPC}(x_t) = u^\star_{t|t}$.

The optimization problem (6.1) can be expressed as a multi-parametric Mixed-Integer Linear Program (mp-MILP), with the parameters being the system state x_t. The mp-MILP can be concisely expressed as follows:

$$\begin{aligned} V^\star(b) = \min_{z,y} \quad & c^\top z + d^\top y, \\ \text{s.t.} \quad & Az + By = b, \\ & z \geq 0,\ y \in \{0,1\}^M \end{aligned} \tag{6.2}$$

with continuous decision variables $z \in \mathbb{R}^n$, binary decision variables $y \in \{0,1\}^M$ and parameters $b \in \mathbb{R}^m$. Let $z^\star(b), y^\star(b)$ be an optimal solution to (6.2) and $V^\star(b)$ be the optimal cost. For a given parameter b, let $\mathcal{F}(b)$ be the set of (z,y) feasible for (6.2) and $V(b,z,y)$ be the cost of any $(z,y) \in \mathcal{F}(b)$, with sub-optimality given by $\frac{V(b,z,y)-V^\star(b)}{|V^\star(b)|}$. Also define $\mathbb{B} = \{b \in \mathbb{R}^m | \mathcal{F}(b) \neq \emptyset\}$ as the set of parameters for which (6.2) is feasible.

In this work, we aim to exploit the parametric nature of the mp-MILP (6.2) to predict a solution $(\tilde{z}(b), \tilde{y}(b)) \in \mathcal{F}(b)$ for real-time MPC, and quantify its sub-optimality using *strategies*. The strategy maps a parameter b to an element of a finite and discrete set $\mathbb{S}$, which describes the complete information necessary to recover a feasible point $(z(b), y(b))$ for (6.2) (if it exists), formally defined next.

Definition 6.1 *A function $s : \mathbb{B} \to \mathbb{S}$ is a strategy for mp-MILP* (6.2) *if there exists a map $R(\cdot)$ such that $\forall b \in \mathbb{B} : R(b, s(b)) = (z(b), y(b)) \in \mathcal{F}(b)$, and evaluating $R(b, s(b))$ is computationally efficient compared to solving* (6.2) *directly.*

For example, in [18] the set $\mathbb{S}$ is given by all possible sets of active constraints for (6.2) and for each $b \in \mathbb{B}$, $s(b)$ picks the active constraints for a $(z,y) \in \mathcal{F}(b)$. The recovery map is then given as the solution of a linear system of equations.

The strategy $s^\star(b)$ is said to be *optimal* at $b \in \mathbb{B}$ if $R(b, s^\star(b)) = (z^\star(b), y^\star(b))$. We construct functions $V_{\text{lb}}(\cdot,\cdot)$, $V_{\text{ub}}(\cdot,\cdot)$ that satisfy the following properties:

1. For any $(z,y) \in \mathcal{F}(b)$ such that $R(b, s(b)) = (z,y)$,
$$V_{\text{lb}}(b, s(b)) \leq V(b,z,y) \leq V_{\text{ub}}(b, s(b)).$$
2. For the optimal strategy $s^\star(b)$,
$$V_{\text{lb}}(b, s^\star(b)) = V^\star(b) = V_{\text{ub}}(b, s^\star(b)).$$

For any $b \in \mathbb{B}$, we use $V_{\text{lb}}(\cdot,\cdot)$, $V_{\text{ub}}(\cdot,\cdot)$ to estimate the quality of a strategy $s(b)$ with respect to $s^\star(b)$. In particular, the sub-optimality of a predicted strategy $\tilde{s}(b)$ and the recovered solution $R(b, \tilde{s}(b)) = (\tilde{z}(b), \tilde{y}(b))$ is over-estimated as

$$\text{sub\_opt}(b, \tilde{s}(b)) = \left| \frac{V_{\text{ub}}(b, \tilde{s}(b)) - V_{\text{lb}}(b, \tilde{s}(b))}{V_{\text{lb}}(b, \tilde{s}(b))} \right|. \tag{6.3}$$

Organization

First, we present our choice of strategy $s(\cdot)$, the recovery map $R(\cdot)$, and the bounding functions $V_{\text{lb}}(\cdot)$, $V_{\text{ub}}(\cdot)$ that meet the desired properties in Section 6.3. Section 6.4 presents a supervised learning framework to approximate the optimal strategy $s^\star(b)$, and evaluate $R(b, s^\star(b))$, $V_{\text{lb}}(b, s^\star(b))$, $V_{\text{ub}}(b, s^\star(b))$ efficiently for predicting solutions to (6.1) online, and evaluate its sub-optimality. Finally, we demonstrate our approach for motion planning using MIMPC and compare against open-source and commercial MILP solvers in Section 6.5.

6.3 Strategy-based Solution to mp-MILPs

In this section, we present our design of the strategy $s(\cdot)$, the recovery map $R(b, s(b))$ and the bounding functions $V_{\text{lb}}(b, s(b))$, $V_{\text{ub}}(b, s(b))$, along with theoretical justification using ideas from the mp-MILP literature.

Preliminaries: Solving MILPs using Branch-and-Bound

Branch-and-Bound (BnB) is a tree search algorithm that solves MILPs, with each node given as the LP sub-problem

$$\begin{aligned} V^\star_{LP}(b, \text{lb}, \text{ub}) = \min_{z,y} \quad & c^\top z + d^\top y, \\ \text{s.t} \quad & Az + By = b, \\ & z \geq 0,\ \text{lb} \leq y \leq \text{ub}, \end{aligned} \tag{6.4}$$

where the binary variable bounds $\text{lb}, \text{ub} \in \{0,1\}^M$. For any $b \in \mathbb{R}^m$, let $\mathcal{F}_{LP}(b, \text{lb}, \text{ub})$, $(z^\star(b, \text{lb}, \text{ub}), y^\star(b, \text{lb}, \text{ub}))$ denote its feasible set and optimal solution respectively. At iteration i of BnB, a collection of sub-problems identified by $\mathcal{C}^i = \{\{\text{lb}^i_k, \text{ub}^i_k\}_{k=1}^{n_i}\}$ is maintained such that they form a *cover* over the set of binary sequences $\{0,1\}^M$:

$$\bigcup_{k=1}^{n^i} [\text{lb}^i_k, \text{ub}^i_k] \supseteq \{0,1\}^M \Rightarrow \bigcup_{k=1}^{n^i} \mathcal{F}_{LP}(b, \text{lb}^i_k, \text{ub}^i_k) \supseteq \mathcal{F}(b).$$

A lower bound on $V^\star(b)$ at iteration i is given as

$$\underline{V}^i(b) = \min_{k \in \{1,\ldots,n^i\}} V^\star_{LP}(b, \text{lb}^i_k, \text{ub}^i_k) \leq V^\star(b),$$

which can be shown in three steps:

1. Let $(\bar{z}, \bar{y}) = \text{argmin}\{c^\top z + d^\top y | (z,y) \in \bigcup_{k=1}^{n^i} \mathcal{F}_{LP}(b, \text{lb}^i_k, \text{ub}^i_k)\}$ and $\bar{k} \in \{1, .., n^i\}$ be the sub-problem such that $(\bar{z}, \bar{y}) \in \mathcal{F}_{LP}(b, \text{lb}^i_{\bar{k}}, \text{ub}^i_{\bar{k}})$. Then $c^\top \bar{z} + d^\top \bar{y} = V^\star_{LP}(b, \text{lb}^i_{\bar{k}}, \text{ub}^i_{\bar{k}})$ due to global optimality of the $\bar{k}$th LP sub-problem.

2. Observe that $V^\star_{LP}(b, \text{lb}^i_{\bar{k}}, \text{ub}^i_{\bar{k}}) = \underline{V}^i(b)$, because otherwise, $\exists l \in \{1, .., n_i\}$ such that $V^\star_{LP}(b, \text{lb}^i_l, \text{ub}^i_l) < V^\star_{LP}(b, \text{lb}^i_{\bar{k}}, \text{ub}^i_{\bar{k}})$, which implies the contradiction $\min\{c^\top z + d^\top y | \mathcal{F}_{LP}(b, \text{lb}^i_l, \text{ub}^i_l)\} < \min\{c^\top z + d^\top y | \bigcup_{k=1}^{n^i} \mathcal{F}_{LP}(b, \text{lb}^i_k, \text{ub}^i_k)\}$.

3. Finally since the sub-problems form a cover, $\underline{V}^i(b) = \min\{c^\top z + d^\top y | (z, y) \in \bigcup_{k=1}^{n_i} \mathcal{F}_{LP}(b, \text{lb}^i_k, \text{ub}^i_k)\} \leq \min\{c^\top z + d^\top y | (z, y) \in \mathcal{F}(b)\} = V^\star(b)$.

Define the set of indices $\mathcal{I}^i \subseteq \{1, \ldots, n^i\}$ such that their corresponding sub-problems have solutions that are also feasible for (6.2), i.e.,

$$\mathcal{I}^i = \{k \in 1, \ldots, n^i | (z^\star_{LP}(b, \text{lb}^i_k, \text{ub}^i_k), y^\star_{LP}(b, \text{lb}^i_k, \text{ub}^i_k)) \in \mathcal{F}(b)\}.$$

Then, an upper bound on $V^\star(b)$ at iteration i is given as

$$V^\star(b) \leq \bar{V}^i(b) = \begin{cases} \min_{k \in \mathcal{I}^i} V^\star_{LP}(b, \text{lb}^i_k, \text{ub}^i_k), & \mathcal{I}^i \neq \emptyset, \\ \infty & \mathcal{I}^i = \emptyset \end{cases}$$

which is evident because $V^\star(b) \leq V^\star_{LP}(b, \text{lb}^i_k, \text{ub}^i_k)\ \forall k \in \mathcal{I}^i$. If $\mathcal{I}^i = \emptyset$, often rounding heuristics are applied to some sub-problem solutions to produce a feasible solution in $\mathcal{F}(b)$. This describes the *bounding* process of BnB.

If $\underline{V}^i(b) \neq \bar{V}^i(b)$, then the search proceeds to the next iteration via the *branching* process, which constructs a new cover $\mathcal{C}^{i+1}$ from $\mathcal{C}^i$ by splitting a sub-problem, say $\{\text{lb}^i_k, \text{ub}^i_k\}$, into two new sub-problems $\{\{\text{lb}^{i+1}_k, \text{ub}^{i+1}_k\}, \{\text{lb}^{i+1}_{k+1}, \text{ub}^{i+1}_{k+1}\}\}$ by fixing one or more variables to 0 in one sub-problem, and to 1 in the other. The branching decisions depend on $\underline{V}^i(b), \bar{V}^i(b)$, the optimal sub-problem solutions, and some tree search heuristics.

The search begins with the root node given by $\mathcal{C}^0 = \{\{\mathbf{0}_M, \mathbf{1}_M\}\}$ defining the LP relaxation of (6.2). The search terminates at iteration "$\star$" when $\underline{V}^\star(b) = \bar{V}^\star(b)$ and the optimal solution is given by the feasible solution that yields $\bar{V}^\star(b)$. This optimality certificate is represented by

1. the optimal cover $\mathcal{C}^\star(b) = \{\{\text{lb}^\star_k, \text{ub}^\star_k\}_{k=1}^{n_\star}\}$ describing the LP sub-problems at the terminal iteration,

2. the optimal binary solution $y^\star(b)$ obtained from the sub-problem corresponding to the upper-bound $\bar{V}^\star(b)$.

Strategy Description for Parametric MILPs

Inspired by the optimality certificate obtained from BnB, we propose the following strategy, bounding functions and recovery map:

$$s(b) = \{\mathcal{C}^\star(b), y^\star(b)\}, \tag{6.5a}$$

$$V_{\text{lb}}(b, s(b)) = \min_{k\in\{1,\dots,n_\star\}} V^\star_{LP}(b, \text{lb}^\star_k, \text{ub}^\star_k), \tag{6.5b}$$

$$V_{\text{ub}}(b, s(b)) = \min_{Az+By^\star(b)=b,\ z\geq 0} \quad c^\top z + d^\top y^\star(b), \tag{6.5c}$$

$$R(b, s(b)) = \operatorname*{argmin}_{Az+By^\star(b)=b,\ z\geq 0} \quad c^\top z + d^\top y^\star(b). \tag{6.5d}$$

The strategy $s(\bar{b})$ for parameter $\bar{b}$ is optimal if it certifies optimality of the MILP (6.2) via $V_{\text{lb}}(\bar{b}, s(\bar{b})) = V^\star(\bar{b}) = V_{\text{ub}}(\bar{b}, s(\bar{b}))$. The next theorem highlights the parametric behaviour of the optimality certificate provided by $s(\bar{b})$, i.e., the set of parameters $\mathcal{P}_{\bar{b}}$ for which $s(\bar{b})$ remains optimal. Thus, for any parameter $b \in \mathcal{P}_{\bar{b}}$, the optimal solution can be computed via (6.5d) without BnB.

Theorem 6.1 *Let $s^\star(\bar{b}) = \{\mathcal{C}^\star(\bar{b}), y^\star(\bar{b})\}$ be the optimal strategy for solving MILP* (6.2) *with the parameter $\bar{b}$. Then there is a set of parameters $\mathcal{P}_{\bar{b}} \subset \mathbb{B}$, given by a union of convex polyhedra for which $s^\star(\bar{b})$ is also optimal,*

$$V_{\text{lb}}(b, s^\star(\bar{b})) = V^\star(b) = V_{\text{ub}}(b, s^\star(\bar{b})) \ \ \forall b \in \mathcal{P}_{\bar{b}}$$

The sets $\mathcal{P}_{\bar{b}_i}$ can be constructed using ideas from multi-parametric programming, but this approach would become intractable as the size of the problem increases. Instead, we propose a supervised classification approach to predict an optimal strategy for a given parameter in the next section. For a predicted strategy $\tilde{s}(b)$, the functions in (6.5b), (6.5c) are used to quantify its sub-optimality compared to $s^\star(b)$. If no feasible solution is found or the predicted strategy is too sub-optimal, an optimal solution can be retrieved from $\tilde{\mathcal{C}}(b)$ by solving MILP sub-problems.

6.4 LAMPOS: Learning-based Approximate MIMPC with Parametric Optimality Strategies

This section presents LAMPOS: (A) an offline supervised learning framework for strategy prediction, and (B) an online algorithm for finding solutions to (6.1). The learning problem of predicting $s^\star(b)$ is split into two classification problems, from parameters b to corresponding labels $(\gamma^\star, v^\star)$ for optimal cover $\mathcal{C}^\star(b)$ and binary solution $y^\star(b)$, respectively. The number of possible strategies/labels is exponential in the problem size, which would make the classification problem intractable as well. To address this issue,

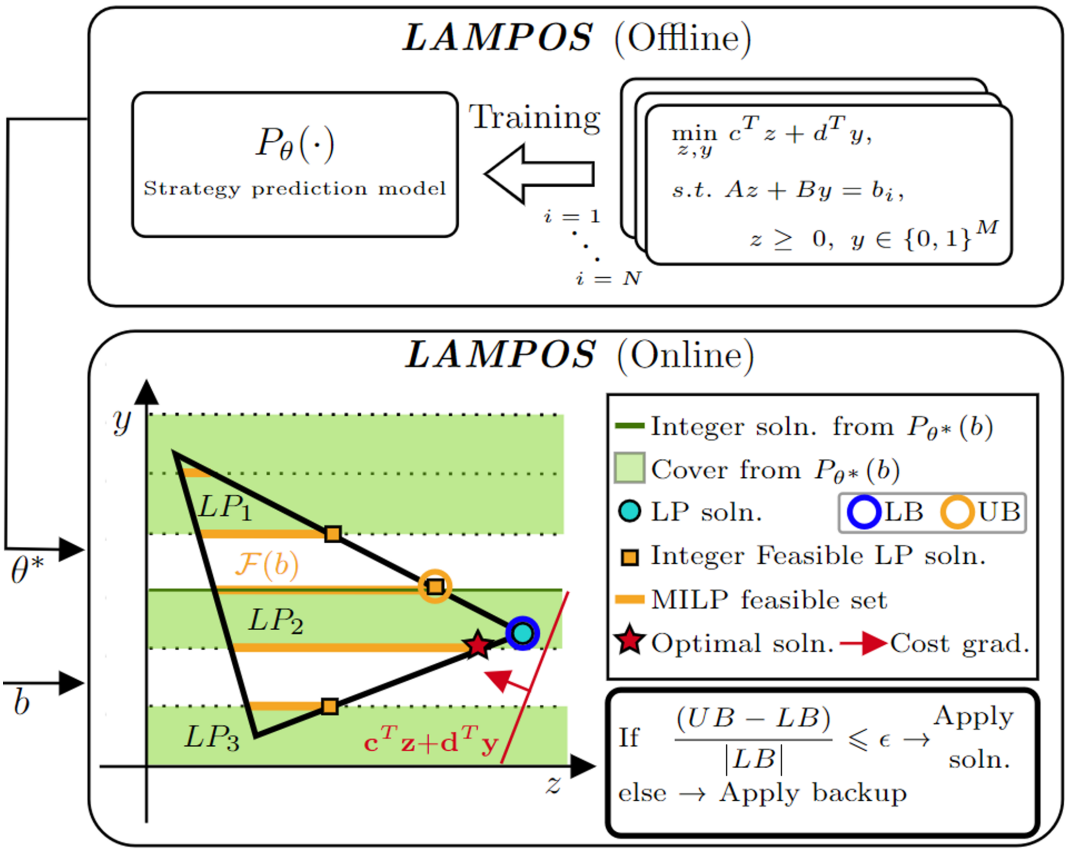


Figure 6.1: We propose LAMPOS, a strategy-based solution approach for mp-MILPs for real-time MPC. Offline, a prediction model $P_\theta(\cdot)$ is trained on various MILP instances to learn a strategy $s(\cdot)$, mapping parameters b to an optimal integer solution and a set of LPs (called a cover) obtained from the leaves of the BnB tree. Online, a solution to the MILP is obtained from the predicted strategy $s(b)$ by solving a set of LPs in parallel. The proposed strategy allows (1) sub-optimality quantification of MILP feasible solutions, and (2) recovery of MILP solution if none were found from the LPs.

we construct our dataset with a limited number of strategies using the approach in [18]. For online deployment, the predicted strategy is used to obtain solutions to the (6.1) using $R(\cdot)$, with sub-optimality quantification using $V_{\text{lb}}(\cdot)$, $V_{\text{ub}}(\cdot)$.

Offline Supervised Learning for Strategy Prediction

Dataset Construction

Our dataset consists of parameter-strategy pairs $(b_i, s(b_i))$ where the strategy $s(b_i) = (\gamma_i,\ v_i)$ consists of a tuple of labels for the cover and binary solution respectively. We

solve (6.2) for each b_i and collect: *(i)* the set of leaves of the BnB tree that constitute our optimal cover $\mathcal{C}^\star(b_i)$ and *(ii)* the binary solution $y^\star(b_i)$, and assign a label to each. Let $\mathcal{S}(\mathcal{B}_T) = \{s_1, s_2, \ldots, s_M\}$ be M strategies corresponding to T i.i.d. parameter samples $\mathcal{B}_T = \{b_1, b_2, \ldots, b_T\}$. To determine the set of labels for the supervised classification problem, we assess the probability of encountering a new strategy with a new i.i.d. sample b_{T+1}, i.e., $\mathbb{P}(s(b_{T+1}) \notin \mathcal{S}(\mathcal{B}_T))$. We adopt the Good-Turing estimator $G = T_1/T$, where T_1 represents the number of strategies that have appeared exactly once, to bound this probability with confidence at least $1 - \beta$ as

$$\mathbb{P}\left(s\left(b_{T+1}\right) \notin \mathcal{S}\left(\mathcal{B}_T\right)\right) \leq G + (2\sqrt{2} + \sqrt{3})\sqrt{\frac{1}{T}\ln\left(\frac{3}{\beta}\right)}.$$

For a fixed confidence $\beta \ll 1$, we sample parameters and update G until the right-hand side bound is less than a desired probability guarantee $\epsilon > 0$. To conclude the dataset construction, we use the insight from Theorem 6.1 to eliminate redundant strategies by searching locally around b_i in the dataset, for strategies $s(b_j)$ which maintain optimality at b_i.

Architecture and Learning problem

The classification problem for predicting the strategy can be solved using popular prediction architectures such as Deep Neural Networks (DNN) and Random Forests (RF), discussed as follows.

DNN-based Architecture

We use a feedforward DNN for cover prediction, with L layers composed together to define a function of the form $\hat{\gamma} = f_L(f_{L-1}(\ldots f_1(b)))$. The output of the lth layer is given by $y_l = f_l(y_{l-1}) = \sigma_l(W_l y_{l-1} + w_l)$ where $W_l \in \mathbb{R}^{p_l \times p_{l-1}}$ and $w_l \in \mathbb{R}^{p_l}$ are the layer's parameters, $y_0 = b$, $y_L = \hat{\gamma}$ and σ_l is the activation function used to model nonlinearities. For binary solution prediction for the MIMPC, we express $y^*(b) = [y_1^*(b), y_2^*(b), \ldots, y_N^*(b)]$ to divide the classification problem into N sub-problems, corresponding to each step along the MPC horizon N. Each sub-problem $j \in (1, 2, \ldots, N)$ consists of finding the label $v_j \in \mathrm{Y}_j$ associated with the binary solution for step j for parameter b, where Y_j is the set of labels for sub-problem j, and is predicted using a H layer DNN with parameters $W_h \in \mathbb{R}^{p_h \times p_{h-1}}$ and $w_h \in \mathbb{R}^{p_h}$ returning a label estimation $\hat{v}_j$. The label for $\hat{y}(b)$ is given by the vector of labels $\hat{v} = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_N]$. This architecture makes the classification task easier than directly recovering the full binary solution $y^*(b)$ due to the high number of different binary solutions in the dataset. Alternatively, any state-of-the-art prediction architecture [35] can be adopted for capturing the temporal dependence in the binary solution. The training process for DNN consists of finding the network parameters that minimize a loss function that encodes misclassification error. For all the classification problems, the Cross Entropy loss function is chosen, defined as

$H(p,q) = -\sum_{i=1}^{K} p_i \log(q_i)$, where p is the true label distribution over K labels and q is the predicted label distribution. The optimization problem for DNN training is solved using Stochastic Gradient Descent.

RF-based Architecture

The RF consists of multiple decision trees that are trained on random subsets of the training data, and the final prediction combines the predictions of all the decision trees in the forest by taking the majority vote. For the classification problems for binary and cover prediction, the Gini impurity criterion can be used as the splitting criterion, which measures the degree of impurity in a set of labels. The Gini impurity is defined as $\text{Gini}(p) = \sum_{i=1}^{K} p_i(1-p_i)$ where p_i is the fraction of samples in a given set that belong to class i. The Gini impurity is minimized by using a greedy approach for selecting the split of the parameter space that maximizes the reduction in impurity.

Online Deployment for MIMPC

After training the prediction models offline, the online deployment of our approach for MIMPC is described in Algorithm 3. The inputs to the algorithm are the trained strategy prediction model $P_{\theta^\star}(\cdot)$, the state of system x_t and the desired sub-optimality tolerance tol. The function `solve_MIMPC`$(\cdot)$ returns the MPC policy $\pi_{MPC}(\cdot)$. Inside it, we first query the prediction model at the current state to obtain a strategy consisting of the cover $\tilde{\mathcal{C}}(x_t)$ and a candidate binary solution $\tilde{y}(x_t)$. The list of LP sub-problems in the cover is augmented with another LP by fixing the binary variable bounds to $\tilde{y}(x_t)$. Then the LPs are solved in parallel, while keeping track of MILP feasible solutions. The solved sub-problems are sorted in the increasing order of cost, with ∞ assigned to the cost of infeasible LPs. The lower bound LB on the optimal cost is provided by the first LP sub-problem. The upper bound UB is obtained from the best MILP feasible solution, if any. If the estimated sub-optimality $\frac{UB-LB}{|LB|}$ is within tolerance, the MPC policy is obtained as $Sz^\star$ where $z^\star$ is the LP solution corresponding to the upper bound and S is a matrix that selects $u^\star_{t|t}$ from $z^\star$. If no MILP feasible solutions were found (meaning $\tilde{\mathcal{I}} = \emptyset$) or the predictions do not meet the sub-optimality tolerance, we send the sorted LP sub-problems to the backup procedure `find_sol`$(\cdot)$ which solves a sequence of MILP sub-problems. The backup returns an optimal solution if the MILP (6.1) is feasible, and nothing otherwise. A deficiency of Algorithm 3 is the requirement of a MILP solver for the backup procedure, which is unavoidable due to the $\mathcal{NP}$-hardness of (6.2) if optimal solutions are required (e.g., a stabilizing MPC policy for hybrid systems [15]). A common heuristic solution is to query the prediction model for multiple binary solution candidates [87, 34], which can be readily incorporated into Algorithm 3.

Algorithm 3: LAMPOS (Online)

Input : $P_{\theta^\star}(\cdot)$, x_t, tol
Output: $\pi_{MPC}(x_t)$
Procedure `solve_MIMPC`(x_t)**:**
  `/* Predict strategy */`
  $[\,\tilde{\mathcal{C}}(x_t) := \{\{\tilde{\text{lb}}_k, \tilde{\text{ub}}_k\}\}_{k=1}^{n_c},\ \tilde{y}(x_t)] \leftarrow P_{\theta^\star}(x_t)$
  $\tilde{\mathcal{C}}(x_t) \leftarrow \tilde{\mathcal{C}}(x_t) \cup \{(\tilde{y}(x_t), \tilde{y}(x_t))\}$
  `/* Solve LPs in parallel */`
  $\tilde{\mathcal{I}} \leftarrow \emptyset$, $\pi_{MPC}(x_t) \leftarrow \emptyset$
  parfor $k = 1$ **to** $n_c + 1$ **do**
    $(V_k, z_k, y_k) \leftarrow$solve_LP$(x_t, \tilde{\text{lb}}_k, \tilde{\text{ub}}_k)$
    if $y_k \in \{0,1\}^M$ **then**
      $\tilde{\mathcal{I}} \leftarrow \tilde{\mathcal{I}} \cup \{k\}$
  end for
  `/* Check sub-optimality */`
  $\{(\bar{V}_k, \bar{\text{lb}}_k, \bar{\text{ub}}_k)\}_{k=1}^{n_c} \leftarrow \text{sort}(\{(V_k, \tilde{\text{lb}}_k, \tilde{\text{ub}}_k)\}_{k=1}^{n_c})$
  $LB = \bar{V}_1$
  if $\tilde{\mathcal{I}} \neq \emptyset$ **then**
    $UB = \min_{k \in \tilde{\mathcal{I}}} V_k$, $z^\star \leftarrow z_{\text{argmin}_{k \in \tilde{\mathcal{I}}} V_k}$
    if $UB - LB \leq tol \cdot |LB|$ **then**
      $\pi_{MPC}(x_t) = Sz^\star$
  if $\pi_{MPC} = \emptyset$ **then**
    $(\bar{V}, \bar{z}, \bar{y}) \leftarrow$`find_sol` $(\{(\bar{V}_k, \bar{\text{lb}}_k, \bar{\text{ub}}_k)\}_{k=1}^{n_c})$
    $\pi_{MPC}(x_t) = S\bar{z}$
return $\pi_{MPC}(x_t)$
Backup `find_sol`$(\{(V_k, \text{lb}_k, \text{ub}_k)\}_{k=1}^{n_c})$**:**
  $V_{n_c+1} \leftarrow \infty$, $(\bar{V}, \bar{z}, \bar{y}) \leftarrow (\infty, \emptyset, \emptyset)$
  for $k = 1$ **to** n_c **do**
    $(\hat{V}_k, \hat{z}_k, \hat{y}_k) \leftarrow$solve_MILP$(x_t, \text{lb}_k, \text{ub}_k)$
    $(\bar{V}, \bar{z}, \bar{y}) \leftarrow$ best_sol$(\{(\hat{V}_i, \hat{z}_i, \hat{y}_i)\}_{i=1}^{k})$
    if $\bar{V} \leq V_{k+1}$ **then**
      break
  end for
return $(\bar{V}, \bar{z}, \bar{y})$

6.5 Numerical Experiments

In this section we demonstrate the effectiveness of our approach for a motion planning problem and compare the performance against MILP solvers: GLPK-MI [82], SCIP [1], Mosek [8] and Gurobi [57]. Our implementation is available at: `https://github.com/shn66/LAMPOS` (●).

MIMPC for 2D Motion planning

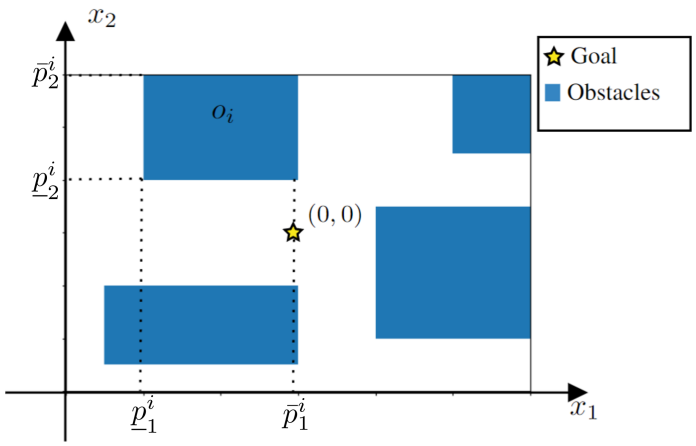


Figure 6.2: Obstacle configuration of the 2D motion planning problem, with the ith obstacle's shape: $\{(X,Y)|\ [\underline{p}_1^i, \underline{p}_2^i] \leq [X,Y] \leq [\bar{p}_1^i, \bar{p}_2^i]\}$.

The motion planning problem is to steer the robot to the origin subject to state-input and obstacle avoidance constraints as depicted in Fig. 6.2. The robot is modelled as an Euler discretized, double integrator with state $x_t = [X_t, \dot{X}_t, Y_t, \dot{Y}_t]$, control $u_t = [\ddot{X}_t, \ddot{Y}_t]$ and sampling period $dt = 0.1s$. Policy $\pi_{MPC}(x_t)$ is computed by solving the mp-MILP (6.6), which is parametric in x_t. The obstacle avoidance constraints are encoded using the big-M method, with binary vectors $\underline{\delta}_{k|t}^i, \bar{\delta}_{k|t}^i \in \{0,1\}^2$ introduced for each obstacle i at time k, totalling $4 \cdot n_{\text{obs}} \cdot N$ binary variables for a prediction horizon of N and $n_{\text{obs}} = 4$ obstacles. The vectors $\bar{x} = -\underline{x} = [3,3,2,2], \bar{u} = -\underline{u} = [2,2]$ define the state-input constraints in (6.6), and $Q = 10^3 I_4$, $R = 50 I_2$, $P = 10^5 I_4$ define the cost matrices. We model the problem using CVXPY and perform experiments for $N = 20, 40$.

$$
\begin{aligned}
\min_{\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\delta}_t} \quad & \left\| P x_{t+N|t} \right\|_\infty + \sum_{k=t}^{t+N-1} \left\| Q x_{k|t} \right\|_\infty + \left\| R u_{k|t} \right\|_\infty \\
\text{s.t.} \quad & x_{k+1|t} = I_2 \otimes \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} x_{k|t} + I_2 \otimes \begin{bmatrix} 0 \\ dt \end{bmatrix} u_{k|t}, \\
& \underline{x} \leq x_{k+1|t} \leq \bar{x},\ \underline{u} \leq u_{k|t} \leq \bar{u}, \\
& \bar{p}^i - \bar{\delta}_k^i M \leq [X_{k+1|t}, Y_{k+1|t}] \leq \underline{p}^i + M \underline{\delta}_k^i, \\
& \mathbf{1}^\top \underline{\delta}_k^i + \mathbf{1}^\top \bar{\delta}_k^i \leq 3, \\
& \underline{\delta}_k^i, \bar{\delta}_k^i \in \{0,1\}^2 \quad \forall i = 1, \ldots, n_{\text{obs}} \\
& x_{t|t} = x_t, \qquad \forall k = t, \ldots, t+N-1.
\end{aligned}
\tag{6.6}
$$

Implementation Details

Dataset construction

For dataset construction, we randomly sample parameters $b = x_t$ and solve MILP (6.6). For each b_i we collect the set of leaves of the BnB tree that represent our optimal cover $\mathcal{C}^*(b_i)$ and the binary solution $y^*(b_i)$. For eliminating redundant strategies we search within the dataset locally around b_i and look for strategies $s(b_j)$ for which the optimality is maintained for b_i. We used SCIP to save the leaves of the BnB solution tree to obtain the covers $\mathcal{C}^\star(b)$. If the leaves/LP sub-problems are unavailable for the user's solver of choice, we provide a recursive algorithm to construct a cover $\mathcal{C}^\star(b)$ in Appendix 6.7. For meeting the probability bound defined in Sec. 6.4, we fix $\beta = 10^{-3}, \epsilon = 10^{-1}$. We collected $\sim 10^5$ samples for both $N = 20, 40$ cases. After data collection, we further process the dataset by reassigning strategies with covers with a large number of LP sub-problems, to another cover with the least sub-optimality and with fewer LP sub-problems than a pre-defined threshold. This limits the online computation for solving the LP sub-problems from the cover in parallel.

Supervised learning

For strategy predictions, we use RF for the $N = 20$ case and DNN for the $N = 40$ case. For RF implementation we used the *RandomForestClassifier* from *sci-kit* setting number of trees $n_t = 10$ and used weighted tree splitting for both cover and binary solution classification to mitigate unbalanced-ness in the dataset. The RFs were trained until prediction accuracies $\geq 97\%$ are achieved for binary and cover predictions. We use Pytorch for our DNN implementation with architectures given by 2 hidden layers with width 64 for binary prediction, and 3 hidden layers with width 128 for cover prediction.

Results

We tested our approach for cases $N = 20, 40$ by sampling 100 initial conditions x_0 and solve (6.6) for the policy $\pi_{MPC}(\cdot)$ until the robot reaches the origin. For Algorithm 3, the LPs were solved using ECOS [44] (for rapid infeasibility detection) and the backup MILP sub-problems using SCIP. We compare LAMPOS against GLPK_MI, SCIP, Mosek and Gurobi for solve times. The solve times of our approach are compared to other solvers in Fig. 6.3, 6.4. Our solve times include prediction time and LP sub-problem solve times. In addition, we also solve (6.6) with SCIP, Mosek and Gurobi with a time-limit of 50ms for $N = 20, 40$ [1], and compare against LAMPOS for sub-optimality of the obtained solution (if any). For each solver, we report the average sub-optimality of feasible solutions and % of instances where it timed-out.

[1] No such interface for GLPK_MI in CVXPY

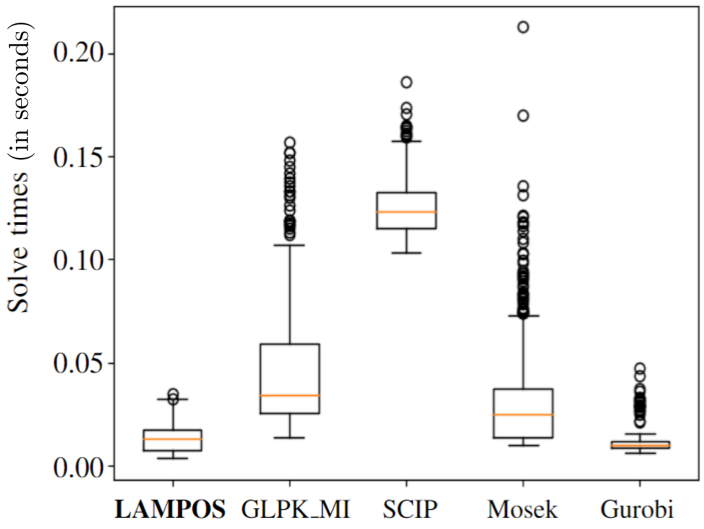


Figure 6.3: Comparison with solve times of other solvers for $N = 20$

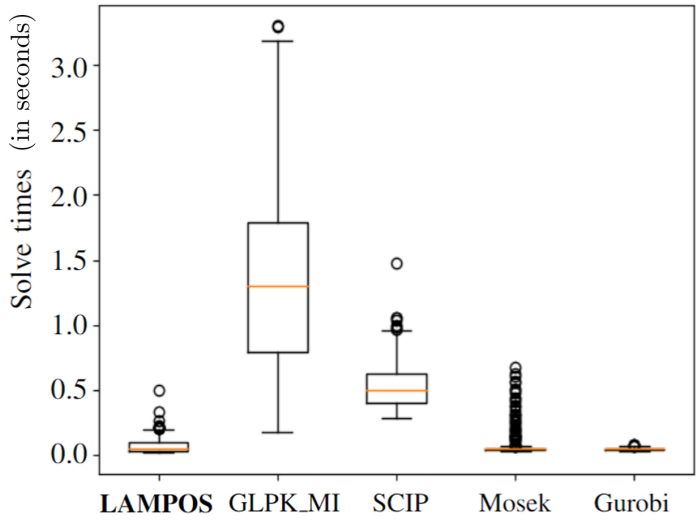


Figure 6.4: Comparison with solve times of other solvers for $N = 40$

Discussion

In Fig. 6.3, 6.4 for solve times, we see that LAMPOS outperforms open-source solvers GLPK_MI, SCIP and is comparable to Mosek, Gurobi. Table 6.1 shows that LAMPOS, Gurobi reliably find high-quality solutions within the time limit compared to SCIP, Mosek (where we use (6.3) for quantifying LAMPOS' sub-optimality). In our experiments, we observed competitive solve times for LAMPOS when $\tilde{y}(b) = y^\star(b)$, but also quick recovery otherwise by reusing the LP sub-problem information from $\tilde{\mathcal{C}}(b)$ during

Table 6.1: Performance comparison with 50ms time limit

Horizon	Metric	Solver			
		LAMPOS	SCIP	Mosek	Gurobi
N=20	Sub-opt (Avg)	**0.04**	0.34	0.16	**1e-8**
	Time-out (%)	**0**	**0**	0.2	0.8
N=40	Sub-opt (Avg)	**0.07**	-	0.2	**1e-10**
	Time-out (%)	18.6	100	22.7	**10.8**

backup calls (observed in 0.016%, 0.014% of the $N = 20, 40$ cases). To improve solve times in the future, instead of solving the LP sub-problems (6.4) explicitly, exploiting their parametric dependence on $(b, \text{lb}, \text{ub})$[143].

6.6 Conclusion

We proposed a strategy-based prediction framework to solve mp-MILPs online with sub-optimality quantification, and demonstrate it for real-time MIMPC. By exploiting the parametric nature of the optimality certificate for mp-MILPs given by the optimal set of LP sub-problems and an optimal integer solution, we observed favourable performance compared to state-of-the-art MILP solvers. For future work, we aim to train prediction models for solving the parametric LP sub-problems to further improve solve-times.

6.7 Appendix

Proof of Theorem 1

Let $\mathcal{S}_{\bar{b}}^\star \subset \mathbb{I}_1^{n_\star}$ be the set of feasible LP sub-problems in the cover $\mathcal{C}^\star(\bar{b}) = \{\{\text{lb}_k^\star, \text{ub}_k^\star\}_{k=1}^{n_\star}\}$, and let $\bar{k}$ be the optimal LP for which $y^\star(\bar{b}, \text{lb}_{\bar{k}}^\star, \text{ub}_{\bar{k}}^\star) = y^\star(\bar{b})$ and $V_{LP}^\star(\bar{b}, \text{lb}_{\bar{k}}^\star, \text{ub}_{\bar{k}}^\star) = V^\star(\bar{b})$.

For sub-problem $k \in \mathcal{S}_{\bar{b}}^\star$, we have from [21, Theorems 6.2, 6.5] that there exists a (convex) polyhedron of parameters b given by $\mathcal{K}^k = \cup_{i=1}^{p_k} \mathcal{K}_i^k \subset \mathbb{B}$ such that each $\mathcal{K}_i^k$ is polyhedral, and $(z^\star(b, \text{lb}_k, \text{ub}_k), y^\star(b, \text{lb}_k, \text{ub}_k))$ are affine functions of b for $b \in \mathcal{K}_i^k$. Define the set $\mathcal{Z}^k = \cup_{i=1}^{p_k} \{(z, y, b) \mid b \in \mathcal{K}_i^k, (z, y) = (z^\star(b, \text{lb}_k, \text{ub}_k), y^\star(b, \text{lb}_k, \text{ub}_k))\}$ for each $k \in \mathcal{S}_{\bar{b}}^\star$ and for the optimal LP $\bar{k}$, define the set $\mathcal{Z}^\star = \{(z, y, b) \mid (z, y, b) \in \mathcal{Z}^{\bar{k}},\ y = y^\star(\bar{b})\}$.

For any parameter $b \neq \bar{b}$, the solution of sub-problem $\bar{k}$ is also optimal for the MILP (6.4) at b if $V_{LP}^\star(b, \text{lb}_{\bar{k}}^\star, \text{ub}_{\bar{k}}^\star) = \min_{i \in \mathcal{S}_{\bar{b}}^\star} V_{LP}^\star(b, \text{lb}_i^\star, \text{ub}_i^\star)$ and $y^\star(b, \text{lb}_{\bar{k}}^\star, \text{ub}_{\bar{k}}^\star)) \in \{0, 1\}^M$.

Thus, the strategy $s^\star(\bar{b})$ is optimal for b if

$$\begin{aligned} &V^\star_{LP}(b, \mathrm{lb}^\star_{\bar{k}}, \mathrm{ub}^\star_{\bar{k}}) = V_{\mathrm{lb}}(b, s^\star(\bar{b})),\ y^\star(b, \mathrm{lb}^\star_{\bar{k}}, \mathrm{ub}^\star_{\bar{k}}) = y^\star(\bar{b}) \\ \Leftrightarrow\ & c^\top z^{\bar{k}} + d^\top y^{\bar{k}} \leq c^\top z^k + d^\top y^k,\ \ (z^{\bar{k}}, y^{\bar{k}}, b) \in \mathcal{Z}^\star, (z^k, y^k, b) \in \mathcal{Z}^k\ \forall k \in \mathcal{S}^\star_{\bar{b}} \backslash \{\bar{k}\}. \end{aligned}$$

Thus, the set of parameters for which $s^\star(\bar{b})$ is the optimal strategy is given by the set

$$\mathcal{P}_{\bar{b}} = \left\{ b \,\middle|\, \exists (z^{\bar{k}}, y^{\bar{k}}, b) \in \mathcal{Z}^\star,\ \exists (z^k, y^k, b) \in \mathcal{Z}^k\ \forall k \in \mathcal{S}^\star_{\bar{b}} \backslash \{\bar{k}\} : c^\top z^{\bar{k}} + d^\top y^{\bar{k}} \leq c^\top z^k + d^\top y^k \right\}$$

which is a union of convex polyhedra ($\because$ affine projection of unions of convex polyhedra $\mathcal{Z}^\star, \mathcal{Z}^k$, intersected by affine halfspaces $c^\top z^{\bar{k}} + d^\top y^{\bar{k}} \leq c^\top z^k + d^\top y^k$). ■

Cover Construction from Partial Set of Leaves

The recursive algorithm below constructs a cover $\{\{\mathrm{lb}_k, \mathrm{ub}_k\}\}_{k=1}^{n_c}$ given the optimal sub-problem $\{\mathrm{lb}^\star, \mathrm{ub}^\star\}$, and a partial list of sub-problems $\{\{\mathrm{lb}_k, \mathrm{ub}_k\}\}_{k=1}^{n_p}$ by adding disjoint *facets* $[\mathrm{lb}_i, \mathrm{ub}_i] \subset [0,1]^M$ until $\cup_{k=1}^{n_c} [\mathrm{lb}_k, \mathrm{ub}_k] \supset \{0,1\}^M$.

Algorithm 4: Recursive cover construction

Input : optimal facet: $\{\mathrm{lb}^\star, \mathrm{ub}^\star\}$, partial cover (list of facets): $\{\{\mathrm{lb}_k, \mathrm{ub}_k\}\}_{k=1}^{n_p}$
Output: Cover $\mathcal{C} = \{\{\mathrm{lb}_k, \mathrm{ub}_k\}\}_{k=1}^{n_c}$
Procedure `mk_cover`$(facet_r, facet\_list)$**:**
  `/* Loop over parallel, adjacent facets of same dimension as` $facet_r$ `*/`
  for $facet_n$ ***in*** $parallel\_facets$ **do**
    `/* Add` $facet_n$ `if not contained in nor intersected by a facet in` $facet\_list$`. Before adding, remove subsets of` $facet_n$ `from` $facet\_list$ `*/`
    if $\nexists facet'$ ***in*** $facet\_list$: $facet' \supset facet_n$ *OR* $facet' \cap facet_n \neq \emptyset$ **then**
      $facet\_list \leftarrow facet\_list \backslash \{facet' : facet' \subseteq facet_n\}$
      $facet\_list \leftarrow facet\_list \cup facet_n$
      `/* Next, call mk_cover` $(\cdot)$ `from a facet containing` $facet_r$, $facet_n$ `*/`
      $facet_r \leftarrow \mathrm{conv\_hull}(facet_r, facet_n)$
      if $facet_r \neq [0,1]^M$ **then**
        $\mathcal{C} \leftarrow$`mk_cover` $(facet_r, facet\_list)$
      break
  end for
return $\mathcal{C}$

Chapter 7

Learning Safe Supervisors for Accelerating Model Predictive Control in Interactive Environments

7.1 Overview

Robotics and autonomous agents have begun integrating into our environment. Decision-making for these agents (e.g. autonomous vehicles) in urban settings, involves motion planning in the presence of uncertain, multi-modal humans and other autonomous agents leading to the development of various solutions for planning and behavior prediction. These solutions can be categorized into three broad approaches: (i) Hierarchical Prediction and Planning [118, 38, 99]: where a sophisticated prediction architecture provides forecasts of the surrounding vehicles which is used for motion planning, (ii) Model-based Integrated Planning and Prediction [48, 148, 105]: where planning and behavior prediction are simultaneously obtained by game-theoretic and joint-optimization approaches for all vehicles, and (iii) End-to-End Learning-based Prediction and Control [78, 65]: where a sophisticated neural network, trained using imitation/reinforcement learning algorithms on realistic datasets, implicitly and jointly forecasts the behavior of surrounding vehicles and a motion plan for the autonomous vehicle. Each of these approaches suffers from either scalability for complex driving scenarios or interpretability and safety of the computed motion plans. For instance in (i), [99] employs Gaussian mixture models to explicitly express multi-modal predictions and positional uncertainty of surrounding vehicles for multi-modal motion planning. While this approach showcases robust navigation capabilities in multi-modal traffic scenarios, it focuses on interactions in simple traffic situations with limited traffic vehicles and does not scale for real-time applicability in complex scenarios with many surrounding vehicles and their multi-modal predictions. The game theoretic approaches in (ii) are generally computationally intractable for scenarios with many vehicles/agents, which is further exac-

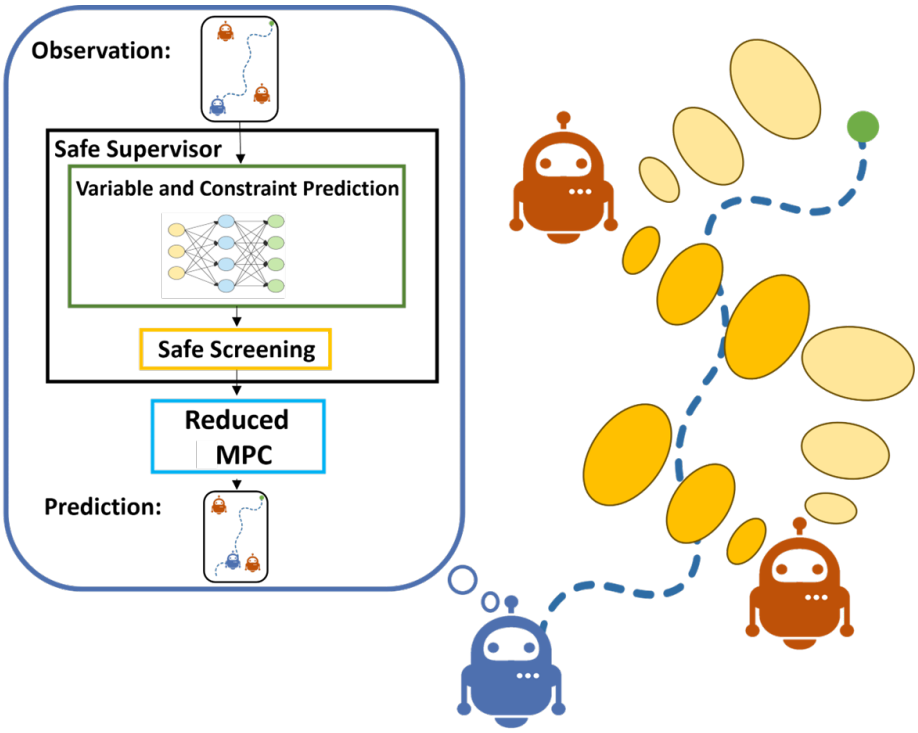


Figure 7.1: Our hierarchical architecture for motion planning with duality-based interaction prediction. Given the environment observation, we classify which *agents and their maneuvers can be eliminated/screened* for solving a reduced, real-time MPC problem.

erbated when the games are multi-modal/mixed. The End-to-End approaches in (iii) are generally scalable to complex scenarios but lack interpretability in their predictions. Given a scene with multiple agents, it is unclear how to interpret which agents and which of their possible maneuvers are relevant for a safe motion plan for the ego agent. In [78], an attention-based architecture discerns relevant vehicles at the current time-step when controlling an autonomous vehicle in a simulated traffic intersection, but does not explicitly consider safety or long-term interactions. Hence, there is a pressing need for a scalable motion planning method in complex interaction-driven scenarios involving possibly multi-modal predictions of the surrounding agents.

Contributions

- We present a duality-based interpretation of the interaction between the autonomous agent and surrounding agents to deduce which variables and constraints are relevant to MPC-based motion planning
- We design a Recurrent Neural Network (RNN) [84] with attention mechanisms

to predict the interactions between the autonomous vehicle and its surrounding vehicles, that is trained on a dataset of MPC solutions

- We present a Hierarchical MPC algorithm that uses the RNN-architecture to predict a prior on which variables and constraints are relevant to the MPC problem. Then using ideas from Lagrangian duality, we identify which constraints and variables can be *safely* eliminated, to consequently define a reduced MPC problem.

7.2 Problem Formulation

Consider a motion planning problem (e.g., 7.1) where the robot/AV aims to reach its destination while satisfying actuation and motion constraints, and avoiding multiple dynamic obstacles. In a general MPC formulation for this problem, the robot solves the following optimal control problem at time t:

$$\begin{aligned}
\min_{\{\theta_{k|t},\, \zeta_{k|t}\}_{k=t}^{t+N-1}} \quad & \sum_{k=t}^{t+N-1} \mathfrak{c}_{k|t}(x_{k|t}, u_{k|t}) && \text{(7.1a)}\\
\text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}, w_{k|t}), && \text{(7.1b)}\\
& o^i_{k+1|t} = f^i_{k|t}(o_{k|t}, \zeta_{k|t}, n_{k|t}), && \text{(7.1c)}\\
& u_{k|t} = \pi_{\theta_{k|t}}(x_{k|t}, o_{k|t}), && \text{(7.1d)}\\
& (x_{k+1|t}, u_{k|t}) \in \mathcal{X}_t \times \mathcal{U}_t, && \text{(7.1e)}\\
& \mathfrak{g}^i_{k|t}(x_{k+1|t}, o^i_{k+1|t}, \zeta^i_{k|t}) \geq 0, && \text{(7.1f)}\\
& x_{t|t} = x_t,\ o_{t|t} = o_t, && \text{(7.1g)}\\
& \forall i \in \mathbb{I}_1^{n_o},\ \forall k \in \mathbb{I}_t^{t+N-1}
\end{aligned}$$

to compute the feedback policy $\pi_t(x_t, o_t) = u^\star_{t|t}$ where

- $x_{k|t}$, $u_{k|t}$, $w_{k|t}$ denote the state, input, and process noise predictions of the robot at time step k. x_t is the measured state of the robot at time t
- $o^i_{k|t}$ denotes the predicted position of obstacle $i \in \mathbb{I}_1^{n_o}$ at time step k and the stacked predictions of all the obstacles are denoted as $o_{k|t} = [o^1_{k|t}, \ldots, o^{n_o}_{k|t}]$. o_t denotes the measured obstacle positions at time t. If the predictions are multi-modal, we define $o_{k|t} = [o_{k|t,1}, \ldots, o_{k|t,J}]$ where $o_{k|t,j}$ denotes the prediction for mode j
- $\theta_{k|t}$ are decision variables that parameterize the robot's control policy along the prediction horizon

- If the predictions are multi-modal, then $\theta_{k|t} = [\theta_{k|t,1}, \ldots, \theta_{k|t,J}]\ \forall k > t$, and similarly, define the multi-modal closed-loop trajectories $x_{k|t}, u_{k|t}$
- $\zeta_{k|t}$ are auxiliary decision variables that scale with the dimension of $o_{k|t}$, introduced for miscellaneous purposes such as for tighter collision avoidance, game-theoretic predictions or probabilistic reformulations
- (7.1a) describes the objective functional given as the sum of time-varying stage cost functionals $\mathfrak{c}_{k|t}(\cdot)$ that act on the predictions $x_{k|t}, u_{k|t}$, e.g., $\mathfrak{c}_{k|t}(x_{k|t}, u_{k|t}) = \mathbb{E}[\ ||x_{k|t} - x_k^{\text{ref}}||_2 + ||u_{k|t} - u_k^{\text{ref}}||_2\ ]$.
- (7.1b) describes the robot's discrete-time dynamics used for closed-loop trajectory predictions with the parameterized policy (7.1d)
- (7.1c) describes the prediction for obstacle i at time step $k+1$, which in general, could be a function of all the obstacles' positions at time step k, and also multi-modal. $n_{k|t}$ is the process noise for obstacle predictions.
- (7.1e) denotes polytopic state and input constraints for the robot's trajectories
- (7.1f) describe the collision avoidance constraints between the robot and obstacle i at time step k, where the functionals $\mathfrak{g}^i_{k|t}(\cdot)$ encode some geometric and probabilistic notion of safety, e.g., $\mathfrak{g}^i_{k|t}(x_{k|t}, o^i_{k|t}, \zeta^i_{k|t}) = \mathbb{P}[||x_{k+1|t} - o^i_{k+1|t}||_2 \geq d_{min}] - 1 - \zeta^i_{k|t}$ models safety when the robot and obstacle i are d_{min} apart with probability at least $1 - \zeta^i_{k|t}$
- (7.1g) incorporates the feedback from the robot and observations of the obstacles at time t

The main challenge in computing $\pi_t(\cdot)$ is the solution of the optimal control problem (1) at real-time frequencies ($\geq$ 10 Hz) in complex scenarios ($n_O \gg 1$) and over long horizons ($N \gg 1$). Although the constraints and decision variables in the optimization problem typically scale as $\mathcal{O}(N \cdot n_O)$, solving (1) via off-the-shelf solvers involves solving linear systems incurring costs of $\mathcal{O}(N \cdot n_O^3)$ (and $\mathcal{O}(N \cdot n_O^3 \cdot J^3)$ in the multi-modal case) even in the best case, which is prohibitive for low-latency, reactive robot behavior.

In this paper, we propose a hierarchical policy architecture for accelerating the computation of the policy $\pi_t(\cdot)$ using Imitation Learning (IL) and Strong Duality. The hierarchy consists of a Safe Supervisor and a reduced MPC problem solver. The *Safe Supervisor* is trained using Imitation Learning (IL) to take the current scene (AV's state, obstacle observations, semantic information) as input and output a set of constraints and variables that can be *safely* eliminated from the MPC optimization problem using ideas from Strong Duality. Subsequently, the reduced MPC problem solver is set up and solved for control computation to accelerate the solution of (1) without compromising on the feasibility of the problem.

7.3 Convexified, Group Regularized MPC Problem and Duality-based Screening

For solving the MPC problem (7.1), we adopt a Sequential Convex Programming (SCP) approach [83]. The convexified MPC can be compactly written as the following second-order cone program:

$$(P): \begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \frac{1}{2}||\mathbf{Q}_t\boldsymbol{\theta}_t||_2^2 + \mathbf{C}_t^\top\boldsymbol{\theta}_t \\ \text{s.t } & \mathbf{A}_t\boldsymbol{\theta}_t + \mathbf{R}_t \in \mathcal{K} := (\bigotimes_{s=1}^{n_c}\mathcal{K}_s) \times \mathcal{K}_{xu} \end{aligned} \tag{7.2}$$

where $n_c = \mathcal{O}(NV)$, $\mathbf{Q}_t \succ 0$ $\mathcal{K}_s$, $\forall s \in \mathbb{I}_1^{n_c}$ are the cones corresponding to the collision avoidance, and let $\mathcal{K}_{ca} = (\bigotimes_{s=1}^{n_c}\mathcal{K}_s)$. $\mathcal{K}_{xu}$ is the cone that corresponds to all the state-input constraints. Let n_θ be the number of decision variables and $S_{k|t}^{i\top}\boldsymbol{\theta}_t$ describe the policy parameters and additional variables $(\theta_{k|t}, \zeta_{k|t})$, and consider the group-regularized l_1 problem:

$$(P_{l_1}): \begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \frac{1}{2}||\mathbf{Q}_t\boldsymbol{\theta}_t||_2^2 + \mathbf{C}_t^\top\boldsymbol{\theta}_t + \lambda \sum_{k\in\mathbb{I}_{t+1}^{t+N},\, i\in\mathbb{I}_1^V} ||S_{k|t}^{i\top}\boldsymbol{\theta}_t||_1 \\ \text{s.t } & \mathbf{A}_t\boldsymbol{\theta}_t + \mathbf{R}_t \in \mathcal{K} := (\bigotimes_{s=1}^{n_c}\mathcal{K}_s) \times \mathcal{K}_{xu} \end{aligned} \tag{7.3}$$

The l_1 penalty promotes sparsity in the decision variables [32]. For example, the l_1 penalty enforces sparsity in the policy parameterization from Chapter 3. Consider the dual of (7.3),

$$(D_{l_1}): \begin{aligned} \min_{\boldsymbol{\mu}_t, \mathbf{g}_t} \quad & \frac{1}{2}||\mathbf{Q}_t^{-1}(\mathbf{A}_t^\top\begin{bmatrix}\boldsymbol{\mu}_t\\ \boldsymbol{\eta}_t\end{bmatrix} - \mathbf{C}_t - \lambda \sum_{k\in\mathbb{I}_{t+1}^{t+N},\, i\in\mathbb{I}_1^V} S_{k|t}^i g_{k|t}^i)||_2^2 \\ & + \mathbf{R}_t^\top\begin{bmatrix}\boldsymbol{\mu}_t\\ \boldsymbol{\eta}_t\end{bmatrix} \\ \text{s.t } & \boldsymbol{\mu}_t \in \bigotimes_{s=1}^{n_c}\mathcal{K}_s^*, \boldsymbol{\eta}_t \in \mathcal{K}_{xu}^*,\ ||g_{k|t}^i||_\infty \leq 1\ \forall k \in \mathbb{I}_{t+1}^{t+N}, i \in \mathbb{I}_1^{n_o} \end{aligned} \tag{7.4}$$

Notice that the feasible set of the dual problem is independent of the parameters of the primal problem. The optimal primal and dual solutions $(\boldsymbol{\theta}_t^\star, \boldsymbol{\mu}_t^\star, \boldsymbol{\eta}_t^\star, \mathbf{g}_t^\star)$ are given by the

KKT conditions:

$$\begin{aligned}
&\mathbf{Q}_t^2\boldsymbol{\theta}_t^\star + \mathbf{C}_t - \mathbf{A}_t^\top \begin{bmatrix}\boldsymbol{\mu}_t \\ \boldsymbol{\eta}_t\end{bmatrix} + \lambda \sum_{k\in\mathbb{I}_1^M, i\in\mathbb{I}_1^{n_o}} S_{k|t}^i g_{k|t}^{i\star} = 0 \\
&\mathbf{A}_t\boldsymbol{\theta}_t^\star + \mathbf{R}_t \in \mathcal{K}, \;\; \boldsymbol{\mu}_t^\star \in \bigotimes_{s=1}^{n_c} \mathcal{K}_s^*, \;\; \boldsymbol{\eta}_t^\star \in \mathcal{K}_{xu}^*, \;\; ||g_{k|t}^{i\star}||_\infty \leq 1, \\
&[\mathbf{A}_t\boldsymbol{\theta}_t^\star + \mathbf{R}_t]_{xu} \circ \boldsymbol{\eta}_t^\star = 0, \;\; [\mathbf{A}_t\boldsymbol{\theta}_t^\star + \mathbf{R}_t]_s \circ [\boldsymbol{\mu}_t^\star]_s = 0 \;\; \forall s \in \mathbb{I}_1^{n_c}, \\
&g_{k|t}^{i\star\top} S_{k|t}^{i\top} \boldsymbol{\theta}_t^\star = ||S_{k|t}^{i\top}\boldsymbol{\theta}_t^\star||_1 \;\; \forall k \in \mathbb{I}_t^{t+N}, i \in \mathbb{I}_1^{n_o}
\end{aligned} \tag{7.5}$$

Observe from the last two equations that

1. if $[\boldsymbol{\mu}_t^\star]_s = 0$, then $[\mathbf{A}_t\boldsymbol{\theta}_t^\star + \mathbf{R}_t]_s \in \text{int}(\mathcal{K}_s)$ and this constraint can be eliminated from (7.3),
2. if $||g_{k|t}^{i\star}||_\infty < 1$, then the variables corresponding to $S_{k|t}^{i\top}\boldsymbol{\theta}_t^\star$ can be eliminated from (7.3).

Let $[\mathbf{A}_t\boldsymbol{\theta} + \mathbf{R}_t]_{xu}$ be the rows corresponding to the state-input constraints. We will use ideas from sensitivity analysis of conic programs to eliminate vehicles to construct a reduced optimization problem as follows. Define the maximum possible violation across all the collision avoidance constraints for an ego trajectory that satisfies state and input constraints,

$$\begin{aligned}
\bar{D} := \max_{s,\boldsymbol{\theta}^s} \min_{\boldsymbol{z}^s} \quad & ||[\mathbf{A}_t\boldsymbol{\theta}^s + \mathbf{R}_t]_s - \boldsymbol{z}^s||_2 \\
& \text{s.t} \;\; s \in \mathbb{I}_1^{n_c}, [\mathbf{A}_t\boldsymbol{\theta}^s + \mathbf{R}_t]_{xu} \in \mathcal{K}_{xu}, \; \boldsymbol{z}^s \in \mathcal{K}_s
\end{aligned}$$

The quantity $\bar{D}$ is guaranteed to be finite because (i) the ego's trajectories and target vehicles' predictions are bounded by dynamical and actuation constraints, and (ii) the vehicle geometries are compact sets [1]. Now let $\bar{\boldsymbol{\theta}}_t^s$ be the solution of (7.3) after eliminating the sth collision avoidance constraint and define constraint violation $\boldsymbol{\delta}^s = [\mathbf{A}_t\bar{\boldsymbol{\theta}}_t^s + \mathbf{R}_t]_s - \arg\min_{\boldsymbol{z}^s\in\mathcal{K}_s} ||[\mathbf{A}_t\bar{\boldsymbol{\theta}}_t^s + \mathbf{R}_t]_s - \boldsymbol{z}^s||_2$. Let $p_t(\boldsymbol{\delta}^s)$ denote the cost of the solution $\bar{\boldsymbol{\theta}}_t^s$, and $p_t(0)$ denote the solution of the original problem (7.3). Then by using the global sensitivity result [22, Chapter 5], we have

$$p_t(0) - p_t(\boldsymbol{\delta}^s) \leq \boldsymbol{\delta}^{s\top}[\hat{\boldsymbol{\mu}}_t^\star]_s. \tag{7.6}$$

So if ϵ is an acceptable deviation in the optimal cost of (7.1) and $||[\boldsymbol{\mu}_t^\star]_s||_2 \leq \frac{\epsilon}{\bar{D}}$, then the optimal cost changes at most by ϵ. Thus, the sth collision avoidance can be ignored without significantly affecting the cost of the computed motion plan.

[1] $\bar{D}$ does not need to be explicitly computed and any $\tilde{D} > \bar{D}$ is sufficient, e.g., a conservative choice could be $\tilde{D} = W$, where W = largest dimension of drivable area.

7.4 Recurrent Attention for Interaction Duals Network (RAID-Net)

The optimal dual variables $\boldsymbol{\mu}_t^\star \in \bigotimes_{s=1}^{n_c} \mathcal{K}_s^*$ and $\boldsymbol{g}_t^\star \in \mathbb{R}^{n_\theta}$ contain information about active constraints and sparsity pattern of the optimal solution for the primal optimization problem (7.2) at time t. The continuous vector can be converted into binary vectors $\tilde{\boldsymbol{\mu}}_t^\star \in \{0,1\}^{n_c}, \tilde{\boldsymbol{g}}_t^\star \in \{0,1\}^{Nn_o}$ by applying the screening rules described in 7.3. If $[\boldsymbol{\mu}_t]_s > 0$, then the corresponding constraint is active and $[\tilde{\boldsymbol{\mu}}_t^\star]_s = 1$. Otherwise, $[\tilde{\boldsymbol{\mu}}_t^\star]_s = 0 \quad \forall s \in \mathbb{I}_1^{n_c}$. For the sparsity pattern, let $\mathcal{I}(s) = (k,i) \in \mathbb{I}_1^N \times \mathbb{I}_1^{n_o}$ be an indexing scheme for $s \in \mathbb{I}_1^{Nn_o}$. Then $[\tilde{\boldsymbol{g}}_t^\star]_s = 0$ denotes $||\tilde{g}_{k|t}^{i\star}||_\infty < 0$ and the variable $S_{k|t}^{i\top}\boldsymbol{\theta}_t^\star$ can be eliminated. Otherwise, its optimal value is non-zero.

We propose Recurrent Attention for Interaction Duals Network (RAID-Net) that predicts the dual class $[\tilde{\boldsymbol{\mu}}_t, \tilde{\boldsymbol{g}}_t^\star]$ given the observation (ob_t) of the environment. The observation may contain continuous or discrete information about the environment that correlates with the dual variables. For instance, the current state of the ego vehicle, target vehicles' positions and velocities, and semantic information such as target vehicles' behavior prediction.

Expert Dataset

With access to a simulation environment, the optimal dual solutions $\{[\boldsymbol{\mu}_t^\star, \boldsymbol{g}_t^\star]\}_{t=0}^T$ (i.e. expert actions in imitation learning nomenclature) are obtained by solving (7.2) at each step t along the trajectory with length T. Then, it is converted into the binary class labels $\{[\tilde{\boldsymbol{\mu}}_t^\star, \tilde{\boldsymbol{g}}_t^\star]\}_{t=0}^T$. The optimal dual class labels and observation of the environment pairs are stored into a dataset $\mathcal{D} = \{(ob_t, [\tilde{\boldsymbol{\mu}}_t^\star, \tilde{\boldsymbol{g}}_t^\star])\}_{t=0}^B$, where B is the user-defined dataset size.

Input Normalization and Encoding

We assume that the observation of the environment is available and includes information about the ego vehicle's states, previous control input, and target vehicles' states along with semantic information about the ego and target vehicles' modes.

The observation is normalized using environment and system information such as maximum states and inputs (restricted by the environment and system limitations), the maximum number of modes per vehicle, etc. In the context of complex interactive autonomous driving scenarios, employing graph encoding of the scene is crucial as it allows for a comprehensive representation of the intricate relationships and interactions among various entities within the scene. Nodes in the graph represent vehicles including the ego vehicle, while directed edges (originating in the ego vehicle) capture the dynamic relationships and interaction between the ego vehicle and target vehicles. We utilize an ego-centric graph encoding using time-to-collision (TTC) between the ego vehicle to the target vehicles to represent the edges. In RAID-Net the observation is separated per i-th

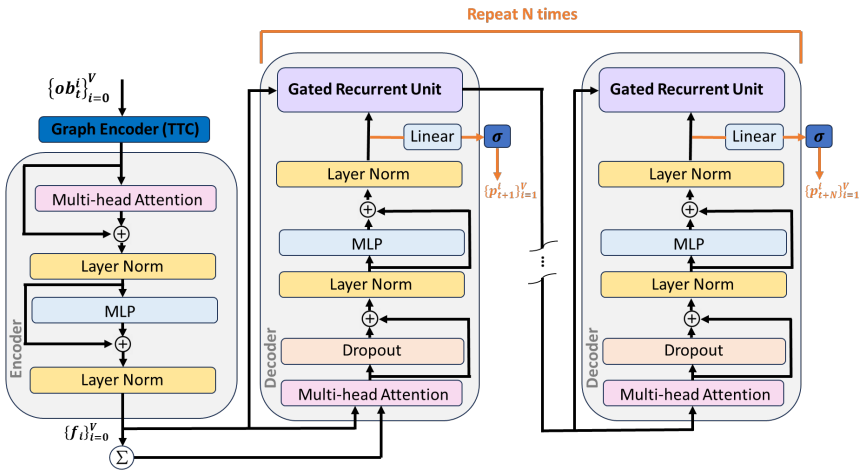


Figure 7.2: Schematic of our Recurrent Attention for Interaction Duals Network (RAID-Net) for predicting relevant constraints for the MPC optimization problem. RAID-net is invariant to the number and order of target vehicles, and has a MPC horizon-independent memory footprint because of its recurrent architecture.

vehicle: $ob_t^i \quad \forall i \in \mathbb{I}_0^{n_o}$, where $i = 0$ represents the ego vehicle and augmented with the TTC graph encoder. Subsequently, a transformer-based encoder embeds the augmented observations, producing a scene representation feature vector denoted as $f_i \in \mathbb{R}^{d_{em}}$. In this work, we define $d_{em} = 2 \times |ob|$, where $|ob|$ represents the dimension of the observation space.

The encoder exhibits permutation invariance, indicating that the network's output remains unchanged even when the order of the input is interchanged. This invariance is achieved by summing the feature vectors $\{f_i\}_{i=0}^{n_o}$ over all vehicles before feeding them into the decoder as depicted in Fig. 7.2.

Network Architecture

The normalized and encoded inputs undergo N sequential passes through an attention-based and gated recurrent network of decoders (the decoders share parameters), as illustrated in Fig. 7.2.

The decoder includes a multi-head attention mechanism to capture the different types of relationships between the encoded inputs [129]. Each head of the attention mechanism focuses on different interactions between the AV and obstacles. See [129] for detailed explanations for a multi-head attention network architecture. The multi-head attention network in the decoder features an embedding dimension of d_{em} and n_h denotes the number of heads.

The decoder also features dropout and layer normalization layers to prevent overfit-

ting and stabilize the training process [125][10]. A multi-layer perceptron (MLP) or a fully-connected neural network is included to allow the model to learn complex interactions and relationships in the data in the embedded dimension. The MLP in RAID-Net decoder is

$$z_\ell = \sigma^{\text{LeakyReLU}}(W_\ell z_{\ell-1} + b_\ell), \quad \ell = 1, ..., L \tag{7.7}$$

where W_ℓ, b_ℓ the parameters of ℓ-th layer of the network with hidden state dimension h_{MLP} where the input and output of the MLP are $z_0, z_L \in \mathbb{R}^{d_{em}\cdot(V+1)}$; $\sigma^{\text{LeakyReLU}}(x) = \max(0, x) + \alpha \min(0, x)$ where α is the negative slope parameter that controls the activation in the negative input region; L is the number of layers. The hidden states and the input to the decoder network are passed to a gated recurrent unit (GRU) which applies an input-dependent gating mechanism to modify the hidden state that will be passed on to the subsequent decoder (See Fig. 7.2). The GRU's input is identical to the corresponding decoder's input denoted as $z \in \mathbb{R}^{d_{em}}$. The number of features or hidden size of the GRU is set to $d_{em} \cdot (V + 1)$ which represents the sequence of embedded states of all vehicles including the ego that will be passed to the next recurrent decoder. The recurrent structure is popular in obstacle motion prediction as it captures the time-dependence mechanism of motion [117, 67]. Similarly, it is useful for complex interaction predictions to capture the time-dependence of the AV's interactions with the obstacles and prevents the vanishing gradient problem that is prevalent in standard recurrent neural networks [40]. Also, the recurrent structure renders the RAID-Net independent of the prediction horizon (N) as decoders share parameters. A linear projection layer ($W_p z + b_p$), defined by the parameters $W_p \in \mathbb{R}^{\frac{n_c+NV}{N}\times d_{em}\cdot(V+1)}$, $b_p \in \mathbb{R}^{\frac{n_c+NV}{N}}$ with sigmoid activation, lifts the sequence of embedded states to predict the class-1 probabilities of dual variables corresponding to the collision avoidance constraints and primal decision variables for obstacle i at time $t+1$: $\boldsymbol{p^i_{t+1}} \in \mathbb{R}^M$, where the m-th element corresponds to the class-1 probability for the m-th mode configuration of the obstacles in the scenario. Collecting the $\{\boldsymbol{p^i_{t+1}}\}_{i=1}^{n_o}$ from each decoder, we get

$$\boldsymbol{p_t} = [\{\boldsymbol{p^i_{t+1}}\}_{i=1}^{V}, ..., \{\boldsymbol{p^i_{t+N}}\}_{i=1}^{V}] \in \mathbb{R}^{n_c+n_\theta} \tag{7.8}$$

Loss function

Given the $i-$the datapoint $(\boldsymbol{p}_i, \boldsymbol{\mu}_i^\star, \boldsymbol{g}_i^\star)$, let $\boldsymbol{p}_i^\mu$, $\boldsymbol{p}_i^g$ denote the class-1 probabilities for the duals $\boldsymbol{\mu}_i$ and $\boldsymbol{g}_i$ respectively. The binary cross-entropy loss function for training the neural network to predict the dual classes is

$$\begin{aligned}\ell(\boldsymbol{p}, \tilde{\boldsymbol{\mu}}^\star, \tilde{\boldsymbol{g}}^\star) = -\tfrac{1}{n}\sum_{i=1}^{n}\Big(\sum_{s=1}^{n_c}(w_p[\boldsymbol{\mu}_i^\star]_s \log[\boldsymbol{p}_i^\mu]_s + (1-[\boldsymbol{\mu}_i^\star]_s)\log(1-[\boldsymbol{p}_i^\mu]_s) \\ + \sum_{s=1}^{Nn_o}(w_p[\boldsymbol{g}_i^\star]_s \log[\boldsymbol{p}_i^g]_s + (1-[\boldsymbol{g}_i^\star]_s)\log(1-[\boldsymbol{p}_i^g]_s)\Big),\end{aligned} \tag{7.9}$$

where $[\boldsymbol{p}_i^{\mu}]_s$, $[\tilde{\boldsymbol{\mu}}^{\star}{}_i]_s$ is the s-th component of the class-1 probability prediction and the target class label for the dual $\tilde{\boldsymbol{\mu}}$ in the i-th sample (with analogous definitions for $\tilde{\boldsymbol{g}}$). n is the training batch size and w_p is the weight corresponding to positive class. By choosing w_p appropriately, the learned classifier can conservatively over-predict class 1 and account for a potential imbalance in a dataset (i.e. number of class 0 $\gg$ number of class 1). *This is critical for safety as increasing w_p will reduce false negative classification of active constraints.*

Online Inference

For online inference, we use the RAID-Net to obtain the class-1 probabilities as $\boldsymbol{p}_t^{RAIDN}$. The selection policy for the dual classes is then given as

$$\pi^{\text{RAIDN}}(ob_t) := \begin{array}{l} \tilde{\boldsymbol{\mu}}_t = \begin{cases} [\tilde{\boldsymbol{\mu}}_{\boldsymbol{t}}]_s = 1 & \forall s \in \mathbb{I}_1^{n_c} \text{ if } [\boldsymbol{p}_t^{\mu}]_s \geq 0.5 \\ [\tilde{\boldsymbol{\mu}}_{\boldsymbol{t}}]_s = 0 & \text{otherwise} \end{cases} \\ \tilde{\boldsymbol{g}}_t = \begin{cases} [\tilde{\boldsymbol{g}}_{\boldsymbol{t}}]_s = 1 & \forall s \in \mathbb{I}_1^{Nn_o} \text{ if } [\boldsymbol{p}_t^{g}]_s \geq 0.5 \\ [\tilde{\boldsymbol{g}}_{\boldsymbol{t}}]_s = 0 & \text{otherwise} \end{cases} \end{array} \tag{7.10}$$

where $\boldsymbol{p}_t^{\mu}, \boldsymbol{p}_t^{g}$ are the RAID-Net output representing the probabilities of the corresponding dual classes being 1, given the observation of the environment at time t.

7.5 Hierarchical MPC with Safe Supervision

In this section, we describe our Hierarchical MPC architecture for accelerating the computation of the MPC solution using RAID-Net.

Safe Screening

As the dual variable class predictions from the policy $\pi^{\text{RAIDN}}(ob_t)$ may be incorrect, eliminating or screening a set of constraints and variables based on the output of a neural network carries a safety risk. Thus, we derive additional conditions for screening variables and constraints to ensure safety (i.e. constraint satisfaction for (7.3)) using strong convexity and concepts from strong duality.

Assumption 7.1 *The matrix $\mathbf{Q}_t^{-1}$ is finite and strictly positive definite.*

Denote objective of the dual problem 7.4 as $d(\boldsymbol{\mu}, \boldsymbol{g})$, and let $\underline{\sigma}, \bar{\sigma} > 0$ be the smallest and largest eigenvalues of $\mathbf{Q}_t^{-1}$ by Assumption 7.1. Then $d(\cdot)$ is $\underline{\sigma}-$strongly convex, with an $\bar{\sigma}-$Lipschitz gradient. Define the "projected gradient" for the dual as

$$\nabla^{\dagger} d(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g}) = [\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g}] - \text{proj}_{\mathcal{K}^*}([\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g}] - \nabla d(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g})),$$

for which $\nabla^\dagger d(\boldsymbol{\mu}^\star, \boldsymbol{\eta}^\star \boldsymbol{g}^\star) = 0$ at the optimal dual solution $(\boldsymbol{\mu}^\star, \boldsymbol{\eta}^\star, \boldsymbol{g}^\star)$ [16]. Then, the following inequality estimates the distance of a dual feasible $(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g})$ from optimality [134]:

$$\|(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g}) - (\boldsymbol{\mu}^\star, \boldsymbol{\eta}^\star, \boldsymbol{g}^\star)\|_2 \leq \frac{1+\bar{\sigma}}{\underline{\sigma}} ||\nabla^\dagger d(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g})||_2. \tag{7.11}$$

Notice that since $\nabla^\dagger d(\boldsymbol{\mu}^\star, \boldsymbol{\eta}^\star, \boldsymbol{g}^\star) = 0$, the bound (7.11) is tight. Define the function

$$\text{gap}(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g}) = \frac{1+\bar{\sigma}}{\underline{\sigma}} ||\nabla^\dagger d(\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{g})||_2, \tag{7.12}$$

and suppose that for some $s \in \mathbb{I}_1^{n_c}$, we have $||[\boldsymbol{\mu}_t]_s||_2 + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) \leq \frac{\epsilon}{\bar{D}}$ for some feasible duals $(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t)$. Then from (7.6),

$$\begin{aligned} p_t(0) - p_t(\boldsymbol{\delta}^s) &\leq \boldsymbol{\delta}^{s\top} [\boldsymbol{\mu}_t^\star]_s \\ &= \boldsymbol{\delta}^{s\top} ([\boldsymbol{\mu}_t^\star]_s - [\boldsymbol{\mu}_t]_s) + \boldsymbol{\delta}^{s\top} [\boldsymbol{\mu}_t]_s \\ &\leq ||\boldsymbol{\delta}^s||_2 (||[\boldsymbol{\mu}_t^\star]_s - [\boldsymbol{\mu}_t]_s||_2 + ||[\boldsymbol{\mu}_t]_s||_2) \\ &\leq \bar{D}(\text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) + ||[\boldsymbol{\mu}_t]_s||_2) \\ &\leq \epsilon \end{aligned} \tag{7.13}$$

Thus if $||[\boldsymbol{\mu}_t]_s||_2 + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) \leq \frac{\epsilon}{\bar{D}}$, the sth constraint can be *safely* screened from (7.3) without significantly affecting the optimal cost. Similarly, if $||g_{k|t}^i||_\infty + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) < 1$ some $k \in \mathbb{I}_{t+1}^{t+N}, i \in \mathbb{I}_1^{n_o}$, then we have

$$\begin{aligned} ||g_{k|t}^{i\star}||_\infty &\leq ||g_{k|t}^i||_\infty + ||g_{k|t}^{i\star} - g_{k|t}^i||_2 \\ &\leq ||g_{k|t}^i||_\infty + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) \\ &< 1 \end{aligned} \tag{7.14}$$

and consequently, the variables $S_{k|t}^i \boldsymbol{\theta}_t$ can be *safely* eliminated from (7.3).

To summarize, given a dual feasible solution $(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t)$ to (7.4), we compute the gap function $\text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t)$ and safely eliminate constraints and variables according to decision rules:

1. The conic collision avoidance constraint $[\mathbf{A}_t \boldsymbol{\theta}_t^\star + \mathbf{R}_t]_s \in \mathcal{K}_s$ can be safely eliminated from (7.3) if

$$||[\boldsymbol{\mu}_t]_s||_2 + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) \leq \frac{\epsilon}{\bar{D}} \tag{7.15}$$

2. The variables $S_{k|t}^i \boldsymbol{\theta}_t$ comprising the policy parameters and exogenous variables for obstacle i at time k can be safely eliminated from (7.3) if

$$||g_{k|t}^i||_\infty + \text{gap}(\boldsymbol{\mu}_t, \boldsymbol{\eta}_t, \boldsymbol{g}_t) < 1 \tag{7.16}$$

Hierarchical Algorithm

For online deployment, we use the RAID-Net in a hierarchical structure with the MPC planner as formalized in Algorithm 5. Given the dual class-1 probabilities, we first recover a dual feasible solution $(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{g}}_t)$ efficiently. Using (7.10), define the following sets at time t that contain all the indices of constraints that are predicted to be active, and all variables that are predicted to have non-zero values:

$$\begin{aligned} \hat{\mathbb{S}}^{\mu} &:= \{s \in \mathbb{I}_1^{n_c} |\ [\tilde{\boldsymbol{\mu}}_t]_s = 1\} \\ \hat{\mathbb{S}}^{\theta} &:= \{(k+t, i) \in \mathbb{I}_{t+1}^{t+N} \times \mathbb{I}_1^{n_c} |\ (k,i) = \mathcal{I}(s), [\tilde{\boldsymbol{g}}_t]_s = 1\} \end{aligned} \tag{7.17}$$

We obtain a feasible dual $\hat{\boldsymbol{g}}_t$ by setting $\hat{g}^i_{k|t} = \mathbf{1}_{\frac{n_\theta}{NV}}$ for all $(k,i) \in \hat{\mathbb{S}}^{\theta}$ and $\mathbf{0}_{\frac{n_\theta}{NV}}$ otherwise. For the remaining dual variables we solve the reduced linear system via least-squares,

$$[\mathbf{A}_t]_{\hat{\mathbb{S}}^{\mu} \cup xu} \mathbf{Q}_t^{-1} [\mathbf{A}_t]^{\top}_{\hat{\mathbb{S}}^{\mu} \cup xu} \begin{bmatrix} [\bar{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}^{\mu}} \\ \bar{\boldsymbol{\eta}}_t \end{bmatrix} = [\mathbf{A}_t]_{\hat{\mathbb{S}}^{\mu} \cup xu} \mathbf{Q}_t^{-1} (\mathbf{C}_t + \lambda \sum_{(k,i) \in \hat{\mathbb{S}}^{\theta}} \mathbf{1}) - [\mathbf{R}_t]_{\hat{\mathbb{S}}^{\mu} \cup xu} \tag{7.18}$$

to obtain the unconstrained minimizer of the *reduced* dual problem (by eliminating all duals $\boldsymbol{\mu}_t$ not in $\hat{\mathbb{S}}^{\mu}$ and fixing $\boldsymbol{g}_t = \bar{\boldsymbol{g}}_t$). Then we project $\bar{\boldsymbol{\eta}}_t$ onto $\mathcal{K}^*_{xu}$ and project $[\bar{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}^{\mu}}$ onto the corresponding cones to get $[\hat{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}^{\mu}}$ and set the other elements of $\hat{\boldsymbol{\mu}}_t$ to be 0.

Finally we compute the $\text{gap}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{g}}_t)$ as in (7.12) using the dual feasible candidate $(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{g}}_t)$ and define the reduce set of variables and constraints via the rules (7.15), (7.16):

$$\begin{aligned} \mathbb{S}^R &:= \{s | \forall s \in \mathbb{I}_1^{n_c} : ||[\hat{\boldsymbol{\mu}}_t]_s||_2 + \text{gap}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{g}}_t) > \frac{\epsilon}{\bar{D}}\} \\ \boldsymbol{\theta}_t^R &:= \begin{cases} 0 & \forall (k,i) \in \mathbb{I}_{t+1}^{t+N} \times \mathbb{I}_1^{n_o} :\ ||\hat{g}^i_{k|t}||_\infty + \text{gap}(\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\eta}}_t, \hat{\boldsymbol{g}}_t) < 1 \\ S^i_{k|t} \boldsymbol{\theta}_t & \text{otherwise} \end{cases} \end{aligned} \tag{7.19}$$

Thus, we can now define the reduced optimization problem

$$\begin{aligned} (P_r): \quad & \min_{\boldsymbol{\theta}_t^R} \quad \frac{1}{2} ||\mathbf{Q}_t \boldsymbol{\theta}_t^R||_2^2 + \mathbf{C}_t^{\top} \boldsymbol{\theta}_t^R \\ & \text{s.t } [\mathbf{A}_t \boldsymbol{\theta}_t^R + \mathbf{R}_t]_{\mathbb{S}^R \cup xu} \in (\bigotimes_{s \in \mathbb{S}^R} \mathcal{K}_s) \times \mathcal{K}_{xu} \end{aligned} \tag{7.20}$$

The Reduced MPC (ReMPC) is given by the optimal solution of (7.20) as

$$u_t = \pi_{\text{ReMPC}}(\mathbf{A}_t, \mathbf{R}_t, \mathbf{Q}_t, \mathbf{C}_t) = u^{\star}_{t|t} \tag{7.21}$$

where $\mathbf{A}_t$ and $\mathbf{R}_t$ represent the initial conditions, state and input constraints, respectively. We assume ob_t is available (from the simulation environment or on-vehicle sensory devices) and contains the necessary information to construct $\mathbf{A}_t, \mathbf{R}_t, \mathbf{Q}_t, \mathbf{C}_t$.

Theorem 7.1 *Let Assumption 7.1 hold and $\boldsymbol{\theta}_t^{R\star}$ be the optimal solution reduced MPC problem at time t computed using Algorithm 5. Then if $\boldsymbol{\theta}_t^{R\star}$ is feasible for the original MPC problem* (7.2)*, then it is the optimal solution, i.e., $\boldsymbol{\theta}_t^{R\star} = \boldsymbol{\theta}_t^{\star}$. Otherwise it is at most $\sqrt{\frac{2n_c\epsilon}{\bar{\sigma}}}$ away from the optimal solution.*

Algorithm 5: Hierarchical MPC (HMPC): Hierarchical Motion Planning with Duality-based Safe Supervision

Require: π^{RAIDN}

Set the task horizon T
while $t < T$ **do**
  $ob_t \leftarrow$ Observation from the environment
  $\mathbf{A}_t, \mathbf{R}_t, \mathbf{Q}_t, \mathbf{C}_t \leftarrow ob_t$ {Constructed from observation}
  $\hat{\mathbb{S}}^{\mu}, \hat{\mathbb{S}}^{\theta} \leftarrow$ Computed using (7.17) from $\pi^{\text{RAIDN}}(ob_t)$ (7.10)
  $\mathbb{S}^R, \boldsymbol{\theta}_t^R \leftarrow$ Reduced variables and constraints set using screening criterion (7.19)
  $u_t \leftarrow \pi_{\text{ReMPC}}(\mathbf{A}_t, \mathbf{R}_t, \mathbf{Q}_t, \mathbf{C}_t)$
  Apply u_t to system
  $t \leftarrow t + 1$
end while

7.6 Example: Planning at a Traffic Intersection

Simulation Environment

We created a customized Python simulation environment for unsignalized traffic intersection based on OpenAI gymnasium [24]. The simulation environment has four node zones: *N, S, W, and E* as shown in Fig. 7.3. From each node zone, vehicles are randomly generated in the following order: Firstly, the ego vehicle is placed at the starting node in *W* zone, starting with a random target node (*N* or *E* defining 2 modes for ego vehicle). Note that the *S* node isn't considered a target for the vehicle starting in *W* zone as it leads to minimal interaction between vehicles. Next, V target vehicles are randomly spawned. V is a random integer from the interval [1,V_{max}], where V_{max} is the maximum possible number of target vehicles defined by the user. In this work, we arbitrarily set $V_{max} = 3$.

Each of these target vehicles originates from distinct zones (*W, S, and E*) and is assigned a randomly determined target node based on the available modes outlined in Table 7.1. The target vehicle spawning from the *E* zone, serving as cross-traffic for the ego vehicle, is provided with four modes for diverse interactive scenarios, while target vehicles starting from other zones have two modes each. The total possible mode configuration of the target vehicles in the scenario is then $M = 16$, which leads to $13 \cdot 16 \cdot 3 = 624$ collision avoidance constraints for (7.2) when $N = 14$.

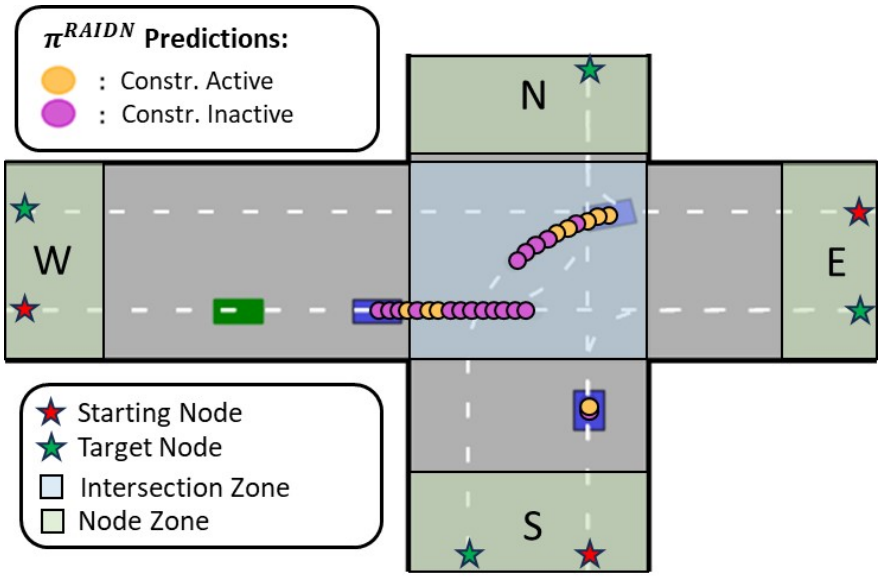


Figure 7.3: An example scene in the custom unsignalized intersection environment. The ego vehicle (green rectangle) is approaching the intersection and interacts with the target vehicles (blue rectangles).The active and inactive constraint predictions from the π^{RAIDN} are depicted as yellow and magenta ellipses, respectively.

Given the starting and target node, each vehicle follows a fixed reference trajectory marked with white dashed lines in Fig. 7.3. The vehicle states are $x = [s, v] \in \mathbb{R}^2$, where s and v are displacement along the reference trajectory and velocity, respectively. The target vehicle starting from the W node begins 8 meters ahead of the ego vehicle, moving at an initial velocity of 8 m/s. Target vehicles from the E zone start with an initial velocity of 8 m/s (7 if slow), while those from the S zone begin with 7 m/s.

Table 7.1: Simulator Vehicle's Modes

Start Node	Target Node	# Modes
W	E, N	2
S	N, E	2
E	W, W (slow), S, N	4

The target vehicle resets to its starting node upon reaching its target node, ending the simulation episode if a collision occurs or when the ego vehicle reaches its target node. Target vehicles employ a Frenet-frame Intelligent Driver Model (fIDM) for safe and human-like acceleration, considering virtual projections of surrounding vehicles on the controlled vehicle's reference trajectory, akin to the generalized Intelligent Driver Model in [76]. The fIDM also features logic gates that prioritize vehicles within intersections,

emulating human-like driving behavior. For instance, the approaching vehicle will yield to the surrounding vehicle that is already in the interaction zone as shown in Fig. 7.3. After computing acceleration inputs for the ego vehicle and target vehicles using the relevant motion planner and fIDM, they are simulated forward by a time step (Δt) along their respective reference trajectories using the kinematic bicycle model.

The observation vector available to the ego vehicle from the simulator is represented as $ob_t = [x_t, u_{t-1}, mode_{ev}, \{o_t^i\}_{i=1}^{V_{max}}, \{mode_i\}_{i=1}^{V_{max}}, \{TTC_i\}_{i=0}^{V_{max}}] \in \mathbb{R}^{17}$, where x_t, u_{t-1}, and $mode_{ev}$ denote the ego states, control input at previous time step, and EV's mode, respectively. o_t^i represents the state of the i-th target vehicle, $mode_i$ denotes its mode, and TTC_i denotes the time-to-collision between the ego vehicle and the i-th target vehicle, with $TTC_0 = 0$. When $V < V_{max}$, we spawn dummy vehicles outside the traffic scene for the rest.

RAID-Net Imitation Learning

We use a behavior cloning algorithm [91] to train the RAID-Net to mimic the expert (optimal dual solutions to 7.2). Data is collected by rolling out N_T trajectories in the aforementioned simulator using the expert policy (7.2) with prediction horizon $N = 14$ and sampling time $\Delta t = 0.2s$. Then, $(ob_t, \tilde{\boldsymbol{\mu}}_t^\star)$ pairs at each time step t are recorded. During the data collection phase, the optimization problem is formulated in CasADi[7] and solved using IPOPT[28] to extract optimal dual solutions. Repeating the process initialized with randomly generated scenarios, 120,315 data points are collected. We decompose the dataset into training (85%) and test (15%) datasets.

In dual-class classification for predicting active constraints, minimizing false negatives is crucial for safety. While low precision is acceptable, high recall is desired, which represents the model's ability to find all positive classes. To counter dataset imbalance and introduce a positive-class bias in the model, we employ $w_p = 60$ in (7.9) during model training. Additionally, we address dataset imbalance using a weighted random sampler in PyTorch during training, assigning a higher weight to the under-represented class to achieve balanced class representation in the training batch. The RAID-Net is

Table 7.2: RAID-Net model parameters

α	p_{dropout}	h_{MLP}	L	n_h
-0.1	0.1	128	6	1

constructed using PyTorch and trained using the training dataset and the loss function (7.9). The parameters used to construct the RAID-Net are reported in Table 7.2. We train the RAID-Net for 3000 epochs using a batch size of 1024 and the Adam with a constant

learning rate of 0.001 and $\beta_1, \beta_2 = (0.9, 0.99)$ [79]. The training process takes in total of 5 hours. [2]

7.7 Results

In this section, we first present the imitation learning results of the RAID-Net. Secondly, we compare the performance of our proposed **HMPC** policy 5 to the full MPC policy (7.2) ($N = 14, \Delta t = 0.2$ s for both policies) across 100 randomly generated scenarios in numerical simulations [2]. During the evaluation of HMPC and full MPC policies, the optimization problems are formulated in CasADi and the MPC SOCPs are solved using Gurobi [57].

RAID-Net Evaluation

The trained RAID-Net model was evaluated on the test dataset. We compare the normalized loss statistics of RAID-Net to an MLP NN (π^{MLP}) trained with the configuration as RAID-Net and following parameters: $h_{\text{MLP}} = 128, L = 6$ in Fig. 7.4a. We report the normalized confusion matrix of RAID-Net in Fig. 7.4b. The RAID-Net model adeptly captures intricate interactions between the ego vehicle and target vehicles, surpassing the capabilities of an MLP NN. Further, the RAID-Net model, utilizing dual-class classification, achieved a recall of 0.95 and a precision of 0.44. Evaluated on the test dataset, it attained 98.1% correct predictions of dual classes with a low 0.055 false negative rate. This performance showcases the proposed architecture's effectiveness in capturing complex interactions between the ego vehicle and target vehicles with multi-modal predictions.

Policy Comparison

The performance metrics over 100 runs are recorded in Table 7.3. We record % of time steps where MPC infeasibility and collision with a target vehicle were detected, along with average % of constraints that were imposed in the MPC. Furthermore, average times for solving (7.2) and (7.20) as well as the average times for querying the RAID-Net. Finally, the average task completion (reaching the target node) time for **HMPC** normalized by that of the full MPC is reported. Fig. 7.3 presents a snapshot of an example scene in the custom simulation environment, visualizing the π^{RAIDN} predictions. The **HMPC** algorithm imposes only the collision avoidance constraints corresponding to the yellow multi-modal prediction of the target vehicle. A video demonstrating the **HMPC** algorithm in multiple scenarios can be found here: https://youtu.be/-TcMeolCLWc

[2] Training and experiments were run on a computer with a Intel i9-9900K CPU, 32 GB RAM, and a RTX 2080 Ti GPU.

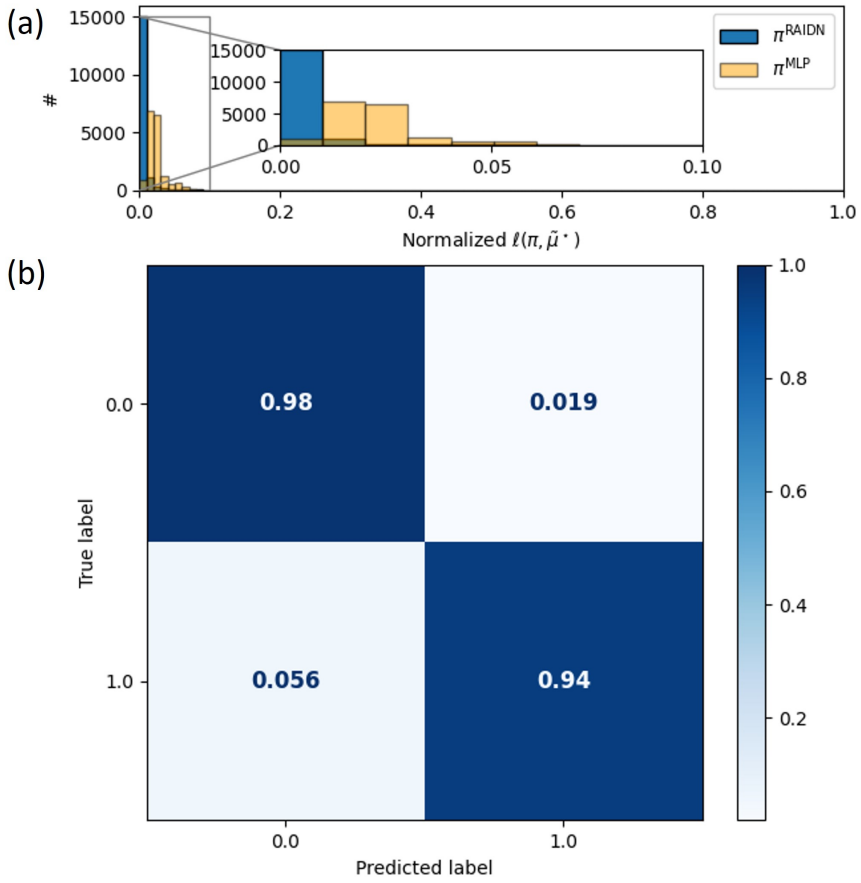


Figure 7.4: RAID-Net evaluation results on a test dataset: (a) histogram of the normalized loss value of RAID-Net versus MLP neural network, (b) normalized confusion matrix of RAID-Net

Table 7.3: Performance metrics across all policies for 7.7.

Performance metric	Full MPC (7.2)	HMPC (5)
Feasibility (%)	98.97	**99.79**
Collision (%)	**0**	4.0
Avg. Constraints Enforced (%)	100	**17.45**
Avg. Solve Time (s)	0.92 ± 0.18	$\mathbf{0.063 \pm 0.073}$
Avg. RAID-Net Query Time (s)	–	$\mathbf{0.013 \pm 0.003}$
Avg. Total Computation Time (s)	0.92 ± 0.18	$\mathbf{0.076 \pm 0.076}$
Avg. normalized task completion time (by that of Full MPC)	1	**0.91**

Discussion

On average, the solve time including the RAID-Net query time for the **HMPC** algorithm is **12 times faster** than that of the full MPC as highlighted in the 6-th row of Table

7.3. This significant acceleration in solving time is attributed to the reduced number of constraints using our proposed algorithm. The **HMPC** algorithm enables real-time applications at frequencies $\geq 10Hz$ by predicting interactions and selectively imposing only safety-critical multi-modal collision avoidance constraints with multiple target vehicles, which is not feasible in real-time otherwise. Additionally, **HMPC** algorithm had slightly higher average feasibility % which is also attributed to imposing fewer constraints. Considering that only 1.52% of the total constraints were active in the test dataset, a prediction of 17.45% for active constraints is a conservative estimation that was deliberately achieved for safety. The **HMPC** algorithm achieved a 0.91 normalized task completion time, indicating a 9% faster task completion compared to control with the full MPC. The **HMPC**-ego vehicle completes the task faster as it is optimistic, imposing fewer constraints, while the full MPC-ego vehicle exhibits a more conservative behavior.

We note that out of 100 runs, the ego vehicle collided with a target vehicle 4 times when controlled using the **HMPC** algorithm compared to 0 times using the full MPC. Collisions occurred due to false negative classification of dual variables leading the **HMPC** algorithm to screen out incorrect constraints.

The results highlight the advantages of screening inactive constraints from the MPC formulation for motion planning in highly interactive traffic scenarios to accelerate the computation of MPC. Numerical simulations revealed that, in practice, very few constraints are active in MPC formulation for motion planning with multi-modal collision avoidance constraints. With the proposed RAID-Net architecture, the model can learn to predict which target vehicles, and which modes along the prediction horizon interact with the ego vehicle given the graph encoding of the scene. Thus by imposing only the relevant, interactive constraints, we achieve over 12 times improvement in solution time while ensuring safety with a high probability.

7.8 Conclusion

We proposed a hierarchical architecture for scalable real-time MPC for complex, multi-modal traffic scenarios, consisting of 1) RAID-Net, an attention-based recurrent NN architecture that predicts dual variable candidates for the MPC problem for identifying relevant ego-target vehicle interactions along the MPC horizon, 2) A screening mechanism based on strong duality which determines which variables and constraints can be safely eliminated, and 3) a reduced Stochastic MPC problem solved without the screened constraints and variables for computational efficiency. We demonstrate our approach at a simulated traffic intersection with interactive vehicles and achieve a 12x speed-up in MPC solve times.

7.9 Appendix

Proof of Theorem 7.1

Suppose that $\boldsymbol{\theta}_t^{R\star}$ is feasible for the original MPC problem (7.2). From (7.14), it is clear that $\boldsymbol{\theta}_t^{R\star}$ can not be more sparse than $\boldsymbol{\theta}_t^{\star}$. Moreover, since (7.20) has fewer constraints than (7.2), the feasible set of (7.2) is *contained* within the feasible set of (7.20). Thus, the optimal solution of (7.20) is less than the cost of (7.2). But $\boldsymbol{\theta}_t^{R\star}$ is feasible for (7.2), and so it must be optimal by convexity of (7.2).

Now suppose that $\boldsymbol{\theta}_t^{R\star}$ is not feasible for (7.2) and let some m constraints out of the n_c conic constraints from the original problem be violated. Let $p_t(\delta^{i_1}, \ldots, \delta^{i_m})$ be the cost of the reduced problem where $\delta^{i_1}, \ldots, \delta^{i_m}$ are the m constraint violations. Then from (7.13), the optimal cost of reduced MPC problem is at most $m\epsilon$ less than than the cost of $\boldsymbol{\theta}_t^{\star}$. From Assumption 7.1 and the duality between Lipshitz gradients and strong convexity [69], we have the cost of (7.2) is $\bar{\sigma}-$strongly convex, for which we have

$$\frac{\bar{\sigma}}{2}||\boldsymbol{\theta}_t^{R\star} - \boldsymbol{\theta}_t^{\star}||^2 \leq p_t(0) - p_t(\delta^{i_1}, \ldots, \delta^{i_m}) \leq m\epsilon \leq n_c\epsilon$$

$$\implies ||\boldsymbol{\theta}_t^{R\star} - \boldsymbol{\theta}_t^{\star}|| \leq \sqrt{\frac{2n_c\epsilon}{\bar{\sigma}}}$$

■

Chapter 8

Conclusion and Future Directions

Opportunities abound in developing computational tools and algorithms towards enabling heterogeneous, collaborative robots to solve real-world problems at the intersection of economic and environmental sustainability. Sectors such as agriculture, manufacturing, aerospace operations, logistics, and transportation offer exciting opportunities in this realm for robotics and autonomy solutions, for example, towards distributed analytics and efficient dispatch. To develop solutions in these areas, our robots require algorithms that equip them with the following capabilities:

(C1) *reliability and adaptability in complex, ever-changing environments*,

(C2) *precision and high performance in task execution despite uncertainty*,

(C3) *fast decision-making for task-level planning and low-level execution*.

In this thesis, we developed data-driven MPC algorithms for autonomous vehicles towards enabling them with the capabilities (C1), (C2) and (C3). The proposed approaches have been validated to be robust, performant adn efficient, both in theory and in practice via hardware and computational experiments. We summarize the contributions of this thesis as follows.

Robustness

Robots operating autonomously in the world must navigate while interacting with the environment and the other agents. For example, autonomous vehicles in complex traffic scenarios need to plan their maneuvers while perceiving the behavior of the vehicles around them. While doing so, it must also be robust to changes and uncertainty in its own behavior. Towards enabling (C1), we have developed (i) set-membership techniques for using limited system trajectory data to quantify model mismatch for robust control within the specified operational constraints, and (ii) MPC algorithms that incorporate multi-modal predictions of the environment.

In [96, 27], we develop convex programming-based, non-parametric estimation techniques to quantify the system's model mismatch and adapt online for robust MPC. Statistically estimated system constants (such as the Lipschitz constant) are used to design efficient set-based representations of the mismatch as a function of the trajectory data, which enables set-membership-based estimation using convex optimization. Contrary to prior works on non-parametric estimation of model mismatch, our approach can efficiently estimate model mismatch over reachable sets of the robot, and with minimal assumptions and hyper-parameter tuning.

In [101, 99], we developed Stochastic MPC algorithms for an autonomous vehicle to track its desired route, operate within constraints and avoid collisions. To accomplish this, we train models on open-source traffic datasets [29] to predict the maneuvers of the vehicles as multi-modal probability distributions, to capture variability in discrete maneuvers (e.g., yield or cross) as well as continuous trajectory realizations (e.g., vehicle speed, turning radius). The main technical contributions in this project focus efficiently addressing the multi-modal predictions for interacting with the environment. To address the uncertainty arising from the multi-modal predictions, the articles develop SMPC formulations that optimize over multi-modal policy parameterizations (as opposed to open-loop sequences). This enhances the feasibility of the SMPC optimization problem by allowing the SMPC's optimal solution to adapt to different realizations of the surrounding vehicles' multi-modal predictions. The main ideas of this project were validated via hardware experiments on a full-scale vehicle for autonomous lane change in the presence of virtual vehicles with bi-modal predictions.

Performance

Most of the popular robotic systems in use today are modeled as nonlinear dynamical systems, be it manipulators, drones, or self-driving cars, and all of which are required to deliver high performance in tightly constrained, uncertain environments. Towards (C2), we develop (i) an efficient value function approximation for optimal control for nonlinear systems to efficiently improve performance from data, and (ii) an optimization-based formulation for collision avoidance in tightly constrained, uncertain environments.

In [97, 95], we iteratively learn convex control invariant sets and value function approximations for constrained optimal control of *flat* nonlinear systems (e.g. kinematic bicycles, quadrotors, DC motors), using historical trajectory data. Our constructions capture global properties of the nonlinear system via the trajectory data, in contrast to local, linearization-based approaches in the literature. The key idea is to exploit the linearity of the *time-shift* dynamics of flat nonlinear systems in the flat output trajectory space, for convex interpolation of trajectory costs within recorded trajectories, akin to Learning MPC [113]. The culmination of this project is the algorithm, "Robust Output-lifted Learning MPC", which was successfully demonstrated for autonomous racing using a 1/10 scale vehicle for robust constraint satisfaction, and iterative lap-time improvement.

In [98], we actively decrease the effect of the uncertainty for collision avoidance by proposing optimization-based collision avoidance formulations that avoid conservative over-approximations of the vehicles' geometry. The key idea is to use the dualized perspective of collision avoidance for efficient collision checking using convex optimization. We propose collision avoidance formulations for various uncertainty characterizations to highlight the general applicability of the approach. Moreover, the approach improves over the prevailing approaches for realistic driving scenarios against a metric that quantifies mobility, comfort, conservatism, and computational efficiency.

Efficiency

Robots need to be sufficiently reactive and make quick decisions to accomplish tasks in dynamic and uncertain environments. MPC offers a unifying framework for low-level feedback control and high-level planning but requires the solution of moderately-sized optimization problems at high frequency ($\geq$ 10 Hz) for real-time control. This is challenging for accomplishing (C3) when the environment involves multiple agents interacting with the robot, or the planning requires combinatorial decision-making.

In [116], we use supervised learning to accelerate the computation of MPC policies for real-time control towards enabling (C3). We developed a supervised learning framework for fast solution of multi-parametric Mixed Integer Linear Programs (mp-MILPs) to predict parametric strategies for fast solution computation, along with sub-optimality certificates. The sub-optimality certificates provide *a priori* quantification of the predictions' quality before applying the computed control to ensure safety so that (C1) is not compromised. The approach also shows favorable performance compared to state-of-the-art MILP solvers (Gurobi, Mosek, SCIP, GLPK) for real-time motion planning using mixed-integer MPC.

Finally in Chapter 7, [72], we develop a hierarchical approach towards accelerating MPC algorithms for motion planning involving complex interactions. The idea is to collect a dataset of sparse solutions to the SMPC problem, and train a transformer-based neural network architecture using imitation learning to prescribe which constraints and decision variables can be eliminated before solving the MPC optimization problem. For ensuring safety online, we robustify the prescriptions using Lagrangian duality so that critical constraints and variables are kept in the MPC formulation. Compared to the prevailing approaches, we demonstrate considerable improvement in computational performance without sacrificing safety.

Future directions

Scaling optimization-based decision-making and control for deployment on robots, and developing algorithms for large-scale, collaborative robotics, are promising directions to build on the ideas of this dissertation towards enabling (C1), (C2) and (C3).

1) **Automated Tuning, Evaluation, and Generalization**
Taking lessons from the successes of modern deep learning, it would be worthwhile to investigate differentiable programming frameworks for automated tuning and evaluation of optimization-based control algorithms (such as those developed in Chapters 2-5) with *limited real-world robot data*. This is crucial for the deployment of reliable optimization-based control algorithms at scale. Additionally, algorithms for out-of-distribution detection, generalization, and adaptation will help us close the control design loop by enabling the collection of safe, real-world data.

2) **Distributed Algorithms for Interactive Decision-making**
The research in Chapters 3, 5 assumes that a single robot bears the computational load of decision-making, which presents a challenge for real-time control in large-scale, interactive scenarios. By exploiting communication between agents for large-scale applications, robots can collaborate by distributing the computation to mitigate the issue of scale and decrease prediction uncertainty in heterogeneous teams via interaction. In this direction, developing algorithms for computing distributed, closed-loop Nash equilibria in the presence of uncertainty for motion planning with heterogeneous robot teams would be a promising avenue.

3) **Hierarchical Algorithms for Large-scale Planning and Control**
Modern decision-making pipelines in robotics [11, 137, 47] feature a hierarchical approach: first, compute coarse high-level plans for completing the task, and then synthesize low-level control for executing the plans. This modular approach decouples the decision-making problem to enable tractable, high-frequency decision-making but at the price of sub-optimal solutions, because the interplay between modules is often not modeled explicitly. A bilevel/Stackelberg game approach for the co-design of hierarchical planners would be a promising direction to investigate to address this shortcoming. The bilevel optimization problem can be reformulated into a tractable single-level problem by approximating the value function of the nested optimization problem using the techniques developed in [97, 113],4. Additionally, there are many open challenges towards building on Chapters 6, 7 to develop safe, data-driven models for solving mixed-integer programs prevalent in large-scale decision-making.

Bibliography

[1] Tobias Achterberg. "SCIP: solving constraint integer programs". In: *Mathematical Programming Computation* 1 (2009).

[2] National Highway Traffic Safety Administration. "Automated Vehicles for Safety". In: *https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety* (2020).

[3] National Highway Traffic Safety Administration. "Crash factors in intersection-related crashes: An on-scene perspective". In: *http://www-nrd.nhtsa.dot.gov/Pubs/811366.pdf* (2010).

[4] Devansh R Agrawal et al. "A Constructive Method for Designing Safe Multirate Controllers for Differentially-Flat Systems". In: *IEEE Control Systems Letters* 6 (2021), pp. 2138–2143.

[5] Teodoro Alamo, Roberto Tempo, and Eduardo F. Camacho. "Randomized Strategies for Probabilistic Solutions of Uncertain Feasibility and Optimization Problems". In: *IEEE Transactions on Automatic Control* 54.11 (2009), pp. 2545–2559. DOI: 10.1109/TAC.2009.2031207.

[6] Mohammad Alsalti et al. "Data-based control of feedback linearizable systems". In: *arXiv preprint arXiv:2204.01148* (2022).

[7] Joel A E Andersson et al. "CasADi – A software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* (2018).

[8] Mosek ApS. "Mosek optimization toolbox for matlab". In: *User's Guide and Reference Manual, Version* 4 (2019).

[9] E Aranda-Bricaire, Ülle Kotta, and CH Moog. "Linearization of discrete-time systems". In: *SIAM Journal on Control and Optimization* 34.6 (1996), pp. 1999–2023.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[11] Abraham Bachrach. "Skydio autonomy engine: Enabling the next generation of autonomous flight". In: *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE Computer Society. 2021, pp. 1–43.

[12] Isin M Balci and Efstathios Bakolas. "Covariance control of discrete-time Gaussian linear systems using affine disturbance feedback control policies". In: *arXiv preprint arXiv:2103.14428* (2021).

[13] Ivo Batkovic et al. "A Robust Scenario MPC Approach for Uncertain Multi-modal Obstacles". In: *IEEE Control Systems Letters* 5.3 (2020), pp. 947–952.

[14] Richard Bellman. "Dynamic programming". In: *Science* 153.3731 (1966), pp. 34–37.

[15] Alberto Bemporad, Francesco Borrelli, and Manfred Morari. "Optimal controllers for hybrid systems: Stability and piecewise linear explicit form". In: *Proceedings of the 39th IEEE Conference on Decision and Control*. Vol. 2. IEEE. 2000.

[16] Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. *Convex analysis and optimization*. Vol. 1. Athena Scientific, 2003.

[17] Dimitri P Bertsekas and David Alfred Castanon. "Adaptive aggregation methods for infinite horizon dynamic programming". In: *IEEE Transactions on Automatic Control* 34.6 (1989), pp. 589–598.

[18] Dimitris Bertsimas and Bartolomeo Stellato. "Online mixed-integer optimization in milliseconds". In: *INFORMS Journal on Computing* 34 (2022).

[19] Franco Blanchini. "Set invariance in control". In: *Automatica* 35.11 (1999), pp. 1747–1767.

[20] H. A. P. Blom and Y. Bar-Shalom. "The interacting multiple model algorithm for systems with Markovian switching coefficients". In: *IEEE Transactions on Automatic Control* 33.8 (1988), pp. 780–783. DOI: 10.1109/9.1299.

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

[22] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[23] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[24] Greg Brockman et al. *OpenAI Gym*. 2016. eprint: arXiv:1606.01540.

[25] Tim Brüdigam et al. "Gaussian Process-based Stochastic Model Predictive Control for Overtaking in Autonomous Racing". In: *arXiv preprint arXiv:2105.12236* (2021).

[26] Tim Brüdigam et al. "Stochastic model predictive control with a safety guarantee for automated driving". In: *IEEE Transactions on Intelligent Vehicles* (2021).

[27] Monimoy Bujarbaruah, Siddharth H Nair, and Francesco Borrelli. "A semi-definite programming approach to robust adaptive MPC under state dependent uncertainty". In: *2020 European Control Conference (ECC)*. IEEE. 2020, pp. 960–965.

[28] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. "An Interior Point Algorithm for Large-Scale Nonlinear Programming". In: *SIAM Journal on Optimization* 9.4 (1999). DOI: 10.1137/S1052623497325107.

[29] Holger Caesar et al. "nuScenes: A multimodal dataset for autonomous driving". In: *arXiv preprint arXiv:1903.11027* (2019).

[30] Giuseppe C Calafiore and Laurent El Ghaoui. *Optimization models*. Cambridge university press, 2014.

[31] Jan-Peter Calliess et al. "Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control". In: *Automatica* 122 (2020), p. 109216.

[32] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. "Enhancing sparsity by reweighted 1 minimization". In: *Journal of Fourier analysis and applications* 14 (2008), pp. 877–905.

[33] John Canny. *The complexity of robot motion planning*. MIT press, 1988.

[34] Abhishek Cauligi et al. "Coco: Online mixed-integer control via supervised learning". In: *IEEE Robotics and Automation Letters* 7 (2021).

[35] Abhishek Cauligi et al. "PRISM: Recurrent Neural Networks and Presolve Methods for Fast Mixed-integer Optimal Control". In: *L4DC*. PMLR. 2022.

[36] Heungseok Chae et al. "Design and implementation of human driving data–based active lane change control for autonomous vehicles". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 235.1 (2021), pp. 55–77.

[37] Yuning Chai et al. "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction". In: *arXiv preprint arXiv:1910.05449* (2019).

[38] Yuxiao Chen et al. "Interactive Multi-Modal Motion Planning With Branch Model Predictive Control". In: *IEEE Robotics and Automation Letters* 7.2 (2022). DOI: 10.1109/LRA.2022.3156648.

[39] Yuxiao Chen et al. "Interactive multi-modal motion planning with branch model predictive control". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 5365–5372.

[40] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[41] Gionata Cimini and Alberto Bemporad. "Exact complexity certification of active-set methods for quadratic programming". In: *IEEE Transactions on Automatic Control* 62 (2017).

[42] McKinsey Company. "Autonomous driving's future: Convenient and connected". In: *https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected* (2023).

[43] JA De Doná et al. "A flatness-based iterative method for reference trajectory generation in constrained NMPC". In: *Nonlinear Model Predictive Control*. Springer, 2009, pp. 325–333.

[44] Alexander Domahidi, Eric Chu, and Stephen Boyd. "ECOS: An SOCP solver for embedded systems". In: (2013).

[45] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[46] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[47] Boston Dynamics. "Do you love MPC? Robot dancing via optimal control". `https://www.youtube.com/watch?v=mlTLxpKdHfA`.

[48] Jose Luis Vazquez Espinoza et al. "Deep interactive motion prediction and planning: Playing games with motion prediction models". In: *Learning for Dynamics and Control Conference*. PMLR. 2022.

[49] Lorenzo Fagiano and Andrew R Teel. "Generalized terminal state constraint for model predictive control". In: *Automatica* 49.9 (2013), pp. 2622–2631.

[50] Jianqing Fan et al. "Local linear regression smoothers and their minimax efficiencies". In: *The annals of Statistics* 21.1 (1993), pp. 196–216.

[51] Thomas Fork, H Eric Tseng, and Francesco Borrelli. "Models and predictive control for nonplanar vehicle navigation". In: *2021 IEEE international intelligent transportation systems conference (ITSC)*. IEEE. 2021, pp. 749–754.

[52] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[53] Zoubin Ghahramani and Sam T Roweis. "Learning nonlinear dynamical systems using an EM algorithm". In: *Advances in neural information processing systems*. 1999, pp. 431–437.

[54] Paul J Goulart, Eric C Kerrigan, and Jan M Maciejowski. "Optimization over state feedback policies for robust control with constraints". In: *Automatica* 42.4 (2006), pp. 523–533.

[55] Melissa Greeff and Angela P Schoellig. "Flatness-based model predictive control for quadrotor trajectory tracking". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 6740–6745.

[56] Philippe Guillot and Gilles Millerioux. "Flatness and submersivity of discrete-time dynamical systems". In: *IEEE Control Systems Letters* 4.2 (2019), pp. 337–342.

[57] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2021. URL: `https://www.gurobi.com`.

[58] Wolfgang Karl Härdle et al. *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.

[59] Navid Hashemi, Justin Ruths, and Mahyar Fazlyab. "Certifying incremental quadratic constraints for neural networks via convex optimization". In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 842–853.

[60] Trevor J Hastie. "Generalized additive models". In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[61] Wilhemus PMH Heemels, Bart De Schutter, and Alberto Bemporad. "Equivalence of hybrid dynamical models". In: *Automatica* 37 (2001).

[62] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. "Cautious model predictive control using gaussian process regression". In: *IEEE Transactions on Control Systems Technology* 28.6 (2019), pp. 2736–2743.

[63] Torsten Hothorn, Frank Bretz, and Peter Westfall. "Simultaneous inference in general parametric models". In: *Biometrical journal* 50.3 (2008), pp. 346–363.

[64] John Houston et al. *One Thousand and One Hours: Self-driving Motion Prediction Dataset*. 2020. arXiv: 2006.14480 [cs.CV].

[65] Yihan Hu et al. "Planning-oriented autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[66] Daniel Ioan et al. "Mixed-integer programming in motion planning". In: *Annual Reviews in Control* 51 (2021).

[67] Boris Ivanovic et al. *MATS: An Interpretable Trajectory Forecasting Representation for Planning and Control*. 2021. arXiv: 2009.07517 [cs.RO].

[68] Eunhyek Joa et al. "Energy-Efficient Lane Changes Planning and Control for Connected Autonomous Vehicles on Urban Roads". In: *arXiv preprint arXiv:2304.08576* (2023).

[69] Sham Kakade, Shai Shalev-Shwartz, Ambuj Tewari, et al. "On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization". In: *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf* 2.1 (2009), p. 35.

[70] Nicholas Kalouptsidis et al. "Adaptive algorithms for sparse system identification". In: *Signal Processing* 91.8 (2011), pp. 1910–1919.

[71] Christoph Kandler et al. "A differential flatness based model predictive control approach". In: *2012 IEEE International Conference on Control Applications*. IEEE. 2012, pp. 1411–1416.

[72] Hansung Kim, Siddharth H Nair, and Francesco Borrelli. "Scalable Multi-modal Model Predictive Control via Duality-based Interaction Predictions". In: *arXiv preprint arXiv:2402.01116* (2024).

[73] Torsten Koller et al. "Learning-based model predictive control for safe exploration". In: *2018 IEEE conference on decision and control (CDC)*. IEEE. 2018, pp. 6059–6066.

[74] Milan Korda, Didier Henrion, and Colin N Jones. "Convex computation of the maximum controlled invariant set for discrete-time polynomial control systems". In: *52nd IEEE Conference on Decision and Control*. IEEE. 2013, pp. 7107–7112.

[75] Basil Kouvaritakis and Mark Cannon. "Model predictive control". In: *Switzerland: Springer International Publishing* 38 (2016), pp. 13–56.

[76] Karsten Kreutz and Julian Eggert. "Analysis of the Generalized Intelligent Driver Model (GIDM) for Uncontrolled Intersections". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021.

[77] Huibert Kwakernaak and Raphael Sivan. *Linear optimal control systems*. Vol. 1. Wiley-interscience New York, 1972.

[78] Edouard Leurent and Jean Mercat. *Social Attention for Autonomous Decision-Making in Dense Traffic*. 2019. arXiv: `1911.12250 [cs.LG]`.

[79] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2017.

[80] Emilio T Maddalena et al. "KPC: Learning-based model predictive control with deterministic guarantees". In: *Learning for Dynamics and Control*. PMLR. 2021, pp. 1015–1026.

[81] Emilio Tanowe Maddalena, Paul Scharnhorst, and Colin N Jones. "Deterministic error bounds for kernel-based learning techniques under bounded noise". In: *Automatica* 134 (2021), p. 109896.

[82] Andrew Makhorin. "GLPK (GNU linear programming kit)". In: *http://www. gnu. org/s/glpk/glpk. html* (2008).

[83] Danylo Malyuta et al. "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently". In: *IEEE Control Systems Magazine* 42.5 (2022), pp. 40–113.

[84] Danilo P Mandic and Jonathon Chambers. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc., 2001.

[85] José María Manzano et al. "Robust learning-based MPC for nonlinear constrained systems". In: *Automatica* 117 (2020), p. 108948.

[86] Tobia Marcucci et al. "Motion planning around obstacles with convex optimization". In: *arXiv preprint arXiv:2205.04422* (2022).

[87] Daniele Masti and Alberto Bemporad. "Learning binary warm starts for multiparametric mixed-integer quadratic programming". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019.

[88] David Q Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[89] Garth P McCormick. "Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems". In: *Mathematical programming* (1976).

[90] Alexandre Megretski and Anders Rantzer. "System analysis via integral quadratic constraints". In: *IEEE Transactions on Automatic Control* 42.6 (1997), pp. 819–830.

[91] Donald Michie, Michael Bain, and Jean Hayes-Michie. "Cognitive models from subcognitive skills". In: *IEE control engineering series* 44 (1990).

[92] Manfred Morari and Jay H Lee. "Model predictive control: past, present and future". In: *Computers & Chemical Engineering* 23.4-5 (1999), pp. 667–682.

[93] James R Munkres. *Topology*. Vol. 2. Prentice Hall Upper Saddle River, 2000.

[94] Richard M Murray. "Nonlinear control of mechanical systems: A Lagrangian perspective". In: *Annual Reviews in Control* 21 (1997), pp. 31–42.

[95] Siddharth H Nair and Francesco Borrelli. "Robust Output-Lifted Learning Model Predictive Control". In: *arXiv preprint arXiv:2303.12127* (2023).

[96] Siddharth H Nair, Monimoy Bujarbaruah, and Francesco Borrelli. "Modeling of dynamical systems via successive graph approximations". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 977–982.

[97] Siddharth H Nair, Ugo Rosolia, and Francesco Borrelli. "Output-lifted learning model predictive control". In: *IFAC-PapersOnLine* 54.6 (2021), pp. 365–370.

[98] Siddharth H Nair, Eric H Tseng, and Francesco Borrelli. "Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 5267–5272.

[99] Siddharth H Nair et al. "Predictive Control for Autonomous Driving with Uncertain, Multi-modal Predictions". In: *arXiv preprint arXiv:2310.20561* (2023).

[100] Siddharth H Nair et al. "Stochastic MPC with Multi-modal Predictions for Traffic Intersections". In: *arXiv preprint arXiv:2109.09792* (2021).

[101] Siddharth H Nair et al. "Stochastic mpc with multi-modal predictions for traffic intersections". In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2022, pp. 635–640.

[102] Julian Nubert et al. "Safe and fast tracking on a robot manipulator: Robust mpc and neural network control". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3050–3057.

[103] Rui Oliveira, Siddharth H Nair, and Bo Wahlberg. "Interaction and Decision Making-aware Motion Planning using Branch Model Predictive Control". In: *arXiv preprint arXiv:2302.00060* (2023).

[104] Masahiro Ono and Brian C Williams. "Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint". In: *2008 47th IEEE Conference on Decision and Control*. IEEE. 2008, pp. 3427–3432.

[105] Lasse Peters et al. "Contingency games for multi-agent interaction". In: *IEEE Robotics and Automation Letters* (2024).

[106] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

[107] Hamed Rahimian and Sanjay Mehrotra. "Distributionally robust optimization: A review". In: *arXiv preprint arXiv:1908.05659* (2019).

[108] Sasa V Rakovic et al. "Invariant approximations of the minimal robust positively invariant set". In: *IEEE Transactions on automatic control* 50.3 (2005), pp. 406–410.

[109] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer School on Machine Learning*. Springer. 2003, pp. 63–71.

[110] Benjamin Recht. "A tour of reinforcement learning: The view from continuous control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 253–279.

[111] Kai Ren, Heejin Ahn, and Maryam Kamgarpour. "Chance-constrained trajectory planning with multimodal environmental uncertainty". In: *IEEE Control Systems Letters* 7 (2022), pp. 13–18.

[112] Ugo Rosolia and Francesco Borrelli. "Learning how to autonomously race a car: a predictive control approach". In: *IEEE Transactions on Control Systems Technology* 28.6 (2019), pp. 2713–2719.

[113] Ugo Rosolia and Francesco Borrelli. "Learning model predictive control for iterative tasks: A computationally efficient approach for linear system". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3142–3147.

[114] Ugo Rosolia and Francesco Borrelli. "Learning model predictive control for iterative tasks. a data-driven control framework". In: *IEEE Transactions on Automatic Control* 63.7 (2017), pp. 1883–1896.

[115] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. "Data-driven predictive control for autonomous systems". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2018), pp. 259–286.

[116] Luigi Russo et al. "Learning for online mixed-integer model predictive control with parametric optimality certificates". In: *IEEE Control Systems Letters* (2023).

[117] Tim Salzmann et al. *Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data*. 2021. arXiv: 2001.03093 [cs.RO].

[118] Tim Salzmann et al. "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer. 2020, pp. 683–700.

[119] Shankar Sastry and Marc Bodson. *Adaptive control: stability, convergence and robustness*. Courier Corporation, 2011.

[120] E Schuster, S Yakowitz, et al. "Contributions to the theory of nonparametric regression, with application to system identification". In: *The Annals of Statistics* 7.1 (1979), pp. 139–149.

[121] Mathijs Schuurmans et al. "Safe, learning-based MPC for highway driving under lane-change uncertainty: A distributionally robust approach". In: *Artificial Intelligence* 320 (2023), p. 103920.

[122] Guanya Shi et al. "Neural lander: Stable drone landing control using learned dynamics". In: *2019 international conference on robotics and automation (icra)*. IEEE. 2019, pp. 9784–9790.

[123] Raffaele Soloperto et al. "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019.

[124] Raffaele Soloperto et al. "Learning-based robust model predictive control with state-dependent uncertainty". In: *IFAC-PapersOnLine* 51.20 (2018), pp. 442–447.

[125] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *J. Mach. Learn. Res.* 15.1 (2014).

[126] Marco Pavone Sumeet Singh and Jean-Jacques Slotine. "Tube-Based MPC: a Contraction Theory Approach". In: *Decision and Control, 2016. CDC. 55th IEEE Conference on*. 2016.

[127] Weehong Tan and Andrew Packard. "Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming". In: *IEEE Transactions on Automatic Control* 53.2 (2008), pp. 565–571.

[128] Shumpei Tokuda et al. "Convex approximation for ltl-based planning". In: (2021).

[129] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: `1706.03762 [cs.CL]`.

[130] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017.

[131] Abraham P Vinod, Arie Israel, and Ufuk Topcu. "On-the-fly control of unknown nonlinear systems with sublinear regret". In: *IEEE Transactions on Automatic Control* (2022).

[132] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[133] Allen Wang, Ashkan Jasour, and Brian C Williams. "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6041–6048.

[134] Po-Wei Wang and Chih-Jen Lin. "Iteration complexity of feasible descent methods for convex optimization". In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1523–1548.

[135] Zejiang Wang, Jingqiang Zha, and Junmin Wang. "Flatness-based model predictive control for autonomous vehicle trajectory tracking". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 4146–4151.

[136] Christopher JCH Watkins and Peter Dayan. "Q-learning". In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[137] Waymo. "Sense, solve and go: The magic of the Waymo driver". `https://www.youtube.com/watch?v=hA_-MkU0Nfw`.

[138] Lianzhen Wei et al. "Autonomous Driving Strategies at Intersections: Scenarios, State-of-the-Art, and Future Outlooks". In: *arXiv preprint arXiv:2106.13052* (2021).

[139] Skylar X Wei et al. "Moving obstacle avoidance: A data-driven risk-aware approach". In: *IEEE Control Systems Letters* 7 (2022), pp. 289–294.

[140] Sze Zheng Yong, Brian Paden, and Emilio Frazzoli. "Computational methods for MIMO flat linear systems: Flat output characterization, test and tracking control". In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 3898–3904.

[141] Shuyou Yu et al. "Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems". In: *Systems & Control Letters* 62.2 (2013), pp. 194–200.

[142] Chiyuan Zhang et al. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64 (2021).

[143] Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. "Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework". In: *IEEE Transactions on Control Systems Technology* 29 (2020).

[144] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. "Optimization-based collision avoidance". In: *IEEE Transactions on Control Systems Technology* (2020).

[145] Xiaojing Zhang et al. "Autonomous parking using optimization-based collision avoidance". In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 4327–4332.

[146] Bingyu Zhou et al. "Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2388–2394.

[147] Jian Zhou, Björn Olofsson, and Erik Frisk. "Interaction-Aware Motion Planning for Autonomous Vehicles with Multi-Modal Obstacle Uncertainty Predictions". In: *IEEE Transactions on Intelligent Vehicles* (2023).

[148] Edward L Zhu and Francesco Borrelli. "A sequential quadratic programming approach to the solution of open-loop generalized nash equilibria". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023.

[149] Jia-Jie Zhu and Georg Martius. "Fast non-parametric learning to accelerate mixed-integer programming for hybrid model predictive control". In: *IFAC-PapersOnLine* 53 (2020).