Ph.D. DISSERTATION

WASSERSTEIN DISTRIBUTIONALLY ROBUST CONTROL AND OPTIMIZATION FOR AUTONOMOUS SYSTEMS

자율시스템을 위한 Wasserstein
분포적 강건 제어 및 최적화

BY

ASTGHIK HAKOBYAN

AUGUST 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

WASSERSTEIN DISTRIBUTIONALLY ROBUST CONTROL AND OPTIMIZATION FOR AUTONOMOUS SYSTEMS

자율시스템을 위한 Wasserstein
분포적 강건 제어 및 최적화

BY

ASTGHIK HAKOBYAN
AUGUST 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

WASSERSTEIN DISTRIBUTIONALLY ROBUST CONTROL AND OPTIMIZATION FOR AUTONOMOUS SYSTEMS

자율시스템을 위한 Wasserstein
분포적 강건 제어 및 최적화

지도교수 양 인 순
이 논문을 공학박사 학위논문으로 제출함

2023년 5월

서울대학교 대학원

전기 · 정보공학부

아스트힉

아스트힉의 공학박사 학위 논문을 인준함

2023년 6월

위 원 장:	박종우
부위원장:	양인순
위 원:	김현진
위 원:	정교민
위 원:	Naira Hovakimyan

Abstract

Distributionally robust control (DRC) and optimization (DRO) have recently become popular approaches for handling uncertain distributional information in stochastic systems with accuracy. In this work, we develop novel control methods for autonomous systems in situations where only limited information is available about the uncertainties in system or environment models. To achieve this, we estimate the uncertainty distribution using disturbance samples or state-of-the-art learning techniques and construct an ambiguity set around the nominal distribution. Our ambiguity set contains all distributions whose Wasserstein distance from the nominal one is less than the given radius. We then solve the optimal control problem with respect to the worst-case distribution within the ambiguity set. However, the resulting problem is infinite-dimensional and intractable. Therefore, we apply modern tools from DRO to develop several methods for solving the Wasserstein DRC (WDRC) problem in various settings with different theoretical properties and applications.

Our first method proposes a novel safety specification tool, the distributionally robust risk map (DR-risk map), for motion planning and control of a mobile robot in a learning-enabled environment. The DR-risk map reliably assesses the conditional value-at-risk of collision with obstacles whose movements are inferred by Gaussian process regression. Our tool measures the risk under the worst-case distribution within the ambiguity set to account for errors in the inferred distribution. To resolve the intractability, we develop a semidefinite programming (SDP) formulation that provides an upper bound of the risk. We apply the DR-risk map to perform motion planning and control of autonomous systems in learning-enabled environments.

Our second method introduces a novel learning-based motion control tool that uses an uncertainty propagation scheme based on an unscented transform to achieve better prediction accuracy and computational efficiency. In addition, this approach replaces

the DR-risk constraint for any arbitrary safety loss function with a novel simpler upper bound.

The WDRC framework can be applied not only to fully observable systems but also to partially observable systems, which are more realistic. In our next stage, we focus on the WDRC problem for partially observable linear stochastic systems and present a new approximation scheme. This method leverages the Gelbrich bound of the Wasserstein distance to penalize deviations from the nominal distribution. We derive a closed-form expression for the optimal control policy and a tractable SDP problem for the worst-case distribution policy in both finite-horizon and infinite-horizon average-cost settings. Our proposed method features several salient theoretical properties, such as a guaranteed cost property and a probabilistic out-of-sample performance guarantee, demonstrating the distributional robustness of our controller. Furthermore, the resulting controller ensures the closed-loop stability of the mean-state system.

Finally, we present a novel distributionally robust differential dynamic programming algorithm for approximately solving the general nonlinear WDRC problem in a tractable and scalable way. It provides a closed-form control policy for nonlinear stochastic systems and therefore is applicable to learning-enabled environments. Our approach features a novel decomposition of the value function and its iterative local-quadratic approximations, making our method tractable and scalable without the need for numerically solving any minimax optimization problems.

We analyze and demonstrate the effectiveness of our methods through simulation studies on various systems, ranging from oscillator synchronization to autonomous driving problems. Our contributions enable controllers that can handle distributional uncertainties in both system and environment dynamics, as well as learning outcomes.

keywords: Distributionally Robust Control, Distributonally Robust Optimization, Motion Planning, Motion Control, Robot Safety

student number: 2021-37761

Contents

List of Tables

List of Figures

Chapter 1

BACKGROUND AND OBJECTIVES

1.1 Motivation and Objectives

Autonomous systems such as self-driving cars, robots, and smart manufacturing systems can have transformative impacts on our society. However, the performance of such systems critically depends on the quality of information about the system model, its environment, and the stochastic uncertainties affecting the system. This becomes particularly challenging when the controller only has access to partial information about the system coming from the noisy measurements. Advances in machine learning allow inferring the unknown models given sensor measurements. However, the accuracy of the inference depends significantly on the quality of the data, statistical models, and learning methods used. Therefore, in practice, obtaining an accurate probability distribution of disturbances is often challenging.

The theory of optimal control addresses uncertain systems with full or partial state information through stochastic or robust control frameworks. Stochastic control approaches assume the accuracy of provided distributional information and utilize it directly for system control. However, relying on unreliable information about uncertainties can lead to undesirable system behavior, resulting in catastrophic events such as collisions and accidents. For instance, as depicted in Fig. 1.1, a collision occurs due

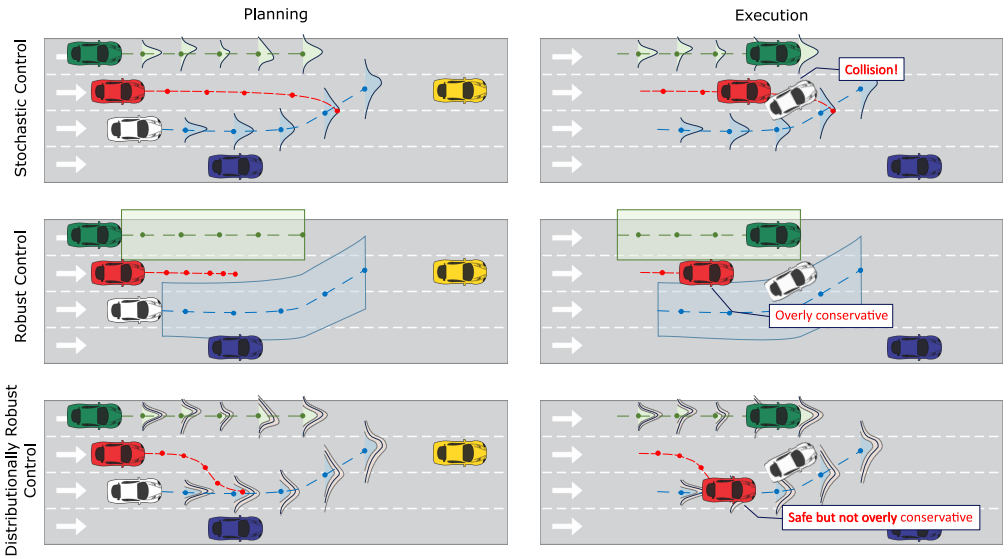


Figure 1.1: Comparison of stochastic, robust, and DRC methods. The planned trajectory of the ego vehicle is shown in red, while the learned trajectories of obstacles are represented by green and blue lines.

to learning inaccuracies when executing a stochastic control policy that uses learned trajectories of surrounding obstacles to control the ego vehicle (red). On the other hand, Robust control methods aim to design a controller for the worst-case realization of uncertainties, disregarding potentially useful but unreliable statistical information about the disturbance distribution. This often leads to overly conservative behavior, exemplified in Fig. 1.1.

The primary objective of this research is to tackle the core question:

How can we design an optimal controller for fully and partially observable autonomous systems that is robust against distributional inaccuracies in given (learned) nominal information?

To address this question, we propose several control approaches based on the *distributionally robust optimization* (DRO) that use limited data to make decisions while hedging against distributional mismatches between the true distribution and the nomi-

nal one. Moreover, we demonstrate that these methods offer theoretical and empirical performance guarantees, including system safety, stability, and out-of-sample performance, among others. By employing the proposed methods, we bridge the gap between stochastic and robust control approaches by striking a balance between performance and conservativeness (e.g., Fig. 1.1).

1.2 Related Works

1.2.1 Optimal Control of Systems Under Uncertainties

The literature on optimal control of systems under uncertainties can be mainly categorized into stochastic and robust methods. Stochastic optimal control methods aim to design a controller by considering the underlying uncertainty distribution, often assuming it to be Gaussian. A notable approach in this direction is the linear quadratic Gaussian (LQG) control method [1–4]. LQG minimizes the expected value of the quadratic cost function given the measurements and assumes known disturbance statistics. By leveraging the certainty equivalence principle, it provides a feedback control policy with the same closed-form expression as in the deterministic case [5]. Under specific conditions, LQG exhibits outstanding asymptotic behavior, resulting in a stable closed-loop system with a steady-state policy [4].

Another popular tool is the stochastic version of model predictive control (MPC) [6], which is often used to handle control problems in nonlinear stochastic systems with uncertainties and disturbances [7, 8]. Stochastic MPC considers the probabilistic nature of uncertainties and optimizes control actions to minimize the expected cost or achieve desired performance criteria over a finite future horizon. It formulates the control problem as a finite-horizon optimization problem, where the control inputs are computed by optimizing a cost function that incorporates the system dynamics and uncertainty distributions. Various techniques from stochastic optimization and numerical optimization, such as stochastic programming-based approaches [9, 10], scenario-

based approaches [11, 12], or simulation-based optimization methods [13, 14], are employed to solve stochastic MPC problems. Stochastic MPC often requires generating multiple scenarios or samples to adequately capture the uncertainty distribution, using techniques like Monte Carlo simulation or sample-based optimization.

On the other hand, robust optimal control addresses uncertainties without assuming a specific underlying distribution. Instead, it considers a prespecified uncertainty set and seeks to find a worst-case controller for achieving robust performance [15–18]. Of particular interest are the H_2 and H_∞-optimal control methods, which are closely associated with robust stabilization of uncertain systems [18–21]. Both methods aim to design a stabilizing controller by minimizing the H_2 or H_∞-norm of the closed-loop system, treating disturbances as external inputs. Although the original problem is formulated in the frequency domain, it can be equivalently formulated as a two-player zero-sum game in the time domain. Robust MPC has witnessed significant developments in the past two decades, aiming to handle uncertainties in stochastic constrained nonlinear optimal control problems. Early work on robust MPC primarily relied on minimax formulations, where control actions are designed with respect to worst-case evaluations of the cost function and constraints that must hold for all possible uncertainty realizations [22, 23]. To address the conservativeness and infeasibility of minmax MPCs, tube-based MPC has been developed, which employs a partially separable feedback control law to handle uncertainties and their interactions with system dynamics [24–26].

In practice, only limited knowledge, such as previous observations, is available about the uncertainties. In such settings, stochastic optimal control methods are not appropriate, as possible inaccuracies are ignored during control design. On the other hand, robust methods result in conservative controllers, as any distributional information about the uncertainties is disregarded. Recently, distributionally robust control (DRC) has emerged as an alternative to stochastic and robust methods, capturing robust yet not overly-conservative performance [27–35]. In DRC, the optimal control

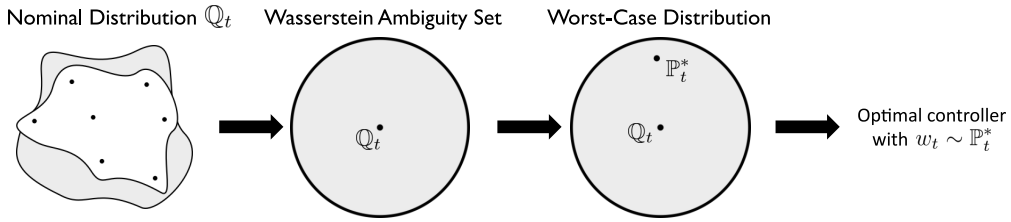


Figure 1.2: Illustration of the concept of DRC.

policy sought to minimize the expected cost with respect to the worst-case probability distribution within an *ambiguity set* (see Fig. 1.2). DRC can be regarded as a dynamic or multi-stage version of DRO. In the literature regarding DRO, it is common to design the ambiguity set based on a nominal distribution constructed from data so that it contains the true distribution with high probability. For example, moment-based ambiguity sets are popular in DRO, which include distributions satisfying some moment constraints [36, 37]. Despite outstanding tractability properties, such sets often yield conservative decisions and require accurate moment estimates. Designing the ambiguity set based on statistical distances to contain distributions close to the given nominal one is another popular option. Among various distances, such as the KL-divergence and Prokhorov metric [38], the Wasserstein metric attracts significant attention not only in DRO [39–41] but also in DRC [29–33, 42]. The Wasserstein ambiguity set has a number of useful features, including offering a powerful finite-sample performance guarantee [39, 43]. Furthermore, it is rich enough to contain relevant distributions, thereby encouraging the DRO problem to avoid providing pathological solutions [40].

In contrast to research on fully observable settings, the literature about partially observable DRC is relatively sparse. A few works are devoted to the distributionally robust version of the LQG control method. For example, [44, 45] propose a minimax LQG controller that minimizes the worst-case performance by restricting the KL-divergence between the disturbance distribution and a given reference distribution. In [46], a partially observable Markov decision process is considered with finite state, action, and observation spaces. The ambiguity set is chosen to bound the moments of

the joint distribution of the transition-observation probabilities. Another type of partially observable systems, namely the Markov jump linear system, is studied in [28]. The authors propose a mechanism for estimating the active mode in a receding horizon fashion and integrate this procedure with a data-driven distributionally robust controller design using the total variation distance. In [31], a data-enabled distributionally robust predictive control method is proposed and studied using noise-corrupted input and output data.

1.2.2 Safety in Learning-Enabled Environments

Safe learning for control is a fundamental problem in the field of robotics. Existing methods can be categorized based on the learning models they employ, namely deterministic or probabilistic approaches. The first class predominantly includes deep neural networks [47–49], while the second class comprises methods like Gaussian Processes (GPs) [50–52], Bayesian linear regression [53, 54], and others. These learning methods are often combined with safety specification tools, such as reachability-based approaches [55–57].

Another direction in ensuring safety during learning is through safe exploration techniques that leverage Lyapunov stability [52, 58, 59] or barrier functions [53, 60]. Learning-based MPC is also a popular approach that applies safe learning to control. Most research efforts in this field focus on improving the prediction model by learning the system dynamics or fine-tuning its parameters [61–63]. Recently, MPC-based safety filters have been introduced to enforce constraint satisfaction for any learning-based controller [64, 65].

In addition to conventional safety specification tools, modern approaches address robot safety through various risk measures. Often, the risk is quantified by the probability of collision, where the uncertainty arises from the learned robot model [51, 66, 67]. Another approach for assessing risk is conditional value-at-risk (CVaR) [68], a coherent measure widely advocated as a rational risk metric in robotics [69]. CVaR quan-

tifies potential safety losses in the tail of the distribution and accounts for rare but catastrophic events, such as collisions. For example, in [70], the authors propose a safety constraint using CVaR to ensure safe robot navigation. Furthermore, in [71], risk-averse policies are learned using offline data by optimizing the CVaR of the cost. However, all the mentioned methods utilize the learned distribution to evaluate safety risk without considering learning errors.

Recent research has addressed learning inaccuracies by employing DRC. Moment-based ambiguity sets are often used to add robustness to chance constraints [72, 73]. However, such sets tend to be overly conservative and rely on the reliability of moment estimates. Wasserstein balls are another popular type of ambiguity sets used in robotics [31, 74, 75]. Both [31] and [75] limit the risk of unsafety through a distributionally robust version of the CVaR constraint using the empirical distribution of the system outputs and the learned mean and covariance of the environment states, respectively.

1.3 Research Contributions

In this thesis, we present four novel Wasserstein distributionally robust control and optimization methods, each designed using different techniques and problem settings. The main properties and features of these methods are summarized in Figure 1.3. The main contributions of this work can be summarized as follows.

First, we address the issue of safety in motion planning and control of learning-based autonomous systems evolving in an unknown dynamic environment. In our method, the obstacles' behavior is inferred via a Gaussian process regression (GPR) using real-time observation. Then, we propose a novel safety specification tool called the *distributionally robust risk map* (DR-risk map) that is robust against errors in learning results about the obstacles' locations. Our risk map utilizes CVaR to measure the risk of unsafety given the worst-case distribution within a Wasserstein ambigu-

Method	System Type	Uncertainty type	Controller Type	Properties
DR-risk map	Nonlinear	Environment	Implicit	▪ Learned environment ▪ Tractable SDP risk upper bound ▪ Probabilistic guarantee of safety
UT-MPC	Nonlinear	System/ Environment	Implicit	▪ Learned system and environment dynamics ▪ Improved prediction accuracy and computational efficiency ▪ Tractable analytical risk upper bound
PO-WDRC	Partially observable, Linear	System	Explicit	▪ Approximation via a Wasserstein penalty ▪ Works in both finite- and infinite-horizon settings ▪ Theoretical properties (guaranteed cost property, probabilistic out-of-sample performance guarantee, closed-loop stability)
DR-DDP	Nonlinear	System	Explicit	▪ Approximation via a Wasserstein penalty ▪ Iterative locally-quadratic approximations ▪ Scalable to high-dimensional systems ▪ Guaranteed cost property

Figure 1.3: The main properties and features of the proposed methods.

ity set. To alleviate the infinite-dimensionality issue of the DR-risk map, we propose a tractable semidefinite programming formulation that provides an upper bound of the DR-risk map. Furthermore, we show that the DR-risk map provides a probabilistic guarantee on the loss of safety. Next, we demonstrate the utility of the risk map in learning-based planning and control. Specifically, we develop a planning algorithm that uses the risk map for generating safe trajectories and introduce an MPC method with a risk constraint that can be evaluated by using its neural network approximation. The performance and utility of the DR-risk map are demonstrated through simulation studies for autonomous vehicles and service robots.

Next, we focus on ensuring the safe motion control of learning-based systems, where the system model is not known, in contrast to the previous approach that only dealt with unknown environment dynamics. We propose learning the unknown dynamics using GPR and then exploiting unscented transform (UT) to improve the computation efficiency and prediction accuracy of both the robot and the environment. To immunize the controller against distributional uncertainties, we again design an MPC controller with the distributionally robust CVaR constraint (DR-CVaR), which com-

bines the advantages of both UT and DRO within a single framework. To overcome the intractability, we devise a simple analytical upper bound of DR-CVaR exploiting UT to estimate the safety loss distribution. As a result, we obtain a tractable distributionally robust UT-MPC algorithm that guides the robot to take cautious actions despite learning inaccuracies. Our experiment results in an autonomous driving problem demonstrate the capability of our algorithm to promote safe motion control in dynamic environments, even in the presence of learning errors.

The distributionally robust control framework is not limited to fully observable systems, such as those mentioned previously, but can also be applied to partially observable systems that are more representative of real-world scenarios. As a result, we tackle the challenge of controlling partially observable stochastic systems, with a particular emphasis on the linear-quadratic case where the actual distribution of system disturbances is not known. We first formulate a Wasserstein distributionally robust control (WDRC) problem and propose a novel approximation technique with a special penalty term using the Gelbrich bound on the Wasserstein distance. The resulting partially observable WDRC (PO-WDRC) problem is solved using the dynamic programming principle to derive a non-trivial Riccati equation alongside the closed-form optimal control policy in both finite- and infinite-horizon settings. Finally, an extensive theoretical analysis is performed for the resulting controller, which is shown to possess a number of salient features, such as a guaranteed cost property, probabilistic out-of-sample performance guarantee, closed-loop stability, etc.

Finally, we introduce a new algorithm called distributionally robust differential dynamic programming (DR-DDP) that can handle a broader range of nonlinear WDRC problems, thereby closing the gap between existing WDRC methods for linear systems and enabling its application in learning-based environments. We develop a novel method that uses a locally quadratic approximation of the nonlinear WDRC problem to provide closed-form control policies that are robust against inaccuracies in the distributional information of the disturbances. For tractability, we use the Kantorovich

duality principle and decompose the value function in a novel way to derive computationally tractable backward and forward passes. The advantage of the proposed approach is not only tractability but also scalability, as there is no need to numerically solve any minimax optimization problem. We show that unlike the standard dynamic programming algorithm for nonlinear WDRC, the computational complexity of our DR-DDP is polynomial in the dimension of the state space. Moreover, the resulting control policy is shown to enjoy guaranteed cost property. We demonstrate the performance of the algorithm on a kinematic car navigation and oscillator synchronization problem, showing its applicability to a wide range of real-world problems.

In summary, the proposed WDRC methods enable the design of controllers that are robust to distributional uncertainties in both system and environment dynamics, even in the presence of learning inaccuracies. Our methods demonstrate exceptional empirical performance while also featuring a number of salient theoretical properties and guarantees.

1.4 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 introduces the DR-risk map, presenting its tractable upper bound and probabilistic guarantee on safety loss. We also describe the motion planning and control algorithms that utilize the DR-risk map to address errors caused by GPR. Simulations demonstrate the effectiveness of this approach in various autonomous navigation problems. Chapter 3 introduces the UT-MPC algorithm, which utilizes a UT-based uncertainty propagation scheme for improved computational efficiency and prediction accuracy. We devise an analytical upper bound of DR-CVaR that exploits the UT approach, ensuring the tractability of the problem. The performance of the proposed method is demonstrated through simulations in an autonomous driving scenario. Chapter 4 addresses the WDRC problem for partially observable linear stochastic systems. We introduce a tractable approximation and de-

rive its solution in both finite- and infinite-horizon average-cost settings. We analyze the theoretical properties of the resulting controller and discuss the stability aspects of the closed-loop system. We demonstrate the performance of the proposed method through numerical experiments on a power system frequency control problem. Finally, Chapter 5 introduces the DR-DDP algorithm for nonlinear stochastic systems. We derive an approximation to the nonlinear WDRC problem and develop a computationally tractable backward and forward passes. Numerical experiments demonstrate the out-of-sample performance of our algorithm and its scalability to high-dimensional state spaces.

Chapter 2

Distributionally Robust Risk Map for Learning-Based Motion Planning and Control: A Semidefinite Programming Approach

2.1 Introduction

Ensuring safety in motion planning and control critically depends on the quality of information about the possibly uncertain environment in which a robot operates. For example, a mobile robot may use sensor measurements to take into account the uncertain behavior of other robots, human agents, or obstacles for collision avoidance. With advances in machine learning, sensing, and computing technologies, the adoption of state-of-the-art learning techniques is rapidly growing for a robot to infer the evolution of its environment. Unfortunately, the accuracy of inference is often poor since it is subject to the quality of the observations, statistical models, and learning methods. Using inaccurately learned information in the robot's decision-making may induce unwanted behaviors and, in particular, may lead to a collision. This work aims to develop a safety risk specification tool that is robust against distribution errors in learned information about moving obstacles and is thus useful for ensuring safety in learning-based motion planning and control.

Safety specification tools for systems with learning-enabled components can be categorized into two classes. The first class concerns the safety of learning-enabled robots, while the second class considers learning-enabled environments. The tools in the first class use or learn reachable sets [56, 76], Lyapunov functions [77, 78], or control barrier functions [60, 79, 80] as a certificate for safety when the system dynamics of robots are unknown. The literature on the second class is relatively sparse. Existing methods to handle learning-enabled environments use chance constraints [81], logistic functions [82], collision detection via Monte Carlo sampling [83], and detection of conflicts between intention and expectation [84], among others. Our method belongs to the second class. Departing from the aforementioned tools, we propose to use a risk measure for safety analysis in learning-enabled environments. Among various risk measures [85–87], we adopt the *conditional value-at-risk* (CVaR) for its capability of distinguishing rare tail events [68, 88].

This work is also related to learning-based motion planning and control, which are the main applications of our safety specification tool. The following two cases are considered in the literature: (i) learning the system dynamics of robots, and (ii) learning the environment. The first case is the most well-studied direction, which is based on RRT* [89, 90], model-predictive control [51, 61, 67, 91], and model-based reinforcement learning (RL) [92–94], among other methods. These tools employ various learning or inference techniques to update unknown system model parameters that are, in turn, used to improve control actions or policies. On the other hand, the methods in the second class emphasize learning the environment. In particular, for learning the behavior (or intention) of obstacles or other vehicles, several methods have been proposed that use inverse RL [95–97], imitation learning [98, 99], and Gaussian mixture models [100, 101], among others. The learned information about environments can then be used in probabilistic or robust motion planning and control algorithms [102–107]. Our method is classified as the second type since it uses the learned information about the motion of obstacles. However, unlike the previous approaches, we emphasize the im-

portance of decision-making that is robust against potential errors caused by learning the environment. For this, we take a distributionally robust optimization (DRO) approach [39–41] to address errors in learned information about the motion of obstacles.

In this work, we propose a novel safety specification tool, which we call the *distributionally robust risk map* (DR-risk map). It is a spatially varying function that specifies the safety risk in a way that is robust against errors in learning or prediction results about the obstacles' locations. Specifically, the obstacles' future trajectories are assumed to be inferred using GPR based on the current and past observations. However, the predicted probability distribution of the obstacles' locations is subject to errors, making it difficult to accurately evaluate the risk of collision. To resolve this issue, our method evaluates the risk under the worst-case distribution in a so-called *ambiguity set*. Thus, the robot's decision made using the DR-risk map will generate a safe behavior even when the true distribution deviates from the learned one within the ambiguity set. Unfortunately, the computation of DR-risk is challenging since it involves the infinite-dimensional optimization problem over the ambiguity set of probability distributions.

The main contributions of this work are threefold. First, we propose a tractable semidefinite programming (SDP) formulation that provides an upper bound of the DR-risk map. The SDP approach, which exploits techniques from DRO, alleviates the infinite-dimensionality issue inherent in the DR-risk map. Further, we provide its dual formulation, which has fewer generalized inequalities, as well as a probabilistic guarantee on the loss of safety. Second, we demonstrate the utility of the DR-risk map in learning-based motion planning. A distributionally robust RRT* algorithm is proposed to use the risk map for generating a safe path despite the learning errors caused by GPR. Third, we devise a motion control tool that employs the neural network (NN) approximation of the DR-risk map. Our method uses MPC with risk constraints that can be evaluated by solving SDPs. To avoid solving the SDPs in real-time, we propose approximating the DR-risk map as an NN, which is then embedded in the MPC prob-

lem. Our NN approximation has the salient feature that the same NN can be used to approximate the DR-risk map for any time and any obstacles since the dependence is encoded in the input information. The performance and utility of the DR-risk map are demonstrated through simulation studies for autonomous vehicles and service robots. The results of our experiments show that our motion planning and control tools successfully ensure safety even in the presence of distribution errors caused by GPR.

This paper has been significantly expanded from its preliminary conference version [74]. The DR-risk map is formally defined, and its SDP approximation and performance guarantee are proposed in this chapter. In particular, the construction of DRO is simplified without sampling from the distribution obtained by GPR. Furthermore, a motion planning algorithm is proposed using the DR-risk map, unlike the conference version, which focuses on motion control. Last but not least, the NN approximation of risk constraints in motion control is newly considered in this chapter. We also clarify the distinction between this paper and our previous work [33]. In [33], a DR-CVaR constraint is used to ensure the safety of the robot in the presence of additive environmental uncertainties by simply considering the empirical distribution. However, the focus of the current paper is entirely different in that we aim to address learning inaccuracies when the motion of the obstacles is learned by GPR. In this distinct setting, our motion control tool is constructed in a novel way exploiting techniques from SDP and NN approximations.

The remainder of the paper is organized as follows. In Section 2.2, we introduce the problem setup and the GPR approach to learning the future trajectories of obstacles. In Section 2.3, we define the DR-risk map and present its tractable reformulation as an SDP. In Section 2.4, we propose a motion planning algorithm using the DR-risk map to address errors caused by GPR. In Section 2.5, the risk map is approximated by an NN and applied to an MPC problem for motion control. Finally, in Section 2.6, we present the application of our risk map to motion planning and control problems through simulations in various environments.

2.2 Preliminaries

2.2.1 Mobile Robot and Obstacles

In this work, we consider a mobile robot modeled by the following discrete-time system:

$$\begin{aligned} x_r(t+1) &= f(x_r(t), u_r(t)) \\ y_r(t) &= Cx_r(t), \end{aligned} \tag{2.1}$$

where $x_r(t) \in \mathbb{R}^{n_x}$, $u_r(t) \in \mathbb{R}^{n_u}$ and $y_r(t) \in \mathbb{R}^{n_y}$ are the robot's state, input, and output, respectively, where the subscript 'r' represents 'robot'. The system output is defined as the Cartesian coordinates of the robot's center of mass (CoM).

The robot navigates a cluttered environment with L moving obstacles, e.g., other robotic vehicles. The motion of the ℓth obstacle is described by the following discrete-time system for $\ell = 1, \ldots, L$:

$$x_o^\ell(t+1) = \phi^\ell(x_o^\ell(t), u_o^\ell(t)) \tag{2.2}$$

$$y_o^\ell(t) = C_o^\ell x_o^\ell(t), \tag{2.3}$$

where $x_o^\ell(t) \in \mathbb{R}^{n_x^\ell}$ and $u_o^\ell(t) \in \mathbb{R}^{n_u^\ell}$ are the obstacle's state and input, respectively. The subscript 'o' represents 'obstacle'. The output $y_o^\ell(t) \in \mathbb{R}^{n_y}$ is the Cartesian coordinates of the obstacle's CoM and has the same dimension as the robot's output $y_r(t)$. Here, ϕ^ℓ is a possibly unknown (nonlinear) function. In practice, ϕ^ℓ can be replaced with its parametric approximation ϕ_w^ℓ, for example, using NNs, and the parameters w can be estimated using training data. See Appendix 2.8.1 for an example. For ease of exposition, we assume that ϕ^ℓ or its parametric approximation is given.

For safety, our robot should navigate within a safe region, which is determined by the obstacles' behaviors. To define the safe region, we over-approximate each obstacle as the smallest enclosing ball centered at the CoM of the obstacle.[1] The safe region

[1] Our method can handle obstacles of any shape through the proposed over-approximation. This approach might be conservative in certain cases. However, the conservativeness is beneficial for ensuring safety.

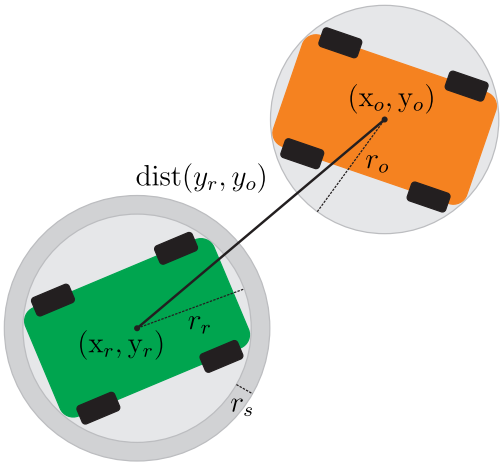


Figure 2.1: The car-like robot (green) is centered at $y_r := (\mathrm{x}_r, \mathrm{y}_r)$, while the obstacle (orange) is centered at $y_o := (\mathrm{x}_o, \mathrm{y}_o)$. The smallest balls enclosing the robot and the obstacle have radii r_r and r_o, respectively. With margin r_s, the safe distance r_ℓ can be chosen as $r_r + r_o + r_s$.

for each obstacle can be defined as the region outside the open ball centered at the obstacle's CoM with *safe distance* $r_\ell > 0$:

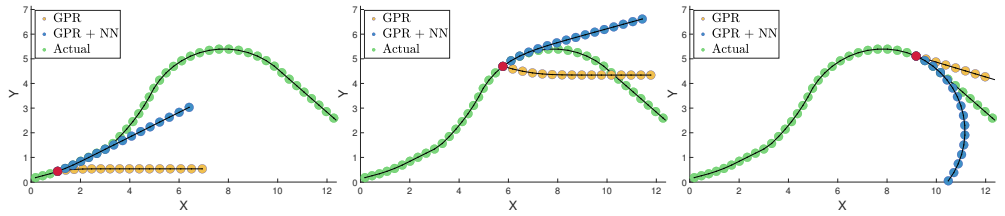
$$\mathcal{Y}^\ell(t) := \left\{ y_r(t) \in \mathbb{R}^{n_y} \mid \mathrm{dist}(y_r(t), y_o^\ell(t)) \geq r_\ell \right\}, \tag{2.4}$$

where $\mathrm{dist}(y_r(t), y_o^\ell(t))$ is the Euclidean distance between the robot's CoM and the obstacle's CoM, defined by

$$\mathrm{dist}(y_r(t), y_o^\ell(t)) := \|y_r(t) - y_o^\ell(t)\|_2.$$

An example of such a configuration is shown in Fig. 2.1, where a car-like robot (green) should navigate to avoid a car-like obstacle (orange). Both the robot and the obstacle are approximated by the smallest balls enclosing them with radii r_r and r_o^ℓ, respectively. Using an additional safety margin r_s, the distance between the CoMs of the robot and the obstacle should be no smaller than the sum of all radii:

$$r_\ell = r_r + r_o^\ell + r_s.$$



(a) $t = 3$ (b) $t = 20$ (c) $t = 29$

Figure 2.2: Trajectories of an obstacle predicted using GPR with and without neural network approximation of the dynamics. The mean of each trajectory is represented by a point, while the covariance is represented by an ellipsoid.

Having L surrounding obstacles, the safe region with respect to all obstacles is defined as the intersection of all the safe regions $\mathcal{Y}^\ell(t)$:

$$\mathcal{Y}(t) := \bigcap_{\ell=1}^{L} \mathcal{Y}^\ell(t).$$

Note that the safe region is time-varying.

2.2.2 Learning the Motion of Obstacles via Gaussian Process Regression

Even though the dynamics ϕ^ℓ of obstacles are assumed to be known or estimated using some function approximators, the actions taken by the obstacles are unknown; thus, our robot has no information about the obstacles' future behaviors. Furthermore, even if the actions were known, the resulting trajectories might include some inaccuracies since ϕ^ℓ might not accurately describe the real motion of the obstacles. To take such uncertainties into account, the observations made by the robot can be useful for inferring (or learning) the obstacles' movements.

In this study, we use GPR, which is one of the most popular non-parametric methods for learning a probability distribution over all possible values of a function [108]. Ideally, GPR can be used to directly infer the future state of obstacle ℓ given the current state information. However, leveraging some information about the system dynamics

as a global model can significantly increase the accuracy of local predictions and reduce the size of the required training data. Hence, in this work, we aim to learn the function ψ^ℓ that corresponds to the control action of the obstacle ℓ given its state information and use it in conjunction with the obstacle dynamics ϕ^ℓ to predict the future trajectory. For ease of exposition, we suppress the superscript ℓ.

GPR is performed on a training dataset, which is constructed from previous observations about the obstacle's state and action. In particular, at stage t, the training input data is chosen as $\hat{\mathrm{x}} = \{x_o(t-1), x_o(t-2), \ldots, x_o(t-M)\}$ with the corresponding training output data $\hat{\mathrm{y}} = \{u_o(t-1), u_o(t-2), \ldots, u_o(t-M)\}$, where M is the number of observations. Since observations are imperfect, we assume that for the ith observation

$$\hat{\mathrm{y}}^i = \psi(\hat{\mathrm{x}}^i) + v^{(i)},$$

where v is an i.i.d. zero-mean Gaussian noise with covariance

$$\Sigma^v = \mathrm{diag}([\sigma_{v,1}^2, \sigma_{v,2}^2, \ldots, \sigma_{v,n_u}^2]).$$

Assuming that each control action has independent entries, the GPR dataset for the jth dimension of control action is constructed as

$$\mathcal{D}_j = \left\{ \left(\hat{\mathrm{x}}^i, \hat{\mathrm{y}}_j^i\right),\ i = 1, \ldots, M \right\}$$

for $j = 1, \ldots, n_u$.

In GPR, each dimension of $\psi(\cdot)$ has a Gaussian prior distribution with mean function $m_j(x)$ and kernel $k_j(x, x')$. In this work, we use a zero-mean prior with the following radial basis function (RBF) kernel:

$$k_j(x, x') = \sigma_{f,j}^2 \exp\left[-\frac{1}{2}(x - x')^\top L_j^{-1}(x - x')\right],$$

where L_j is a diagonal length scale matrix and $\sigma_{f,j}^2$ is a signal variance. The prior on the noisy observations is a normal distribution with mean function $m_j(\hat{\mathrm{x}}^i)$ and covariance function $K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma_{v,j}^2 I$, where $K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}})$ denotes the $M \times M$ covariance matrix of training input data, i.e., $K_j^{(l,k)}(\hat{\mathrm{x}}, \hat{\mathrm{x}}) = k_j(\hat{\mathrm{x}}^{(l)}, \hat{\mathrm{x}}^{(k)})$.

For a new arbitrary test point $\mathbf{x}$, the posterior distribution of the jth output entry is also Gaussian. Its mean and covariance are calculated as follows:

$$\mu_u^j(\mathbf{x}) = m_j(\mathbf{x}) + K_j(\mathbf{x}, \hat{\mathrm{x}})(K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma_{v,j}^2 I)^{-1}(\hat{\mathrm{y}}_j - m_j(\hat{\mathrm{x}})) \tag{2.5}$$

$$\Sigma_u^j(\mathbf{x}) = k_j(\mathbf{x}, \mathbf{x}) - K_j(\mathbf{x}, \hat{\mathrm{x}})(K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma_{v,j}^2 I)^{-1} K_j(\hat{\mathrm{x}}, \mathbf{x}). \tag{2.6}$$

The resulting GP approximation of $\psi(\mathbf{x})$ is given by

$$\psi(\mathbf{x}) \sim \mathcal{N}(\mu_u(\mathbf{x}), \Sigma_u(\mathbf{x})),$$

where

$$\mu_u(\mathbf{x}) = [\mu_u^1(\mathbf{x}), \mu_u^2(\mathbf{x}), \ldots, \mu_u^{n_u}(\mathbf{x})]^\top$$

and

$$\Sigma_u(\mathbf{x}) = \mathrm{diag}([\Sigma_u^1(\mathbf{x}), \Sigma_u^2(\mathbf{x}), \ldots, \Sigma_u^{n_u}(\mathbf{x})]).$$

The GP approximation of the obstacle's input is computed given its current state. At stage t, for each prediction time $t+k$, where $k = 1, \ldots, K$ and K is the prediction horizon, the obstacle's state and action are approximated as a joint Gaussian distribution of the form

$$\begin{bmatrix} x_o(t+k) \\ u_o(t+k) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \tilde{\mu}_x^{t,k} \\ \tilde{\mu}_u^{t,k} \end{bmatrix}, \begin{bmatrix} \tilde{\Sigma}_x^{t,k} & \tilde{\Sigma}_{xu}^{t,k} \\ \tilde{\Sigma}_{ux}^{t,k} & \tilde{\Sigma}_u^{t,k} \end{bmatrix} \right),$$

where the superscript (t,k) denotes the $(t+k)$th prediction at stage t. By the first-order Taylor expansion of (2.5) and (2.6), the mean and covariance of $u_o(t+k)$ are obtained as

$$\begin{aligned} \tilde{\mu}_u^{t,k} &= \mu_u(\tilde{\mu}_x^{t,k}) \\ \tilde{\Sigma}_u^{t,k} &= \Sigma_u(\tilde{\mu}_x^{t,k}) + \nabla\mu_u(\tilde{\mu}_x^{t,k})\tilde{\Sigma}_x^{t,k}\left(\nabla\mu_u(\tilde{\mu}_x^{t,k})\right)^\top \\ \tilde{\Sigma}_{xu}^{t,k} &= \tilde{\Sigma}_x^{t,k}(\nabla\mu_u(\tilde{\mu}_x^{t,k}))^\top. \end{aligned} \tag{2.7}$$

To propagate the obstacle's state with the new distribution information about $u_o(t+k)$, we perform the following update starting from the current state $x_o(t)$: Set $\tilde{\mu}_x^{t,0} =$

$x_o(t)$ and $\tilde{\Sigma}_x^{t,0} = \mathbf{0}$, and successively linearize ϕ around $(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k})$:

$$\begin{aligned}
\tilde{\mu}_x^{t,k+1} &= \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k}), \\
\tilde{\Sigma}_x^{t,k+1} &= \nabla_x \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k}) \tilde{\Sigma}_x^{t,k} \nabla_x \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k})^\top \\
&\quad + \nabla_u \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k}) \tilde{\Sigma}_u^{t,k} \nabla_u \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k})^\top \\
&\quad + 2\nabla_x \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k}) \tilde{\Sigma}_{xu}^{t,k} \nabla_u \phi(\tilde{\mu}_x^{t,k}, \tilde{\mu}_u^{t,k})^\top.
\end{aligned} \tag{2.8}$$

The corresponding mean and covariance of the obstacle's output $y_o(t+k)$ are computed by

$$\tilde{\mu}_y^{t,k} = C_o \tilde{\mu}_x^{t,k}, \quad \tilde{\Sigma}_y^{t,k} = C_o \tilde{\Sigma}_x^{t,k} C_o^\top. \tag{2.9}$$

As mentioned previously, we assume that we are given only the estimate of the obstacle dynamics ϕ. For comparison, we also performed GPR without the neural network approximation of the dynamics, directly inferring the states of the obstacles without predicting its control input. As shown in Figure 2.2 (a), the predicted trajectories in the early stages do not accurately follow the actual trajectory, as there is limited information from the previous observations. Furthermore, even with more data collected, it becomes impossible to predict the trajectory's curvature when there is a sudden change in the obstacle's heading angle (Figure 2.2 (b)). It is worth emphasizing that although the predictions are not entirely accurate, incorporating the neural network dynamics significantly improves the prediction accuracy compared to the case of directly predicting the states. Over time, as long as there are no sudden changes in the obstacle's behavior, the learned trajectory gradually becomes closer to the actual trajectory (Figure 2.2 (c)). This example illustrates that the prediction results of GPR are not always reliable, even when we have an estimate of the obstacle's dynamics. To guarantee safety even in such cases, we propose a distributionally robust approach, which is designed to be *proactive* to errors in learning such sudden changes.

2.3 Distributionally Robust Risk Map with Wasserstein Distance

To perform safe motion planning and control, the robot may want to estimate the risk of collision at any location in the configuration space with respect to the L obstacles. However, it is challenging to measure the risk of collision in a reliable way since the results of GPR may be inaccurate, as demonstrated in the previous section. To resolve this issue, we propose the *distributionally robust risk map*, which is a spatially varying function of the robot's current position. It estimates the conditional value-at-risk (CVaR) of collision in a distributionally robust manner using the possibly erroneous results of GPR.

2.3.1 Measuring the Risk of Collision Using CVaR

To begin, we define the *loss of safety* at each prediction time $t+k$, evaluated at t, with respect to obstacle ℓ as

$$\mathcal{J}_{t,k}(y_r, y_o^\ell) = -\|y_r(t+k) - y_o^\ell(t+k)\|_2^2. \tag{2.10}$$

It follows from (2.4) that $\mathcal{J}_{t,k}(y_r, y_o^\ell) + r_\ell^2$ is non-positive if and only if the robot navigates in the safe region $\mathcal{Y}^\ell(t+k)$. However, due to the uncertainty in the predicted $y_o^\ell(t+k)$, it may be too conservative to impose the deterministic constraint $\mathcal{J}_{t,k}(y_r, y_o^\ell) + r_\ell^2 \leq 0$.

Instead, we consider the CVaR of the loss of safety, defined by

$$\mathrm{CVaR}_\alpha^{\mathrm{P}_{t,k}^\ell}\left[\mathcal{J}_{t,k}(y_r, y_o^\ell)\right] := \min_{z\in\mathbb{R}} \mathbb{E}^{\mathrm{P}_{t,k}^\ell}\left[z + \frac{(\mathcal{J}_{t,k}(y_r, y_o^\ell) - z)^+}{1-\alpha}\right],$$

where $\mathrm{P}_{t,k}^\ell$ is the probability distribution of $y_o^\ell(t+k)$, obtained by GPR (2.9) at time t, and $(z)^+ := \max\{z, 0\}$. The CVaR of $\mathcal{J}_{t,k}(y_r, y_o^\ell)$ measures the conditional expectation of the loss within the $(1-\alpha)$ worst-case quantile as illustrated in Fig. 2.3. Thus, if $\mathrm{CVaR}_\alpha^{\mathrm{P}_{t,k}^\ell}\left[\mathcal{J}_{t,k}(y_r, y_o^\ell)\right] + r_\ell^2 \leq 0$, then the robot is located in the safe region with a probability of no less than α.

CVaR has several advantages over its popular alternative, *value-at-risk* (VaR), or, equivalently, chance constraints.[2] First, unlike VaR or chance constraints, CVaR is capable of distinguishing rare events as it takes into account the tail distribution through conditional expectation [68]. Second, CVaR is a convex risk measure unlike VaR and thus is more computationally tractable than VaR for general probability distributions [109]. Third, as opposed to VaR, CVaR is *coherent* in the sense of Artzner *et al.* [110] and is advocated as a rational risk measure in robotics applications [69]. Thus, CVaR has recently received a considerable attention in the robotics community [111–113].

In practice, it is unlikely that we can accurately compute the CVaR of the loss of safety since $\mathrm{P}_{t,k}^{\ell}$ obtained by GPR is imperfect. To handle such distribution errors, we propose using the following distributionally robust version of CVaR:

$$\text{DR-CVaR}_{\alpha,\theta}\big[\mathcal{J}_{t,k}(y_r, y_o^{\ell})\big] := \sup_{\mathrm{Q}_{t,k}^{\ell} \in \mathbb{D}_{t,k}^{\ell}} \mathrm{CVaR}_{\alpha}^{\mathrm{Q}_{t,k}^{\ell}}\big[\mathcal{J}_{t,k}(y_r, y_o^{\ell})\big], \tag{2.11}$$

which measures the risk of unsafety for the worst-case distribution in an ambiguity set $\mathbb{D}_{t,k}^{\ell}$. We consider the *Wasserstein ambiguity set*, constructed as a ball with radius $\theta_{t,k} > 0$ around the nominal distribution $\mathrm{P}_{t,k}^{\ell}$, obtained by GPR, i.e.,

$$\mathbb{D}_{t,k}^{\ell} := \{\mathrm{Q} \in \mathcal{P}(\mathbb{R}^{n_y}) \mid W_2(\mathrm{Q}, \mathrm{P}_{t,k}^{\ell}) \leq \theta_{t,k}\}, \tag{2.12}$$

where $W_2(\mathrm{Q}, \mathrm{P}_{t,k}^{\ell})$ is the 2-Wasserstein distance between Q and $\mathrm{P}_{t,k}^{\ell}$. The p-Wasserstein metric $W_p(\mathrm{Q}, \mathrm{P})$ between two distributions Q and P supported on $\Xi \subseteq \mathbb{R}^m$ is defined as

$$W_p(\mathrm{Q}, \mathrm{P}) := \left[\min_{\kappa \in \mathcal{P}(\Xi^2)} \left\{ \int_{\Xi^2} \|y - y'\|^p \, \mathrm{d}\kappa(y, y') \mid \Pi^1\kappa = \mathrm{Q}, \Pi^2\kappa = \mathrm{P} \right\}\right]^{1/p},$$

where κ is the transportation plan, the ith marginal of which is denoted by $\Pi^i\kappa$. It represents the minimum cost for transporting mass from Q to P using *non-uniform*

[2]The VaR of a real-valued random variable X is defined as $\mathrm{VaR}_{\alpha}(X) := \inf\{x \in \mathbb{R} \mid F_X(x) \geq \alpha\}$, where F_X is the cumulative distribution function of X. Thus, the VaR constraint $\mathrm{VaR}_{\alpha}(X) \leq \delta$ is equivalent to the chance constraint $\mathrm{Prob}\{X \leq \delta\} \geq \alpha$. Furthermore, $\mathrm{VaR}_{\alpha}(X) \leq \mathrm{CVaR}_{\alpha}(X)$ as shown in Fig. 2.3.

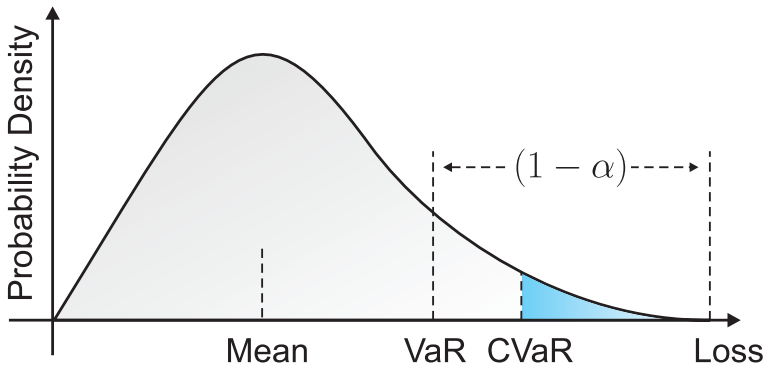


Figure 2.3: Conditional value-at-risk of a random loss.

perturbations with the cost of moving a unit mass from y to y' prescribed by $\|y - y'\|^p$, where $\|\cdot\|$ is a norm on $\mathbb{R}^m$.

It is worth emphasizing that the role of radius $\theta_{t,k}$ is different from that of α. As $\mathrm{CVaR}_\alpha(X)$ considers the conditional expectation of X over the worst-case $(1-\alpha)$ quantile, α is able to correctly control the conservativeness of $\mathrm{CVaR}_\alpha(X)$ only when the probability distribution of X is known precisely. However, this is no longer valid when the probability distribution is inaccurate. The distributionally robust risk aims to tackle this issue by enhancing robustness against distribution errors. The radius $\theta_{t,k}$ controls the size of allowable distribution errors, unlike α. As observed in our experiments, even with a large α, CVaR is insufficient for ensuring safety in learning-enabled environments since it is unable to anticipate distributional uncertainties such as learning errors in GPR (refer to the last part of the supplementary video clip).

The Wasserstein metric is also known as the *earth mover's distance*, as it can be interpreted as the minimum cost of turning one pile of earth into another, where each distribution is viewed as a unit amount of earth. The Wasserstein ambiguity sets have several advantages over other types of ambiguity sets. First, the Wasserstein metric incorporates a notion of how close two points in the support are to each other unlike, for example, phi-divergences. Thus, Wasserstein DRO problems avoid providing

unreasonable pathological solutions [40]. Second, the Wasserstein ambiguity sets provide a powerful finite sample guarantee for empirical nominal distributions, and this feature is useful in sequential decision-making problems [30, 39]. Third, Wasserstein DRO is strongly related to the regularization techniques in machine learning and can be applied to alleviate overfitting [41].

Concerning all the obstacles, we define the *distributionally robust risk map* (DR-risk map) $\mathcal{R}_{t,k} : \mathbb{R}^{n_y} \to \mathbb{R}$ for prediction time $t+k$, evaluated at t, as

$$\mathcal{R}_{t,k}(y_r) := \max_{\ell=1,\dots,L} \mathcal{R}^{\ell}_{t,k}(y_r, \mathcal{Y}^{\ell}), \tag{2.13}$$

where

$$\mathcal{R}^{\ell}_{t,k}(y_r, \mathcal{Y}^{\ell}) := \Big(\text{DR-CVaR}_{\alpha,\theta}\big[\mathcal{J}_{t,k}(y_r, y_o^{\ell})\big] + r_{\ell}^2\Big)^{+}. \tag{2.14}$$

The DR-risk map returns the maximum risk for all obstacles. Its value is zero if there is no risk; otherwise, its value is positive. In our safe motion planning and control methods, the following constraint is used to limit the risk of collision:

$$\mathcal{R}_{t,k}(y_r) \leq \delta, \tag{2.15}$$

where $\delta \geq 0$ is a risk tolerance parameter.

It is important to note that the computational complexity of the risk map increases linearly with the number of obstacles due to the maximum operator involved in its computation. However, as the number of obstacles increases, the feasible region of the robot that satisfies the constraint (2.15) shrinks, making it harder to find feasible solutions.

2.3.2 Semidefinite Programming Formulation

Unfortunately, it is nontrivial to directly compute the DR-risk map $\mathcal{R}_{t,k}(y_r)$ or its proxy $\text{DR-CVaR}_{\alpha,\theta}\big[\mathcal{J}_{t,k}(y_r, y_o^{\ell})\big]$ as this involves an infinite-dimensional optimization problem over the set of probability distributions. We reformulate it as a finite-dimensional problem by exploiting some structural properties of CVaR and Wasser-

stein distance. The following theorem presents the result of reformulation as a semidefinite program (SDP), where the dependence on t, k and ℓ is encoded solely in $y_r(t+k), \tilde{\mu}_y^{t,k,\ell}$ and $\tilde{\Sigma}_y^{t,k,\ell}$. Later, this feature will allow us to approximate the risk map by a single NN, independent of t, k and ℓ.

Theorem 2.1. *Let $\mathrm{P}_{t,k}^{\ell}$ be the distribution of $y_o^{\ell}(t+k)$ with mean $\tilde{\mu}_y^{t,k,\ell}$ and covariance $\tilde{\Sigma}_y^{t,k,\ell}$, estimated by GPR. Then, the DR-CVaR* (2.11) *has the following upper-bound:*

$$
\begin{aligned}
\min\ & z + \frac{\tau + \varepsilon + \mathrm{Tr}[Z] + \lambda\big(\theta_{t,k}^2 - \|\tilde{\mu}_y^{t,k,\ell}\|_2^2 - \mathrm{Tr}[\tilde{\Sigma}_y^{t,k,\ell}]\big)}{1-\alpha} \\
\text{s.t.}\ & \begin{bmatrix} \lambda I - \Gamma & \gamma + \lambda\tilde{\mu}_y^{t,k,\ell} \\ \big(\gamma + \lambda\tilde{\mu}_y^{t,k,\ell}\big)^\top & \varepsilon \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} \lambda I - \Gamma & \lambda\big(\tilde{\Sigma}_y^{t,k,\ell}\big)^{1/2} \\ \lambda\big(\tilde{\Sigma}_y^{t,k,\ell}\big)^{1/2} & Z \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} \Gamma + I & \gamma - y_r(t+k) \\ \big(\gamma - y_r(t+k)\big)^\top & \tau + z + \|y_r(t+k)\|_2^2 \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} \Gamma & \gamma \\ \gamma^\top & \tau \end{bmatrix} \succeq 0 \\
& \lambda \in \mathbb{R}_+,\ z \in \mathbb{R},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R}^{n_y} \\
& \Gamma \in \mathbb{S}^{n_y},\ \varepsilon \in \mathbb{R}_+,\ Z \in \mathbb{S}_+^{n_y}.
\end{aligned}
\tag{2.16}
$$

Its proof is contained in Appendix 2.8.2. The SDP problem (2.16) can be solved using well-known algorithms, such as interior-point methods [114–116], splitting methods [117], augmented Lagrangian methods [118], etc. Its dual problem is more of an interest, as it involves fewer generalized inequalities.

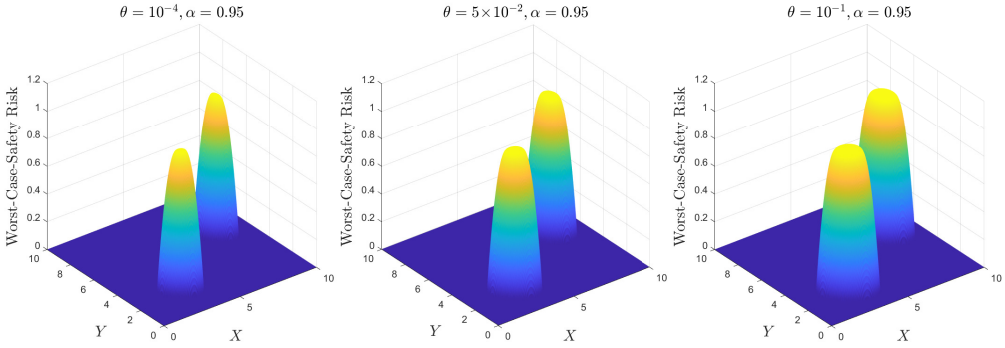


Figure 2.4: Risk maps for two obstacles with means $\tilde{\mu}_y^{t,k,1} = (3, 2.5)$, $\tilde{\mu}_y^{t,k,2} = (8, 6)$ and covariances $\tilde{\Sigma}_y^{t,k,1} = \mathrm{diag}[0.003, 0.002]$, $\tilde{\Sigma}_y^{t,k,2} = \mathrm{diag}[0.001, 0.004]$ for $\theta = \{10^{-4}, 5 \times 10^{-2}, 10^{-1}\}$ and $\alpha = 0.95$.

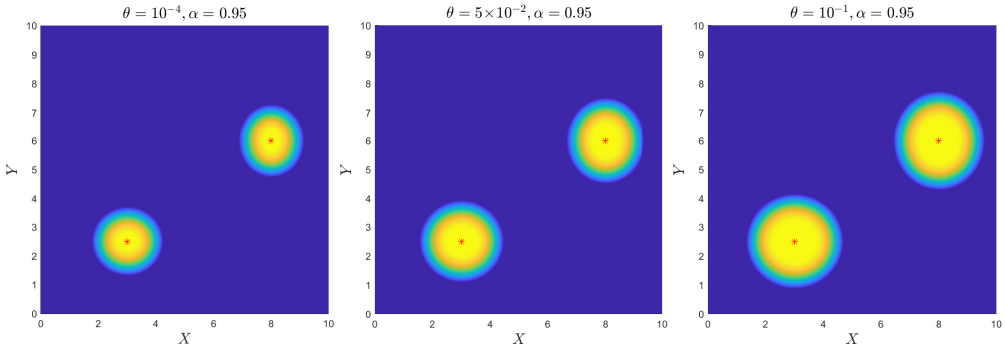


Figure 2.5: Projection of the risk maps onto the robot's configuration space.

Corollary 2.1. *The dual problem of* (2.16) *can be expressed as the following SDP:*

$$
\begin{aligned}
&\max\ 2W_{12}^\top y_r(t+k) - \mathrm{Tr}[W_{11}] - \|y_r(t+k)\|_2^2 \\
&\text{s.t.}\ \ \frac{\theta_{t,k}^2 - \|\tilde{\mu}_y^{t,k,\ell}\|_2^2 - \mathrm{Tr}[\tilde{\Sigma}_y^{t,k,\ell}]}{1-\alpha} - 2X_{12}^\top \tilde{\mu}_y^{t,k,\ell} \\
&\qquad - \mathrm{Tr}[X_{11} + Y_{11} + 2Y_{12}^\top \left(\tilde{\Sigma}_y^{t,k,\ell}\right)^{1/2}] \geq 0 \\
&\qquad X_{11} + Y_{11} = W_{11} + V_{11} \\
&\qquad X_{12} + W_{12} + V_{12} = 0 \\
&\qquad W_{22} = 1,\ V_{22} = \frac{1}{1-\alpha} - 1 \\
&\qquad X_{22} \leq \frac{1}{1-\alpha},\ Y_{22} \preceq \frac{1}{1-\alpha} I \\
&\qquad Y \in \mathbb{S}_+^{2n_y},\ X, W, V \in \mathbb{S}_+^{n_y+1}.
\end{aligned}
\tag{2.17}
$$

Furthermore, the duality gap is zero.

Its proof can be found in Appendix 2.8.2. The dual problem is also a tractable SDP problem, which can be solved using the same algorithms as for the primal. However, the dual problem (2.17) has less linear matrix inequality constraint in addition to a number of linear equality and inequality constraints, which are easier to handle for most of the off-the-shelf solvers than the positive semidefinite constraints in the primal problem (2.16). The dual problem is useful in some cases the SDP solver might fail to solve (2.16) due to numerical issues. We can use the solution to the dual problem if there is no primal solution returned by the solver.

2.3.3 Example of DR-Risk Maps

By discretizing the robot's configuration space and solving either (2.16) or (2.17) for all discretized points, we can construct the desired DR-risk map (2.13). Fig. 2.4 shows examples of such risk maps, which are obtained by solving the primal problem for a risk confidence level $\alpha = 0.95$ with two obstacles ($L = 2$) at stage $t + k$. In the shown risk maps, the estimated means and covariances for two obstacles' CoMs are set to $\tilde{\mu}_y^{t,k,1} = [3, 2.5], \tilde{\mu}_y^{t,k,2} = [8, 6]$ and $\tilde{\Sigma}_y^{t,k,1} = \mathrm{diag}[0.003, 0.002], \tilde{\Sigma}_y^{t,k,2} = \mathrm{diag}[0.001, 0.004]$, respectively. Each peak of the risk map is located at the worst-case mean of each obstacle's CoM with a value of $r_\ell^2 = 1$. The risk diminishes as the robot moves away from the obstacle. Fig. 2.4 demonstrates that the non-zero area of the risk map expands as the radius θ increases. Also, the peak area for a bigger radius becomes flatter, meaning that more regions are considered "risky". Therefore, the robot's decision using this map will be more robust against errors in the estimated distribution as the Wasserstein ambiguity set gets larger.

Fig. 2.5 shows the projection of the risk map onto the robot's configuration space. It shows that a bigger θ generates a more conservative risk map. The risky area enlarges with the size of our ambiguity set. The computation time for constructing the DR-risk map with the two obstacles is reported in Table 2.1. Here, the SDP problem is

Table 2.1: Computation time for constructing the DR-risk map with $L = 2$ obstacles, averaged over 40,000 positions of the robot.

Radius θ	10^{-4}	10^{-2}	5×10^{-2}	10^{-1}
Computation Time (ms)	8.6 ± 0.26	4.6 ± 0.53	3.8 ± 0.05	3.8 ± 0.05

solved for each obstacle separately using a conic solver, called MOSEK [119]. Even though the computation slows down near the obstacles, the overall computation time is relatively small for all θ's.

For an efficient construction of the risk map, we propose an NN approximation in Section 2.5.1. The NN approach avoids any discretization of the robot's configuration space or training of multiple networks for different t, k, and ℓ because such dependence is encoded in $y_r(t+k)$, $\tilde{\mu}_y^{t,k,\ell}$ and $\tilde{\Sigma}_y^{t,k,\ell}$ as previously mentioned. In the following two sections, we present applications of the DR-risk map to safe motion planning and control in learning-enabled environments.

2.3.4 Probabilistic Guarantee on the Loss of Safety

An advantage of using the Wasserstein ambiguity sets in DRO is that one can obtain a non-asymptotic probabilistic performance guarantee. For example, it is shown in [41] that Wasserstein DRO provides an out-of-sample performance guarantee when the nominal distribution is chosen as an empirical distribution. However, we consider the case where the nominal distribution is obtained by GPR. We show that the DR-risk map (2.13) provides a probabilistic guarantee on the true loss of safety (2.10) under the following assumption on GPR result.

Assumption 2.1. *Let $\mathbb{P}$ denote the probability measure for the GP dataset $\mathcal{D} = \{(\hat{x}^i, \hat{y}_j^i)\}_{i=1}^M$, and let $\tilde{\mu}_{y,\mathcal{D}}^{t,k}$ and $\tilde{\Sigma}_{y,\mathcal{D}}^{t,k}$ denote the mean and the covariance matrix of $y_o(t+k)$ obtained from GPR performed at time t using $\mathcal{D}$. We assume that there exist*

a non-negative constant $\omega_{\mathcal{D}}^{t,k}$ *and an* $n_y \times n_y$ *positive semidefinite matrix* $\Omega_{\mathcal{D}}^{t,k}$ *such that for some* $p \in (0,1)$ *the following probabilistic error bound holds:*

$$\mathbb{P}\Big\{\mathcal{D} \mid \|y_o(t+k) - \tilde{\mu}_{y,\mathcal{D}}^{t,k}\| \leq \omega_{\mathcal{D}}^{t,k}\Big\} \geq (1-p)^k,$$

and $\tilde{\Sigma}_{y,\mathcal{D}}^{t,k} \preceq \Omega_{\mathcal{D}}^{t,k}$ *with probability 1.*

Assumption 2.1 represents a probabilistic bound on the GPR result evaluated viewing the dataset $\mathcal{D}$ as a random variable. This performance requirement for GPR can be satisfied via a probabilistic uniform error bound for GPR under some mild conditions such as the Lipschitz continuity of $\psi(\cdot)$ [120, 121].[3] We now establish a probabilistic guarantee on the loss of safety in the following theorem:

Theorem 2.2. *Suppose that Assumption 2.1 is satisfied. Consider the Wasserstein ambiguity set* $\mathbb{D}_{t,k}$ *with time-varying radius* $\theta_{t,k}$*. If the radius satisfies the following condition*

$$\theta_{t,k}^2 \geq (\omega_{\mathcal{D}}^{t,k})^2 + \mathrm{Tr}[\Omega_{\mathcal{D}}^{t,k}], \tag{2.18}$$

then the DR-risk map $\mathcal{R}_{t,k}^{\mathcal{D}}$ *constructed using the GP dataset* $\mathcal{D}$ *provides the following probabilistic guarantee on the loss of safety* (2.10)*:*

$$\mathbb{P}\left\{\mathcal{D} \,\middle|\, \mathcal{J}_{t,k}(y_r, y_o) \leq \mathcal{R}_{t,k}^{\mathcal{D}}(y_r) - r^2\right\} \geq (1-p)^k.$$

Thus, the probabilistic bound holds for any ambiguity set $\mathbb{D}_{t,k}$ *with a time-invariant radius* $\theta \geq \max_{t,k} \theta_{t,k}$*.*

Its proof can be found in Appendix 2.8.2. Theorem 2.2 confirms that the DR-risk map is capable of dealing with errors in the GPR results, often occurring due to sudden changes in the obstacle's motion pattern.[4] The theorem considers a time-varying

[3]Note, that the uniform error bound for GPR can only be derived when the obstacle dynamics ϕ are known. However, when using a neural network approximation, an additional analysis of the bound is required.

[4]Theorem 2.2 holds for each obstacle. The extension to a similar probabilistic guarantee on the joint loss of safety for all obstacles is straightforward under the assumption that the GP datasets for all obstacles are independent. Then, the guarantee on the loss of safety holds with a probability of $(1-p)^{Lk}$.

Wasserstein radius $\theta_{t,k}$ computed depending on the error in the estimated obstacle's position. As a result, the distributional robustness of our risk map is adaptively adjusted according to the currently available data. At each time t, with zero prediction horizon $k = 0$, $\theta_{t,0}$ can be set to 0 and thus the ambiguity set is a singleton that contains only the current position $y_o(t)$ of the obstacle. As we predict further, i.e., $k > 0$, the GP error bound grows, resulting in a larger radius $\theta_{t,k}$ determined using (2.18). This adaptive construction of the ambiguity sets suggests a way to adjust $\theta_{t,k}$ depending on inaccuracies in the GPR results. However, the GP error bound in Assumption 2.1 is often loose, limiting the practical use of (2.18). If that is the case, the radius can be calibrated using the collision probability as presented in Section 2.6.

2.4 Application to Learning-Based Distributionally Robust Motion Planning

As the first application of the DR-risk maps, we propose a learning-based motion planning algorithm based on RRT* [122]. Unlike previous RRT algorithms, our algorithm takes into account possible errors in the learned distribution of the obstacles' behaviors.

2.4.1 Main Algorithm

The motion planning algorithm presented in this section is an online sampling-based algorithm for computing a path from the robot's starting point to the goal point in near real-time, taking into account moving obstacles. The overall algorithm, similar to the original RRT* algorithm, consists of the nearest neighbor search, steering towards the sampled node, safety check, and rewiring. Inspired by [123], the path is generated only for a given time, after which the robot executes the committed trajectory and restarts the planning process from a new initial state, removing unreachable nodes from the tree. The key extension to the original algorithm is the use of the DR-risk map for

safety checks. In addition, the algorithm leverages GPR to infer the future trajectories of the obstacles based on either the system dynamics (2.3) or its approximation (2.28). The risk map in (2.13) is employed to guarantee the safety of the derived paths in two stages. First, each node computed in the growing stage of the tree is classified as either safe or unsafe based on the risk value to later include it in or exclude it from the safe subtree. Second, the cost function of planning includes the risk value to escape possibly unsafe nodes. Steering towards a sampled node is performed according to the given robot's dynamics (2.1) by applying controls that satisfy the input constraints. Moreover, when changing the parent from one node to another, the feasibility of the trajectories and control inputs are checked once again to meet the given requirements.

Our learning-based distributionally robust RRT* (DR-RRT*) algorithm is presented in Algorithm 1, given goal state q_{goal}, maximum depth K, risk weight constant w, other hyper-parameters θ, α and r_ℓ for computing risk, as well as the radius r_{RRT} for neighborhood construction, computed as in [122, Theorem 38].

At the beginning of the algorithm, $\mathcal{T}$ is set as an empty tree to be expanded later. Initially, the GP dataset D^ℓ is also an empty set. In each iteration, a new safe subtree $\mathcal{T}_{\text{safe}}$ is defined (Line 5). Then, the robot's state $x_r(t)$ as well as the obstacle's state and action $x_o^\ell(t)$ and $u_o^\ell(t)$ are observed at current stage t, as performed in Line 6. Thereafter, the tree is constructed with $x_r(t)$ as the root (Line 7). Since there might be some nodes that are unreachable from the current state, we remove the corresponding edges and vertices in Line 8. These nodes are all nodes that do not root from the current state $x_r(t)$. In Line 9, the pruned tree is updated with a new depth value starting from the root, the depth of which is set to $k = 0$.

Having new perceived information about the obstacles' motions, we perform GPR in Line 10–16. Here, the GP dataset is updated with new observations, after which the GP approximation of $\psi^\ell(\mathbf{x})$ is updated by learning mean and covariance functions $\mu_u^{\ell,j}(\mathbf{x})$ and $\Sigma_u^{\ell,j}(\mathbf{x})$ as in (2.5) and (2.6). To predict the trajectory of each obstacle starting from $t+1$ to $t+K$, the mean and covariance at t are initialized as the current

observation and the zero covariance matrix, respectively. In Line 15–16, the mean and the covariance of the obstacle's action, state and output are computed by (2.7), (2.8) and (2.9). Here, K corresponds to the desired time horizon or, equivalently, the maximum depth of the path.

Using the new prediction results, the safe tree updated in Line 17–22 using the nodes of $\mathcal{T}$ satisfying the risk constraint $\mathcal{R}_{t,k}(C_r q) \leq \delta$ with depth less than threshold K. This is accomplished by calculating the DR-risk $\mathcal{R}_{t,k}(C_r q)$ for all nodes according to (2.13), where the SDP problem (2.16) or its dual (2.17) needs to be solved for each obstacle. Here, k corresponds to the depth of the node, and therefore the predictions of step k are used to compute the risk for a node of depth k. The new value of risk is used to update the costs for the corresponding nodes. Next, in Line 23–24, we proceed to the expansion of the tree $\mathcal{T}$ for some fixed time τ, where $\mathcal{T}_{\text{safe}}$ is also updated with new nodes. The details of the tree expansion are given in Algorithm 2 and Section 2.4.2.

When the planning time is over, the best partial path is retrieved and passed to execution, being constructed from the root of the safe tree towards the goal (Line 25), where the current state corresponds to q_0.[5] The robot follows the path for one step by driving it towards the next state q_1 in the planned path (Line 26). The algorithm continues until the distance between the tree root (the current robot state) and the desired q_{goal} is no greater than tolerance ϵ.

For real-time execution of the algorithm, it is necessary for the robot to operate while the planning is being performed. This can be achieved by executing Line 26 in parallel with the remaining parts of the algorithm. To ensure the termination of the algorithm, the tree will be grown until the planning time reaches T_s seconds.

[5] All paths returned by the algorithm are feasible since the safe tree is constructed only from the feasible nodes satisfying the safety condition.

2.4.2 Tree Expansion and Rewiring

The tree expansion and rewiring algorithm is given in Algorithm 2. Similar to the classical RRT*, the tree is expanded by randomly choosing a point in the configuration space (Line 2). Then, in Line 3 the node to be extended is chosen as the minimizer of

$$c(q, q_{\mathrm{rand}}) = c(q) + \mathcal{L}(q, q_{\mathrm{rand}}),$$

where $\mathcal{L}(q, q_{\mathrm{rand}})$ is the length of the path from q to q_{rand} and $c(q)$ is the cost of node q, defined as

$$c(q) = c(\mathrm{Parent}(q)) + w\mathcal{R}_{t,k}(C_r q) + \mathcal{L}(\mathrm{Parent}(q), q). \tag{2.19}$$

The worst-case risk is taken into account in $c(q)$, where the SDP problem is solved for $(t+k)$th prediction performed at current stage t with k being the depth of node q.

In Line 4 the depth k for the new node is set to the depth of the nearest node incremented by 1 for computing risk in the next step. The new node q_{new} is obtained in Line 5 by steering the chosen best node towards q_{rand}. Here, the control input is chosen as the one with the least cost $c(q_{\mathrm{new}})$. The safety risk is given by (2.13) and computed by solving the SDP (2.16) or its dual (2.17) for all $\ell = 1, \ldots, L$.

In Line 6, the neighborhood of q_{new} is constructed from the nodes in safe subtree $\mathcal{T}_{\mathrm{safe}}$ with distance less than r_{RRT} to q_{new}. The best parent of q_{new} is chosen in Lines 7–12. The parent is initialized as q_{nearest}. However, this is changed if the cost to q_{new} via q_{near} is less than the cost via q_{nearest} and the new path is feasible. The node q_{new} with the updated parent is added to the tree in Line 13 only after selecting the parent. The subtree $\mathcal{T}_{\mathrm{safe}}$ is also updated if the risk of the node q_{new} with depth k is less than the threshold δ (Line 16–17).

Similar to the original RRT* algorithm, the rewiring of the neighborhood nodes is performed in Line 18–26 after the process of growing the tree is completed. For all q_{near} in $\mathcal{N}_{\mathrm{near}}$, the cost is calculated taking q_{new} as parent. If the new cost is less than the existing one and the path is feasible, the parent of q_{near} in both $\mathcal{T}$ and $\mathcal{T}_{\mathrm{safe}}$ is

changed to q_{new}. The costs for q_{near} as well as its children nodes are updated to take into account the cost for q_{new}. Unlike the original RRT* algorithm, in Line 25–26 we also update the safe subtree, where the edge from q_{new} to q_{near} is added if the new depth is less than K. Otherwise, q_{near} is removed from the subtree to keep the safe subtree within the maximum depth K.

2.4.3 Graphical Illustration

A step-by-step example of our algorithm is illustrated in Fig. 2.6, where the blue ball represents an obstacle centered at the predicted mean at time steps $k = 0, \ldots, K$ with $K = 5$. In Fig. 2.6-I, the robot is steered from the old root to the new root. Thus, the part of the tree not growing from the new root is pruned. The vertices in orange with depth 3 and 4 have positive risks with respect to the predicted obstacle's location for $k = 3$ and $k = 4$, respectively. Hence, these nodes are not included in the safe subtree $\mathcal{T}_{\text{safe}}$. In Fig. 2.6-II, q_{rand} is sampled in the configuration space and the corresponding q_{nearest} is selected from $\mathcal{T}_{\text{safe}}$ with the lowest cost. A q_{new} is found by steering q_{nearest} towards q_{rand}. In Fig 2.6-III, a ball of neighbors for q_{new} is created (in orange). This ball includes nodes in green as well as q_{nearest}. The current lowest cost is set to the cost from the root of the tree to q_{new} via q_{nearest}.

In Fig 2.6-IV, the costs to q_{new} via other neighboring nodes are computed. It is observed that the length to q_{new} and the risk are larger via other neighbors than via q_{nearest}. This is because the depth of q_{new} becomes 5 and the risk is computed for obstacles at $k = 5$, whereas when q_{nearest} is the parent, the depth of q_{new} is 4 and the obstacle is farther from the node. Therefore, in Fig 2.6-V, the parent of q_{new} is chosen as q_{nearest}. Also, q_{new} is added to the safe subtree since the risk is non-positive. Fig. 2.6-VI illustrates the rewiring process, where the cost for the neighbor node improves when its parent is q_{new}.

Our motion planning method is a learning-based algorithm based on CC-RRT* [124], another real-time algorithm for probabilistically feasible motion planning built upon

the chance-constrained RRT (CC-RRT) algorithm [125] and the original RRT* [122]. Unlike CC-RRT*, our algorithm first learns the distribution of the obstacles' future trajectories from new observations and replaces the probability of collision with the distributionally robust risk map defined in (2.13). Then, instead of chance constraints, the DR-risk map is used as a constraint to ensure safety as well as to penalize possibly risky trajectories in the cost function. It is well known that CVaR constraints induce more conservative behaviors compared to chance constraints. Moreover, our DR-risk map yields to take into account possible errors in the learned distribution of the obstacles' behaviors that in practice cannot be captured by CC-RRT*. As an extension to CC-RRT, distributionally robust RRT (DR-RRT) is introduced in [72], where a moment-based ambiguity set is used, unlike our algorithm. The resulting deterministic constraint is similar to the one in CC-RRT* with the difference that it leads to a stronger constraint tightening. On the contrary, our DR-RRT* uses CVaR constraints in addition to the Wasserstein ambiguity set, which inherently takes into account moment ambiguity, thereby providing an additional layer of robustness as mentioned in Appendix 2.8.2. Furthermore, it is worth mentioning that most motion planning algorithms work only for a restricted set of problems. For example, in both CC-RRT* and DR-RRT, the region occupied by obstacles should be represented by a convex polytope with uncertainties in translation, while in both Risk-RRT* [106] and Risk-Informed-RRT* [126] the risk map is constructed as a grid by discretizing the state space. On the contrary, our method does not impose such restrictions, allowing any obstacle of an arbitrary shape and motion as long as the loss can be constructed as a piecewise quadratic function.

2.5 Application to Learning-Based Distributionally Robust Motion Control

In addition to motion planning, our DR-risk map can be used for motion control in risky environments. As the second application, we propose a learning-based motion control technique that limits the risk of collision in a distributionally robust way. In this case, our motion controller determines a control input that is robust against errors in learned information about the obstacles' movements.

We formulate the motion control problem as the following MPC problem with DR-risk constraints:

$$\min_{\mathbf{u},\mathbf{x},\mathbf{y}} \quad J(x_r(t), \mathbf{u}) := \sum_{k=0}^{K-1} c(y_k, u_k) + q(y_K) \tag{2.20a}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \tag{2.20b}$$

$$y_k = Cx_k \tag{2.20c}$$

$$x_0 = x_r(t) \tag{2.20d}$$

$$\mathcal{R}_{t,k}(y_k) \leq \delta \tag{2.20e}$$

$$x_k \in \mathcal{X} \tag{2.20f}$$

$$u_k \in \mathcal{U} \tag{2.20g}$$

where $\mathbf{x} := (x_0, \ldots, x_K)$, $\mathbf{u} := (u_0, \ldots, u_{K-1})$, $\mathbf{y} := (y_0, \ldots, y_K)$ are the robot's predicted state, input and output trajectories over the prediction horizon K. The constraints (2.20b) and (2.20g) should be satisfied for $k = 0, \ldots, K-1$, the constraint (2.20c) should hold for $k = 0, \ldots, K$, and the constraints (2.20e) and (2.20f) are imposed for $k = 1, \ldots, K$. Here, the stage-wise cost function $c : \mathbb{R}^{n_y} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and the terminal cost function $q : \mathbb{R}^{n_y} \to \mathbb{R}$ are chosen to penalize the deviation from the reference trajectory y^{ref} and to minimize the control effort as follows:

$$c(y_k, u_k) = \|Q(y_k - y_k^{ref}\|_2^2 + \|Ru_k\|_2^2$$
$$q(y_K) = \|Q_f(y_K - y_K^{ref})\|_2^2,$$

where $Q, Q_f, R \succ 0$ are the state and control weight matrices. The sets $\mathcal{X}$ and $\mathcal{U}$ represent the state and input constraint sets, respectively, which are assumed to be polyhedra for simplicity.

The constraint (2.20e) integrates the risk map into the controller synthesis by limiting the DR-risk (2.11) to user-specified tolerance level δ. When $f(x_k, u_k)$ is a linear function, the DR-MPC problem can be reformulated into a bi-linear SDP by writing the risk constraint in the SDP form (2.16). However, solving such a problem is a computationally expensive task. To alleviate the computational issue, we propose to approximate the DR-risk map by an NN that can be trained offline.

2.5.1 Neural Network Approximation of DR-Risk Map

Consider the feed-forward NN in Fig. 2.7 with $\mathcal{J}$ layers and $\mathcal{N}_i$ nodes in each layer with a ReLU activation function. The inputs of the NN are the robot's position $y_r(t+k)$ and the parameters of the predicted distribution of the obstacles' behaviors, $\tilde{\mu}_y^{t,k,\ell}$ and $(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}$, while the target is the solution of the SDP problem (2.16).[6]

For any position of the robot and the predicted position of the obstacle, the risk map computed in (2.14) can be approximated using the NN as

$$\mathcal{R}_{NN}^{t,k,\ell}(y_r, \mathcal{Y}^\ell; \theta, \alpha) = \left(a_{\mathcal{J}}^{k,\ell} + r_\ell^2\right)^+, \tag{2.21}$$

where

$$h_i^{k,\ell} = \max\{0, a_i^{k,\ell}\},\ i = 1, \ldots, \mathcal{J} - 1 \tag{2.22}$$

$$a_i^{k,\ell} = W_i h_{i-1}^{k,\ell} + b_i,\ i = 1, \ldots, \mathcal{J}. \tag{2.23}$$

Here, $W_i \in \mathbb{R}^{\mathcal{N}_i \times \mathcal{N}_{i-1}}$ and $b_i \in \mathbb{R}^{\mathcal{N}_i}$ are the weight and bias, $h_i^{k,\ell} \in \mathbb{R}^{\mathcal{N}_i}$ and $a_i^{k,\ell} \in \mathbb{R}^{\mathcal{N}_i}$ are the output and activation of the ith layer with $h_0^{k,\ell} \in \mathbb{R}^{\mathcal{N}_0}$ being the input of the network with $\mathcal{N}_0 = n_y(n_y + 5)/2$. The activation function in (2.22) follows from the definition of ReLU. The input of the network is constructed from the robot's position $y_r(t + k) \in \mathbb{R}^{n_y}$ and the parameters of the predicted distribution of the obstacles' behaviors $\tilde{\mu}_y^{t,k,\ell}$ and $(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}$ as follows:

$$h_0^{k,\ell} = \left[y_r(t+k)^\top, (\tilde{\mu}_y^{t,k,\ell})^\top, \mathrm{vech}\left[(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}\right]^\top\right]^\top,$$

[6] The architecture in Fig. 2.7 assumes fixed θ and α. However, these parameters may also be added as additional input variables to the NN.

where $\mathrm{vech}[\cdot]$ is an operator vectorizing the lower triangular elements of the matrix. Note that the NN is independent of t, k, and ℓ since the dependence is encoded in the input information. Therefore, we can use a single NN to approximate the DR-risk maps for all t, k, and ℓ. Moreover, like the exact risk map in (2.13), the approximate risk map does not require additional information such as the true positions of the obstacles. Real-time behaviors are captured through the inputs of the NN, namely the robot's position and the probability distribution of the obstacles' positions inferred via GPR. This feature is inherited from our distributionally robust formulation that focuses on the worst-case distribution determined not by the current obstacle configuration but by the learned distribution.

To train the NN, a dataset is created by solving (2.16) for different values of $y_r(t+k), \tilde{\mu}_y^{t,k,\ell}$ and $(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}$ for fixed θ and α. Thereafter, the NN is trained via backpropagation to approximate the DR-risk map. As an example, the mean squared error (MSE) and mean average error (MAE) for all training, validation, and test samples are reported in Table 2.2, showing that both errors are small.

To validate this approach, we compare the DR-risk map and its NN approximation computed using 50,000 random realizations of y_k, $\tilde{\mu}_y^{t,k,\ell} \sim \mathcal{U}[0,10]^2$ and $\tilde{\Sigma}_y^{t,k,\ell} \sim \mathcal{U}[0,0.7]^3$ for $\theta = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}$ and $\alpha = 0.95$. We also randomly generate the radius $r_\ell \sim \mathcal{U}[0,0.2]$ and the risk tolerance level $\delta_\ell \sim \mathcal{U}[0,0.5r_\ell^2]$ to show the flexibility of our approximation method. As shown in Table 2.3, the probability that the approximate risk map reports safe events as unsafe is quite small. Furthermore, the approximate risk map is not so conservative since the probability of misreporting unsafe events as safe is also small. These results show the validity of our NN approximation approach.

2.5.2 Approximate Distributionally Robust MPC

Using the NN approximation of the DR-risk map, we eliminate the need to solve the optimization problem (2.16) in the constraints of the MPC problem (2.20). Moreover,

since only the inputs $y_r(t+k)$, $\tilde{\mu}_y^{t,k,\ell}$ and $\tilde{\Sigma}_y^{t,k,\ell}$ of the NN depend on the t, k and ℓ, the same NN can be used for all time stages and obstacles, by simply providing appropriate inputs to the NN. Therefore, the use of our NN approximation significantly reduces the computational burden required to solve the MPC problem. More specifically, we obtain the following approximate MPC problem.

Proposition 2.1. *Suppose that the NN approximation* (2.21) *of the DR-risk map is given for fixed parameters θ and α. If the risk map in* (2.20e) *is replaced with the NN approximation, the DR-MPC problem* (2.20) *can be expressed as follows:*

$$\min \quad J(x_r(t), \mathbf{u}) := \sum_{k=0}^{K-1} c(y_k, u_k) + q(y_K) \tag{2.24a}$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \tag{2.24b}$$

$$y_k = Cx_k \tag{2.24c}$$

$$x_0 = x_r(t) \tag{2.24d}$$

$$h_0^{k,\ell} = \left[y_k^\top, (\tilde{\mu}_y^{t,k,\ell})^\top, \operatorname{vech}\left[(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}\right]^\top\right]^\top \tag{2.24e}$$

$$W_{\mathcal{J}} h_{\mathcal{J}-1}^{k,\ell} + b_{\mathcal{J}} + r_\ell^2 \leq \delta \tag{2.24f}$$

$$h_i^{k,\ell} = \lambda_i^{k,\ell} + W_i h_{i-1}^{k,\ell} + b_i \tag{2.24g}$$

$$h_i^{k,\ell} \geq 0,\ \lambda_i^{k,\ell} \geq 0 \tag{2.24h}$$

$$(\lambda_i^{k,\ell})^\top h_i^{k,\ell} = 0 \tag{2.24i}$$

$$x_k \in \mathcal{X} \tag{2.24j}$$

$$u_k \in \mathcal{U}, \tag{2.24k}$$

where W_i and b_i are the weights and the bias for the ith layer. Constraints (2.24f)–(2.24i) *are imposed for $i = 1, \ldots, \mathcal{J}$.*

Proof. Consider the feasible set for constraint (2.20e):

$$\begin{aligned} FS_{true}^k :=& \{y_k \in \mathbb{R}^{n_y} \mid \max_{\ell=1,\ldots,L} \mathcal{R}_{t,k}^\ell(y_k, \mathcal{Y}^\ell) \leq \delta\} \\ =& \{y_k \in \mathbb{R}^{n_y} \mid \mathcal{R}_{t,k}^\ell(y_k, \mathcal{Y}^\ell) \leq \delta \ \forall \ell\}. \end{aligned}$$

Using the NN approximation (2.21) of the risk map, the feasible set can be approximated by

$$FS^k_{NN} := \{y_k \in \mathbb{R}^{n_y} \mid \mathcal{R}^{t,k,\ell}_{NN}(y_r, \mathcal{Y}^\ell; \theta, \alpha) \leq \delta \ \forall \ell\}. \tag{2.25}$$

For fixed i, k and ℓ, the ReLU in (2.22) can be interpreted as projecting a_i onto the non-negative orthant, i.e.,

$$h_i = \underset{x \in \mathbb{R}^{\mathcal{N}_i}}{\arg\min} \Big\{ \frac{1}{2}\|x - a_i\|_2^2 \mid x \geq 0 \Big\}. \tag{2.26}$$

Since (2.26) is a convex optimization problem, $h_i = x^*$ and λ_i^* are its primal and dual optimal solutions if and only if the following KKT conditions are satisfied:

$$\begin{aligned} x^* &= \lambda_i^* + a_i \\ (\lambda_i^*)^\top x^* &= 0 \\ \lambda_i^* &\geq 0 \\ x^* &\geq 0. \end{aligned} \tag{2.27}$$

Replacing constraint (2.20e) in the original MPC problem with (2.25) and then expressing ReLU (2.22) as (2.27), we obtain the approximate DR-MPC problem. □

The problem (2.24) can be solved using nonlinear programming algorithms, such as interior-point methods [127, 128], sequential quadratic programming [129, 130]. Moreover, it can also be solved using spatial branch-and-bound algorithms that exploit the bilinear nature of the nonconvex constraint. Similarly, branch-and-bound algorithms [131–134] can be used replacing the nonlinear ineqalities (2.24e)–(2.24h) with corresponding big-M constraints. In this work, for computational efficiency, we employ the interior-point solver implemented in FORCES Pro, which is tailored to efficiently find a locally optimal solution for multistage optimization problems [135].

2.6 Simulation Results

In this section, we provide two case studies to demonstrate the performance and utility of our DR-risk map: one for motion planning and another for motion control. All algorithms were implemented in MATLAB and run on a PC with a 3.70 GHz Intel Core i7-8700K processor and 32 GB RAM. The SDP problems (2.16) and (2.17) were solved using a conic solver, called MOSEK [119]. In the motion control experiment, the FORCES Pro [135] was used to solve the DR-MPC problem.[7]

2.6.1 Motion Planning

As with the first case study, motion planning is performed using our learning-based DR-RRT* in dynamic 2D environments. We consider a car-like robot with the following discrete-time kinematics:

$$\begin{aligned}
\mathrm{x}_r(t+1) &= \mathrm{x}_r(t) + T_s v_r(t)\cos(\theta_r(t)) \\
\mathrm{y}_r(t+1) &= \mathrm{y}_r(t) + T_s v_r(t)\sin(\theta_r(t)) \\
\theta_r(t+1) &= \theta_r(t) + T_s v_r(t)\tan(\delta_r(t))/L_r,
\end{aligned}$$

where $\mathrm{x}_r(t), \mathrm{y}_r(t)$ and $\theta_r(t)$ are the states of the vehicle—representing the Cartesian coordinates of the robot's CoM and its heading angle, while the velocity $\mathrm{v}_r(t)$ and steering angle $\delta_r(t)$ are the control inputs. The sampling time is $T_s = 0.1\ sec$, and $L_r = 0.8\ m$ is the length of the robot. Note that the robot can be covered by a circle with radius $r_r = 1$.

We consider two different scenarios: (i) a 2D environment with obstacles with unknown dynamics, and (ii) a 2D environment with obstacles with single integrator dynamics. In both cases, the parameters for the risk map are chosen as $\alpha = 0.95, r_s = 0.1$ and $r_o^\ell = 1$ for all $\ell = 1, 2$, while the maximum depth for the tree is chosen as $K = 10$. The control inputs for the robot are limited to $|v_r(t)| \leq 5\ m/v^2$ and $|\delta_r(t)| \leq 30\ deg$. In the beginning of the algorithm, since there are no observations,

[7]The source code of our DR-RRT* and DR-MPC implementation is available at `https://github.com/CORE-SNU/DR-Risk-Map`.

the GPR dataset $\mathcal{D}^\ell$ includes only the current values of the ℓth obstacle's states and inputs. New samples are added to the dataset as time goes on.

Highway Scenario

In the first scenario, the robotic vehicle navigates a highway-like 2D environment with $L = 2$ obstacles with unknown behaviors. We parameterize the dynamics model ϕ^ℓ as described in Appendix 2.8.1 using a previously obtained transition dataset of 10^5 observations and a feedforward NN with 3 hidden layers, 20 neurons in each. The state for each obstacle consists of the Cartesian coordinates of its CoM and the heading angle, while the inputs are its velocity and angular acceleration.

Fig. 2.8 shows the trajectories generated by learning-based DR-RRT* for $\theta = 10^{-4}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}$ at different time instances, where two obstacles are shown in green. The goal point is on the second lane. For this experiment the risk tolerance level $\delta = 0.2205$ is set to be 5% of the maximum possible risk $r_\ell^2 = (r_r + r_o^\ell + r_s)^2$. Fig. 2.8 (a) presents the situation when the first obstacle changes the lane from the third to the second lane. Since the obstacle will be on the same lane as the robot according to the prediction, all paths generated by DR-RRT* except for $\theta = 10^{-4}$ choose to move to the third lane. The case of $\theta = 10^{-4}$ is less conservative than the other cases as expected.

After safely avoiding the obstacle, the robot needs to switch back to the second lane to reach the goal point. As shown in Fig. 2.8 (b), the prediction of another obstacle's future motion indicates that the obstacle will continue following the second lane, while in reality, it plans to move to the third lane. Since DR-RRT* with $\theta = 10^{-4}$ considers errors in prediction only in a small ball, the robot chooses to overtake the obstacle, performing risky maneuvers. Meanwhile, the robot with a larger θ makes a safer decision, staying in the third lane. In Fig. 2.8 (c), all cases reach the desired goal point, completing the algorithm. Overall, it is observed that the case with the smallest radius $\theta = 10^{-4}$ generates the most aggressive (but still safe) path. Increasing

the radius drives the robot farther away from the obstacles, thereby guaranteeing safe navigation with enough of a safety margin. Clearly, $\theta = 10^{-1}$ ensures a larger safety margin compared to the case of 10^{-2} or 5×10^{-2}.

Fig. 2.9 illustrates how the tree grows at $t = 18$ in the case of $\theta = 10^{-2}$. The tree starts from the current state of the robot. At the same time, GPR is executed to predict the obstacles' future motions. Unfortunately, the prediction capability is poor when there are abrupt changes in the behavior of the obstacles. However, the prediction errors are taken into account in our DR-risk map, guaranteeing safety even when the prediction is not accurate. The grey tree corresponds to $\mathcal{T}$ obtained using Algorithm 1. However, to ensure safety, only the nodes with depth less than or equal to K and satisfying the risk constraint are added to the safe subtree $\mathcal{T}_{\text{safe}}$. The best path (in red) given to the robot for execution is then chosen from $\mathcal{T}_{\text{safe}}$.

Table. 2.4 shows the cumulative cost of the trajectories generated by DR-RRT* with different θ's. A bigger radius induces a more conservative behavior, driving the robot away from the shortest path. Thus, the total trajectory length and the cost increase with θ.

To examine the robustness of our method depending on the ambiguity set size and determine an appropriate radius θ, the average collision probability is computed for $N = 1000$ realizations of GP dataset $\mathcal{D}^\ell$. In particular, we assume a zero-mean Gaussian measurement noise with variance $0.001I$ and learn hyperparameters of the GP prior based on each realization of $\mathcal{D}^\ell$. The probability of collision is calculated as the collision rate averaged over N simulations, i.e.,

$$\mathrm{P}_{\text{coll}}^{t,k,\ell} = \hat{\mathbb{P}}\left\{\mathcal{J}_{t,k}(y_r^*, y_o^\ell) + r_\ell^2 > 0\right\},$$

where $\hat{\mathbb{P}}$ is the empirical distribution of the GP dataset, and $y_r^*(t+k)$ is the robot's position at time $t+k$ planned at stage t using the learned distribution with $\mathcal{D}^\ell$, while $y_o^\ell(t+k)$ is the actual position of obstacle ℓ. The overall collision rate is then computed

as

$$\mathrm{P}_{\mathrm{coll}} = \bigcup_{t=0}^{T} \bigcup_{k=0}^{K} \bigcup_{\ell=1}^{L} \mathrm{P}_{\mathrm{coll}}^{t,k,\ell}.$$

The results of our analysis are reported in Table 2.4. For all θ's, the collision probability is very small and decreases with the size of the ambiguity set. Therefore, one can adjust the robustness of the robot's decision by choosing a radius θ to reach the desired level of collision probability. In this example, $\theta = 5 \times 10^{-2}$ is a reasonable choice if the targeted collision rate is 1%.

Road Intersection Scenario

In the second scenario, we consider a road intersection, where an obstacle has an unknown behavior with a single integrator dynamics:

$$x_o(t+1) = x_o(t) + T_s u_o(t),$$

where $x_o(t)$ is the obstacle's position and $u_o(t)$ is the velocity vector in each direction. This setting allows us to compare our method with other algorithms that can only handle limited problem classes. Specifically, we compare our method to the classical RRT* [122] as well as the CC-RRT* algorithm [124]. This comparison is impossible in the first scenario, where angular uncertainties are considered in addition to the placement uncertainties; CC-RRT* can only handle the latter. In the case of RRT*, we assume that the prediction results are accurate and consider the predicted mean to be the actual obstacle's position, ignoring uncertainties. In CC-RRT*, the obstacle is over-approximated as an octagon to attain its polytopic representation. CC-RRT* uses chance constraints assuring that the probability of navigating in the safe set is greater than or equal to α. We set the risk weight in the cost (2.19) to $w = 0$ to ensure the same conditions for all algorithms.

Fig. 2.10 shows the simulation results of DR-RRT* with $\theta = 10^{-4}, 10^{-3}, 5 \times 10^{-3}$ and comparisons to RRT* and CC-RRT* at different time instances. In Fig. 2.10 (a),

the robot reaches the intersection without considering the obstacle, as it is still not interfering with the robot's path. The obstacle is trying to turn right, which is predicted well by GPR. However, as shown in Fig. 2.10 (b), when the robot is trying to steer left, the obstacle abruptly changes its decision to turn left. This situation is clearly not predicted well by GPR, and therefore RRT* and CC-RRT* both fail to find a feasible solution. However, our DR-RRT* takes into account such an error in the learning result, guiding the robot to avoid a collision. Even though DR-RRT* succeeds in generating a collision-free path for all θ's, the path with smaller θ is riskier than that with a bigger one. With the biggest radius ($\theta = 5 \times 10^{-3}$), the robot avoids the obstacle with a sufficient safety margin. Finally, Fig. 2.10 (c) shows the completed paths generated by DR-RRT*, whereas both RRT* and CC-RRT* fail to complete their paths. We can conclude that RRT* is not suitable for motion planning in a highly uncertain environment, while CC-RRT* is applicable if the prediction results are accurate, as it does not consider distributional errors. However, our DR-RRT* is capable of performing safe path planning even with the existence of distributional errors in the learning results.

Similar to the previous scenario, the probability of collision is computed using perturbed predictions with the same perturbation parameters. Both RRT* and CC-RRT* fail to complete motion planning, and thus the probability of collision is 1 for both. In the case of our DR-RRT*, the collision probability is 0, meaning that there is no collision for all Wasserstein ambiguity sets considered in this specific experiment.

2.6.2 Motion Control

In the second case study, we consider a motion control problem for a service robot in a cluttered environment such as a restaurant. The mobile robot is assumed to move

according to the following double integrator dynamics:

$$x_r(t+1) = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_r(t) + \begin{bmatrix} T_s^2/2 & 0 \\ 0 & T_s^2/2 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} u_r(t),$$

where $x_r(t) = (\mathrm{x}_r(t), \mathrm{y}_r(t), \mathrm{v}_{xr}(t), \mathrm{v}_{yr}(t)) \in \mathbb{R}^4$ is the robot's state at time t, consisting of the Cartesian coordinates of its CoM and the corresponding velocity vector, and the input $u_r(t) = (\mathrm{a}_{xr}(t), \mathrm{a}_{yr}(t)) \in \mathbb{R}^2$ is chosen as the acceleration vector. Again, T_s is the sampling time, selected as $0.1\ sec.$

The circular robot of radius $r_r = 0.09$ aims to track a given reference trajectory in a cluttered 2D environment with some static and dynamic obstacles that may represent other service robots or human agents. Each of $L = 3$ dynamic obstacles is a circular object of radius $r_o^\ell = 0.1$, and the safety margin is set to be $r_s = 0.01$. The control input for the robot is limited to lie in $\mathcal{U} := \{u \in \mathbb{R}^2 \mid \|u\|_\infty \leq 4\}$, while its state is restricted to $\mathcal{X} := \{x \in \mathbb{R}^4 \mid (0, 0, -2, -2) \leq x \leq (6, 6, 2, 2)\}$. Each of the $L_{\text{stat}} = 5$ static obstacles is approximated by an ellipsoid, defined as $\mathcal{O}_{\text{stat}}^i := \{x \in \mathbb{R}^2 \mid (x - x_{\text{stat}}^i)^\top P_i^{-1}(x - x_{\text{stat}}^i) \leq 1\}$, where x_{stat}^i is the center of ith elliptical obstacle and $P_i = P_i^\top \succ 0$ determines how far the ellipsoid extends in every direction from x_{stat}^i. The following additional constraints are added to problem (2.24) to avoid the static obstacles:

$$(y_k - x_{\text{stat}}^i)^\top P_i^{-1}(y_k - x_{\text{stat}}^i) \geq 1 \quad \forall i = 1, \ldots, L_{\text{stat}}.$$

The NN approximation of the DR-risk map is performed as described in Section 2.5.1. We sample 500,000 different values of $y_r(t+k)$ and $\tilde{\mu}_y^{t,k,\ell}$ from $\mathcal{U}[0, 6]^2$ and $\operatorname{vech}\left[(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}\right]$ from $\mathcal{U}[0, 0.1]^3$ and divide them into training, validation and testing datasets with a ratio of $0.8 : 0.1 : 0.1$.

We begin the MPC algorithm by applying GPR to predict the mean $\tilde{\mu}_y^{t,k,\ell}$ and covariance $\tilde{\Sigma}_y^{t,k,\ell}$ for all dynamic obstacles $\ell = 1, \ldots, L$ for future time steps $k =$

$1, \ldots, K$ based on the latest $M = 10$ observations of the obstacles' behaviors. This step is repeated in every time stage t before solving the optimization problem (2.24).

We compare the performance of our approximate DR-MPC (2.24) with that of the CVaR-constrained sample average approximation MPC (CVaR-MPC) [70] with $N = 100$ sample data generated from the predicted distribution, as well as the chance-constrained MPC (CC-MPC) for elliptical obstacles [136]. The risk confidence level is chosen as $\alpha = 0.95$. For CVaR-MPC and DR-MPC, the risk tolerance level $\delta = 4 \times 10^{-4}$ is set to be 1% of the maximum possible risk $r_\ell^2 = (r_r + r_o^\ell + r_s)^2$. In our approximate DR-MPC, the radius is chosen as $\theta = 10^{-5}, 10^{-4}, 10^{-2}$.

Fig. 2.11 shows the simulation results for the three MPC methods with prediction horizon $K = 10$. In both CVaR-MPC and CC-MPC, the GPR prediction results are used for risk assessment. However, due to some sudden and unpredictable movements of the obstacles, the GPR results are not trustworthy. As shown in Fig. 2.11 (a), the mobile robot follows the reference trajectory and approaches the first dynamic obstacle. In this stage, all controllers try to avoid the obstacle by passing it on the left with different safety margins. However, even though CC-MPC finds a feasible solution under the inaccurately predicted distribution, a collision occurs in reality due to the prediction and approximation errors. Similarly, after a few steps, the robot controlled by CVaR-MPC collides with the obstacle. Unlike the two controllers, DR-MPC controls the robot to safely avoid the obstacle and continue following the reference trajectory despite the inaccurate GPR results. This is because, instead of directly using the learned distribution, DR-MPC considers the risk of unsafety with respect to the worst-case distribution within distance θ from the learned one. This is shown in Fig. 2.11 (b), where the robot has already passed the obstacle. The radius θ affects the behavior of the robot in a way that increasing it results in a more risk-averse steering behavior. In particular, DR-MPC with $\theta = 10^{-2}$ generates the most conservative trajectory, while the trajectory for $\theta = 10^{-5}$ is the least safe, being close to that generated by CVaR-MPC. This is because, as $\theta \to 0$, the ambiguity set vanishes, and DR-CVaR reduces to CVaR. In

Fig. 2.11 (c), the robot approaches the third and fourth dynamic obstacles. Similar to the previous situation, DR-MPC guides the robot to safely avoid the obstacles with some safety margins depending on the size of the ambiguity set. Finally, as shown in Fig. 2.11 (d), the robot controlled by our DR-MPC method successfully reaches the goal point, unlike the other two methods.

The cumulative costs incurred by the three methods are reported in Table 2.5.[8] Obviously, the cost increases as the controller becomes more conservative, as the robot drives away from the obstacles with larger safety margins.

Table 2.5 also shows the probability of collision, averaged over $1,000$ different GP datasets computed similarly to the motion planning case. CC-MPC has the highest probability of collision, followed by CVaR-MPC. This is justified by the fact that chance constraint can be equivalently expressed using value-at-risk (VaR), while CVaR-MPC uses CVaR. By definition, it holds that $\mathrm{VaR}[X] \leq \mathrm{CVaR}[X]$, and therefore the CVaR-based CVaR-MPC induces more conservative behavior compared to CC-MPC. Our DR-MPC reduces the collision probability to 0.034 even with a very small ambiguity set ($\theta = 10^{-5}$). Increasing the radius to $\theta = 10^{-2}$ further reduces the probability of collision with the obstacles to 0.001.

The computation time reported in Table 2.5 is measured from the starting point to the goal point. The results show that CC-MPC and DR-MPC with $\theta = 10^{-5}$ take a similar amount of time to complete motion control, while CVaR-MPC is slightly slower due to the number of constraints in the optimization problem for each sample. As for the remaining θ's, increasing the safety of the robot is comparatively computationally heavy as finding a feasible trajectory satisfying the risk constraints becomes more time-consuming. From these results, we can conclude that it is reasonable to use $\theta = 10^{-4}$ in this problem, which produces a sufficiently robust behavior with moderate operation cost and computation time.

[8] In the cases of CC-MPC and CVaR-MPC, we continued to perform motion control even after collisions.

2.7 Conclusions

We have proposed a novel risk assessment tool, called the DR-risk map, for a mobile robot in a cluttered environment with moving obstacles. Our risk map is robust against distribution errors in the obstacles' motions predicted by GPR. For computational tractability, an SDP formulation was introduced along with its dual SDP. The utility of the risk map was demonstrated through its application to motion planning and control. The DR-RRT* algorithm uses the DR-risk map in the cost and constraint to generate a safe path in the presence of learning errors. Furthermore, to reduce the computational cost, an NN approximation of the risk map was proposed and embedded into our MPC problem for motion control. The results of our simulation studies demonstrate the capability of the DR-risk map to preserve safety under learning errors.

2.8 Appendix

2.8.1 Neural Network Approximation of Obstacle Dynamics

As mentioned in Section 2.2.1, the system model of obstacles might be unknown in practice. However, with some observation data, an approximate model ϕ_w of ϕ can be constructed using NNs. In this work, we use feedforward NNs with ReLU activation functions and $\mathcal{L}_\phi$ hidden layers to approximate the obstacles' dynamics. The input of the NN consists of the obstacles' state and action vectors at each time stage. The target of the NN is chosen as the difference between the next state and the current state to take advantage of the discrete nature of the dynamics. The training data is collected through the observation of N_ϕ random transitions $(x_o(t), u_o(t), x_o(t+1))$, constructing the input and target datasets $D_{\text{in}}^\ell = \big\{(x_o(t), u_o(t))\big\}_{t=0}^{N_\phi-1}$ and $D_{\text{tar}}^\ell = \big\{x_o(t+1) - x_o(t)\big\}_{t=0}^{N_\phi-1}$, respectively. Given the datasets, the NN $\hat{\phi}_w$ is trained by minimizing the mean squared error:

$$L_\phi(w) = \sum_{t=0}^{N_\phi-1} \frac{1}{2}\|\hat{\phi}_w(x_o(t), u_o(t)) - (x_o(t+1) - x_o(t))\|^2,$$

where the parameter vector w represents the network weights. As a result of optimization, we obtain the following approximate model for obstacle dynamics:

$$\phi_w(x_o(t), u_o(t)) = x_o(t) + \hat{\phi}_w(x_o(t), u_o(t)), \tag{2.28}$$

which replaces the function ϕ in the obstacle dynamics (2.8).

2.8.2 Proofs

Proof of Theorem 2.1

Proof. We use the definition of CVaR to wrtie the DR-risk as follows:

$$\begin{aligned}\text{DR-CVaR}_{\alpha,\theta}\big[\mathcal{J}(y_r, y_o)\big] &= \sup_{\mathrm{Q}\in\mathbb{D}} \inf_{z\in\mathbb{R}} \left(z + \frac{1}{1-\alpha}\mathbb{E}^{\mathrm{Q}}\big[\big(\mathcal{J}(y_r, y_o) - z\big)^+\big] \right) \\ &\leq \inf_{z\in\mathbb{R}} \left(z + \frac{1}{1-\alpha} \sup_{\mathrm{Q}\in\mathbb{D}} \mathbb{E}^{\mathrm{Q}}\big[\big(\mathcal{J}(y_r, y_o) - z\big)^+\big] \right),\end{aligned}$$

where the inequality follows from the minimax inequality.

It is well known that for the standard Euclidean norm $\|\cdot\|_2$ the 2-Wasserstein distance between two normal distributions $\mathrm{Q} = \mathcal{N}(\mu_1, \Sigma_1)$ and $\mathrm{P} = \mathcal{N}(\mu_2, \Sigma_2)$ has a closed-form expression [137]:

$$W_2(\mathrm{Q}, \mathrm{P}) = \sqrt{\|\mu_1 - \mu_2\|_2^2 + B^2(\Sigma_1, \Sigma_2)},$$

where

$$B^2(\Sigma_1, \Sigma_2) := \mathrm{Tr}\Big[\Sigma_1 + \Sigma_2 - 2\big(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\big)^{1/2}\Big].$$

Consider the following convex uncertianty set, which is the projection of $\mathbb{D}$ onto the space of means and covariances:

$$\mathcal{U}_\theta(\tilde{\mu}, \tilde{\Sigma}) = \Big\{(\mu, \Sigma) \in \mathbb{R}^{n_y} \times \mathbb{S}_+^{n_y} \mid \|\mu - \tilde{\mu}\|_2^2 + B^2(\Sigma, \tilde{\Sigma}) \leq \theta^2\Big\}. \tag{2.29}$$

The uncertainty set $\mathcal{U}_\theta(\tilde{\mu}, \tilde{\Sigma})$ is convex and compact since it is the projection of the Wasserstein ball. We now leverage the Gelbrich hull, defined in [41], which contains all distributions supported on Ξ whose mean and covariance fall into the uncertainty set

$\mathcal{U}_\theta(\tilde{\mu}, \tilde{\Sigma})$. In our case, since we consider nominal Gaussian distributions, the Gelbrich hull is identical to the Wasserstein ball $\mathbb{D}$ defined in (2.12). Due to nonlinearity of covariance matrix in the underlying distribution, it is reasonable to perform change of variables and represent the uncertainty set $\mathcal{U}_\theta(\tilde{\mu}, \tilde{\Sigma})$ by the second-order moment $M = \mathbb{E}[y_o y_o^\top] = \Sigma + \mu\mu^\top$. Then the new uncertainty set $\mathcal{V}_\theta(\tilde{\mu}, \tilde{\Sigma})$ will be defined as:

$$\mathcal{V}_\theta(\tilde{\mu}, \tilde{\Sigma}) = \big\{(\mu, M) \in \mathbb{R}^{n_y} \times \mathbb{S}_+^{n_y} \mid (\mu, M - \mu\mu^\top) \in \mathcal{U}_\theta(\tilde{\mu}, \tilde{\Sigma})\big\}, \tag{2.30}$$

which is also a convex set.

Now, we use the fact that the Gelbrich hull or the 2-Wasserstein ball in our case can be expressed as the union of Chebyshev ambiguity sets with means and covariances in the uncertainty set (2.29). Equivalently, using the uncertainty set (2.30), the Gelbrich hull can be viewed as the union of Chebyshev ambiguity sets with first- and second-order moments in the uncertainty set (2.30), i.e.,

$$\begin{aligned}\mathbb{D} &= \bigcup_{(\mu,\Sigma)\in\mathcal{U}_\theta(\tilde{\mu},\tilde{\Sigma})} \mathcal{C}(\mathbb{R}^{n_y}, \mu, \Sigma) \\ &= \bigcup_{(\mu,M)\in\mathcal{V}_\theta(\tilde{\mu},\tilde{\Sigma})} \mathcal{C}(\mathbb{R}^{n_y}, \mu, M - \mu\mu^\top),\end{aligned}$$

where $\mathcal{C}(\mathbb{R}^{n_y}, \mu, \Sigma)$ is the Chebyshev ambiguity set containing all distributions on $\mathbb{R}^{n_y}$ with mean μ and covariance bounded above by Σ. Thus, we have

$$\sup_{\mathrm{Q}\in\mathbb{D}} \mathbb{E}^{\mathrm{Q}}\big[\big(\mathcal{J}(y_r, y_o) - z\big)^+\big] = \sup_{(\mu,M)\in\mathcal{V}_\theta(\tilde{\mu},\tilde{\Sigma})} \sup_{\mathrm{Q}\in\mathcal{C}(\mathbb{R}^{n_y},\mu,M-\mu\mu^\top)} \mathbb{E}^{\mathrm{Q}}\big[\big(\mathcal{J}(y_r, y_o) - z\big)^+\big].$$

In the above equation, the inner optimization problem measures the risk for all distributions with given first- and second-order moments, while the outer one considers the ambiguity in those moments with respect to the Wasserstein distance. Such two-layered optimization provides additional robustness, accounting for moment ambiguities.

From [138, Lemma A.1] the inner supremum gets the following dual form:

$$
\begin{cases}
\inf & \tau + 2\gamma^\top \mu + \langle \Gamma, M \rangle \\
\text{s.t.} & \tau + 2\gamma^\top y_o + \langle \Gamma, y_o y_o^\top \rangle \geq \mathcal{J}(y_r, y_o) - z \; \forall y_o \\
& \tau + 2\gamma^\top y_o + \langle \Gamma, y_o y_o^\top \rangle \geq 0 \; \forall y_o \\
& \tau \in \mathbb{R}, \gamma \in \mathbb{R}^{n_y}, \Gamma \in \mathbb{S}^{n_y}
\end{cases}
$$

$$
= \begin{cases}
\inf & \tau + 2\gamma^\top \mu + \mathrm{Tr}[\Gamma M] \\
\text{s.t.} & \begin{bmatrix} \Gamma + I & \gamma - y_r \\ (\gamma - y_r)^\top & \tau + z + \|y_r\|_2^2 \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} \Gamma & \gamma \\ \gamma^\top & \tau \end{bmatrix} \succeq 0 \\
& \tau \in \mathbb{R}, \gamma \in \mathbb{R}^{n_y}, \Gamma_+ \in \mathbb{S}^{n_y},
\end{cases}
\tag{2.31}
$$

where the second problem is obtained by replacing the quadratic constraint with the corresponding semidefinite one. By weak duality, the dual provides an upper bound of the inner supremum. Applying minimax inequality and replacing the inner supremum with its dual, we arrive at the following upper bound for the worst-case expectation:

$$
\inf_{\tau,\gamma,\Gamma} \Big\{ \tau + \sup_{(\mu,M)\in\mathcal{V}_\theta(\tilde{\mu},\tilde{\Sigma})} \big(2\gamma^\top \mu + \mathrm{Tr}[\Gamma M]\big) \qquad \big| \text{ constraints in (2.31)} \Big\}. \tag{2.32}
$$

The inner supremum has an interesting form, which can be rewritten by the support function of $\mathcal{V}_\theta(\tilde{\mu}, \tilde{\Sigma})$ evaluated at $(2\gamma, \Gamma)$. The support function $\sigma_{\mathcal{V}_\theta(\tilde{\mu},\tilde{\Sigma})}(q, Q)$ for any

$q \in \mathbb{R}^m$ and $Q \in \mathbb{S}^m$ can found by solving the following SDP problem [41]:

$$\begin{aligned}\sigma_{\mathcal{V}_\theta(\tilde{\mu},\tilde{\Sigma})}(q,Q) = \inf_{\lambda,\varepsilon,Z}\ & \lambda(\theta^2 - \|\tilde{\mu}\|_2^2 - \mathrm{Tr}[\tilde{\Sigma}]) + \varepsilon + \mathrm{Tr}[Z] \\ \text{s.t.}\ & \begin{bmatrix} \lambda I - Q & \lambda\tilde{\mu} + \frac{q}{2} \\ \lambda\tilde{\mu}^\top + \frac{q^\top}{2} & \varepsilon \end{bmatrix} \succeq 0 \\ & \begin{bmatrix} \lambda I - Q & \lambda\tilde{\Sigma}^{1/2} \\ \lambda\tilde{\Sigma}^{1/2} & Z \end{bmatrix} \succeq 0 \\ & \lambda \in \mathbb{R}_+, \varepsilon \in \mathbb{R}_+, Z \in \mathbb{S}_+^m.\end{aligned}$$

The result of the theorem follows from replacing the support function with the corresponding SDP and plugging in the expression for the worst-case expectation back into DR-risk. □

Proof of Corollary 2.1

Proof. To derive the dual of (2.16), we write the Lagrangian functions with multipliers $X, Y, W, V \succeq 0$ and $\eta, \beta \geq 0$ as

$$\begin{aligned}\mathscr{L} = z + & \frac{\Big[\tau + \varepsilon + \mathrm{Tr}[Z] + \lambda\big(\theta^2 - \|\tilde{\mu}\|_2^2 - \mathrm{Tr}[\tilde{\Sigma}]\big)\Big]}{1-\alpha} \\ & - \langle X_{11}, \lambda I - \Gamma\rangle - 2X_{12}^\top(\gamma + \lambda\tilde{\mu}) - X_{22}\varepsilon - \langle Y_{11}, \lambda I - \Gamma\rangle \\ & - 2\langle Y_{12}, \lambda\tilde{\Sigma}^{1/2}\rangle - \langle Y_{22}, Z\rangle - \langle W_{11}, \Gamma + I\rangle - 2W_{12}^\top(\gamma - y_r) \\ & - W_{22}(\tau + y_r^\top y_r + z) - \langle V_{11}, \Gamma\rangle - 2V_{12}^\top\gamma - V_{22}\tau \\ & - \langle U, Z\rangle - \eta\lambda - \beta\varepsilon,\end{aligned}$$

where X_{ij} is the (i,j) entry of matrix X and $\langle\cdot,\cdot\rangle$ is the matrix inner product. The dual function g is obtained by minimizing the Lagrangian function with respect to the

primal variables:

$$
\begin{aligned}
g = &-\mathrm{Tr}[W_{11}] - 2W_{12}^\top y_r - W_{22} y_r^\top y_r + \min_z (1 - W_{22}) z \\
&+ \min_\lambda \Big(\frac{\theta^2 - \|\tilde{\mu}\|_2^2 - \mathrm{Tr}\big[\tilde{\Sigma}\big]}{1-\alpha} - \mathrm{Tr}\big[X_{11} + Y_{11} + 2Y_{12}^\top \tilde{\Sigma}^{1/2}\big] \\
&- 2X_{12}^\top \tilde{\mu} - \eta \Big) \lambda + \min_\gamma (-2X_{12} - 2W_{12} - 2V_{12})^\top \gamma \\
&+ \min_\varepsilon \Big(\frac{1}{1-\alpha} - X_{22} - \beta \Big) \varepsilon + \min_\tau \Big(\frac{1}{1-\alpha} - W_{22} - V_{22} \Big) \tau \\
&+ \min_\Gamma \langle X_{11} + Y_{11} - W_{11} - V_{11}, \Gamma \rangle + \min_Z \langle I - Y_{22} - U, Z \rangle.
\end{aligned}
$$

Finally, solving the inner minimization problems and maximizing the dual function g with respect to the dual variables, we obtain the dual form (2.17).

Note that there exist strictly feasible points for the primal problem (2.16) for any $\tilde{\mu} \in \mathbb{R}^{n_y}$ and $\tilde{\Sigma} \in \mathbb{S}_+^{n_y}$. For example, let

$$
\begin{aligned}
&\gamma = -2\tilde{\mu}, && z = 2\tilde{\mu}^\top y_r - \frac{1}{2}\|y_r\|_2^2 - 2\|\tilde{\mu}\|_2^2 + 1, \\
&\lambda = 2, && \Gamma = I \succ 0, \\
&Z = 4\tilde{\Sigma} + I, && \varepsilon = \tau = 2\gamma^\top \gamma > 0.
\end{aligned}
$$

Then the constraints in (2.16) hold with strict inequalities. Therefore, Slater's condition holds and so does strong duality. □

Proof of Theorem 2.2

Proof. Let $\mathrm{P}_{t,k}^{\mathcal{D}}$ and $\mathbf{P}_{t,k}$ denote the probability distribution obtained by GPR and the Dirac measure concentrated at $y_o(t+k)$, respectively. It follows from the definition of 2-Wasserstein distance that

$$
W_2(\mathrm{P}_{t,k}^{\mathcal{D}}, \mathbf{P}_{t,k}) \leq \|y_o(t+k) - \tilde{\mu}_{y,\mathcal{D}}^{t,k}\|^2 + \mathrm{Tr}\big[\tilde{\Sigma}_{y,\mathcal{D}}^{t,k}\big].
$$

Therefore, Assumption 2.1 implies that

$$
W_2(\mathrm{P}_{t,k}^{\mathcal{D}}, \mathbf{P}_{t,k}) \leq \Big(\omega_{\mathcal{D}}^{t,k}\Big)^2 + \mathrm{Tr}\big[\Omega_{\mathcal{D}}^{t,k}\big] \leq \theta_{t,k}^2
$$

holds with probability no less than $(1-p)^k$. It indicates that the true probability distribution $\mathbf{P}_{t,k}$ is contained in the Wasserstein ambiguity set with radius $\theta_{t,k}$ around the learned distribution $\mathrm{P}_{t,k}^{\mathcal{D}}$ with probability no less than $(1-p)^k$. Thus, by the definition of DR-CVaR,

$$\mathbb{P}\left\{\mathcal{D} \mid \mathrm{CVaR}_{\alpha}^{\mathbf{P}_{t,k}}[\mathcal{J}_{t,k}(y_r, y_o)] \leq \text{DR-CVaR}_{\alpha,\theta}[J(y_r, y_o)]\right\} \geq (1-p)^k,$$

where $\text{DR-CVaR}_{\alpha,\theta}$ depends on the training data $\mathcal{D}$ via the radius $\theta_{t,k}$ and the learned distribution $\mathrm{P}_{t,k}^{\mathcal{D}}$. Since $\mathbf{P}_{t,k}$ is the Dirac delta measure concentrated at $y_o(t+k)$,

$$\mathrm{CVaR}_{\alpha}^{\mathbf{P}_{t,k}}[\mathcal{J}_{t,k}(y_r, y_o)] = \mathcal{J}_{t,k}(y_r, y_o).$$

Moreover, it follows from the definition of the DR-risk map $\mathcal{R}$ that

$$\text{DR-CVaR}_{\alpha,\theta}\big[\mathcal{J}(y_r, y_o)\big] + r^2 \leq \mathcal{R}_{t,k}^{\mathcal{D}}(y_r, \mathcal{Y}) \leq \mathcal{R}_{t,k}^{\mathcal{D}}(y_r).$$

Thus, the result follows. □

Algorithm 1: Learning-based DR-RRT*

Input: $q_{\text{goal}}, K, \theta, \alpha, r_\ell, w, r_{\text{RRT}}$;

$\mathcal{T} = \emptyset, \mathcal{D}^\ell \leftarrow \emptyset$;

while $\|\text{Root}(\mathcal{T}) - q_{\text{goal}}\|_2 > \epsilon$ **do**

  $t \leftarrow \text{clock}()$;

  $\mathcal{T}_{\text{safe}} \leftarrow \emptyset$;

  Observe $x_r(t)$ and $x_o^\ell(t), u_o^\ell(t)$ for all ℓ;

  $\text{Root}(\mathcal{T}) \leftarrow x_r(t)$;

  Remove unreachable nodes from $\mathcal{T}$;

  Reset node depth;

  for $\ell = 1$ *to* L **do**

    $\mathcal{D}_j^\ell \leftarrow \mathcal{D}_j^\ell \cup \left\{(x_o^\ell(t), u_{o,j}^\ell(t))\right\},\ j = 1, \ldots, n_u^\ell$;

    GP approximation of $\psi^\ell(\mathbf{x})$ via (2.5)–(2.6);

    $\tilde{\mu}_x^{t,0,\ell} \leftarrow x_o^\ell(t), \tilde{\Sigma}_x^{t,0,\ell} \leftarrow \mathbf{0}$;

    for $k = 0$ *to* $K - 1$ **do**

      Compute $\tilde{\mu}_u^{t,k,\ell}$, $\tilde{\Sigma}_u^{t,k,\ell}$ and $\tilde{\Sigma}_{xu}^{t,k,\ell}$ from (2.7);

      Update $\tilde{\mu}_y^{t,k+1,\ell}$ and $\tilde{\Sigma}_y^{t,k+1,\ell}$ by (2.8)–(2.9);

  for $\forall q \in \mathcal{T}$ *with* $\text{Depth}(q) \leq K$ **do**

    $k \leftarrow \text{Depth}(q)$;

    Update $\mathcal{R}_{t,k}(C_r q)$ by solving (2.16);

    Update $c(q)$ by solving (2.19);

    if $\mathcal{R}_{t,k}(C_r q) \leq \delta$ **then**

      Add q to $\mathcal{T}_{\text{safe}}$;

  while $\text{clock}() \leq \tau$ **do**

    Expand the tree using Algorithm 2

  Plan path $(\text{Root}(\mathcal{T}_{\text{safe}}), q_1, \ldots, \ldots, q_K)$ in $\mathcal{T}_{\text{safe}}$;

  Drive $x_r(t)$ to q_1;

Algorithm 2: Tree expansion and rewiring

Input: $\mathcal{T}, \mathcal{T}_{\text{safe}}, t$;

$q_{\text{rand}} \leftarrow \text{Sample}()$;

$q_{\text{nearest}} \leftarrow \text{NearestNeighbor}(\mathcal{T}_{\text{safe}}, q_{\text{rand}})$;

$k \leftarrow \text{Depth}(q_{\text{nearest}}) + 1$;

$(q_{\text{new}}, c(q_{\text{new}}), \mathcal{R}_{t,k}(C_r q_{\text{new}})) \leftarrow \text{Steer}(q_{\text{nearest}}, q_{\text{rand}})$;

$\mathcal{N}_{\text{near}} \leftarrow \text{Near}(\mathcal{T}_{\text{safe}}, q_{\text{new}}, r_{\text{RRT}})$;

$q_{\min} \leftarrow q_{\text{nearest}}, c_{\min} \leftarrow c(q_{\text{new}})$;

for $q_{\text{near}} \in \mathcal{N}_{\text{near}}$ **do**

  $k \leftarrow \text{Depth}(q_{\text{near}}) + 1$;

  $c_{\text{near}} \leftarrow c(q_{\text{near}}) + w\mathcal{R}_{t,k}(C_r q_{\text{new}}) + \mathcal{L}(q_{\text{near}}, q_{\text{new}})$;

  if $c_{\text{near}} < c_{\min}$ *and* $\text{Feas}(q_{\text{near}}, q_{\text{new}})$ **then**

    $q_{\min} \leftarrow q_{\text{near}},\ c_{\min} \leftarrow c_{\text{near}}$;

$c(q_{\text{new}}) \leftarrow c_{\min}, \text{Parent}(q_{\text{new}}) \leftarrow q_{\min}$;

$k \leftarrow \text{Depth}(q_{\text{new}})$;

Add q_{new} to $\mathcal{T}$;

if $\mathcal{R}_{t,k}(C_r q_{\text{new}}) \leq \delta$ **then**

  Add q_{new} to $\mathcal{T}$;

for $q_{\text{near}} \in \mathcal{N}_{\text{near}}$ **do**

  $k \leftarrow \text{Depth}(q_{\text{new}}) + 1$;

  $c_{\min} \leftarrow c(q_{\text{new}}) + w\mathcal{R}_{t,k}(C_r q_{\text{near}}) + \mathcal{L}(q_{\text{new}}, q_{\text{near}})$;

  if $c_{\min} \leq c(q_{\text{near}})$ *and* $\text{Feas}(q_{\text{new}}, q_{\text{near}})$ **then**

    $c(q_{\text{near}}) = c_{\text{near}}$;

    $\text{Parent}(q_{\text{near}}) \leftarrow q_{\text{new}}$;

    Update children nodes of q_{near} ;

    if $\text{Depth}(q_{\text{near}}) > K$ **then**

      Remove q_{near} and its children from $\mathcal{T}_{\text{safe}}$;

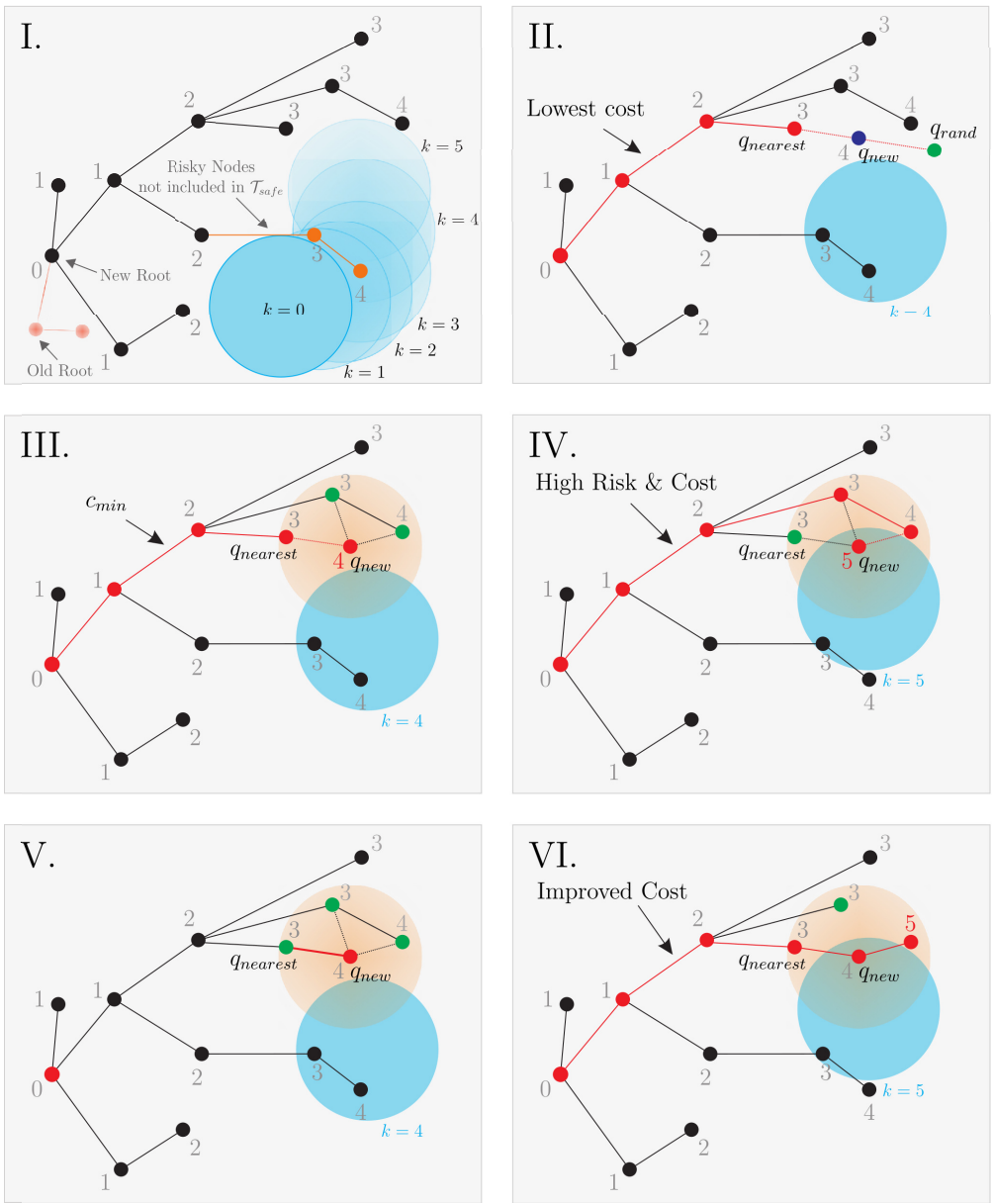


Figure 2.6: Illustrative example of learning-based DR-RRT*. The blue ball represents an obstacle (at different time instances) centered at the predicted mean.

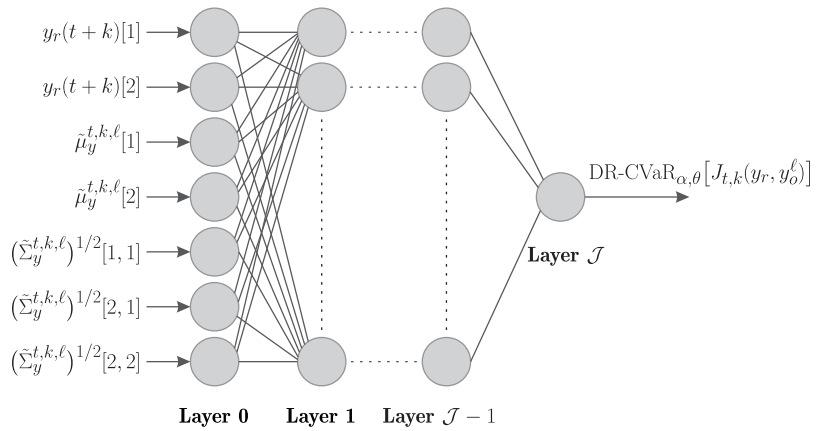


Figure 2.7: Feed-forward NN for approximating the DR-risk map for fixed θ and α. The inputs are the robot's position y_r and the parameters of the predicted distribution of the obstacles' behaviors $\tilde{\mu}_y^{t,k,\ell}$ and $\text{vech}\big[(\tilde{\Sigma}_y^{t,k,\ell})^{1/2}\big]$, while the target is the DR-risk. Here, $[i]$ refers to the ith entry of a vector, while $[i,j]$ is the entry in the ith row and the jth column of a matrix.

Table 2.2: Mean squared error (MSE) and mean average error (MAE) for the NN approximation of the DR-risk map with 405,000 training, 45,000 validation, and 50,000 test data points.

Radius θ		10^{-5}	10^{-3}	10^{-2}
MSE	**Train**	9.036×10^{-7}	2.780×10^{-6}	2.909×10^{-6}
	Validation	9.710×10^{-7}	2.994×10^{-6}	2.569×10^{-6}
	Test	9.100×10^{-7}	3.343×10^{-6}	2.538×10^{-6}
MAE	**Train**	2.637×10^{-4}	4.449×10^{-4}	4.473×10^{-4}
	Validation	2.808×10^{-4}	3.624×10^{-4}	2.756×10^{-4}
	Test	2.806×10^{-4}	3.866×10^{-4}	2.757×10^{-4}

Table 2.3: Probability of the approximate risk map reporting wrong results.

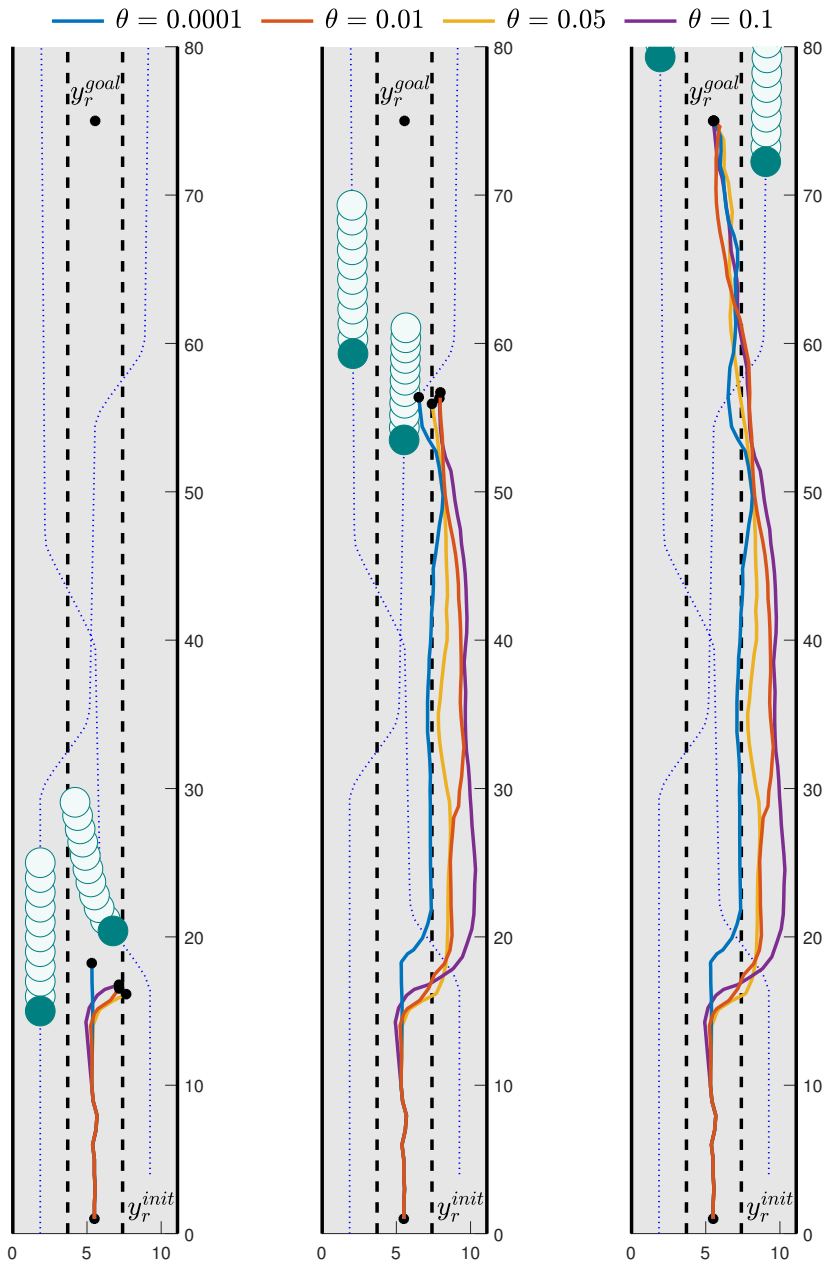
Radius θ	Safe events reported as unsafe	Unsafe events reported as safe
10^{-5}	1.5×10^{-3}	4.0×10^{-3}
10^{-4}	1.4×10^{-3}	1.1×10^{-3}
10^{-3}	1.3×10^{-3}	1.0×10^{-3}
10^{-2}	1.2×10^{-3}	8.4×10^{-4}

Table 2.4: The total operation cost and collision probability for the highway scenario.

Radius θ	10^{-4}	10^{-2}	5×10^{-2}	10^{-1}
Cumulative Cost	3222.64	3224.32	3302.63	3796.14
Collision Probability	0.018	0.014	0.008	0.005

Table 2.5: Total operation cost, collision probability, and total computation time for CC-MPC, CVaR-MPC, and DR-MPC.

	CC-MPC	CVaR-MPC	DR-MPC (θ)		
			10^{-5}	10^{-4}	10^{-2}
Cumulative Cost	1.245	3.665	5.707	18.430	30.681
Collision Probability	0.74	0.056	0.034	0.005	0.001
Computation Time (sec)	63.082	71.513	64.786	69.494	74.856



(a) $t = 19$ (b) $t = 60$ (c) $t = 80$

Figure 2.8: Application of learning-based DR-RRT* to a car-like robot on a highway for $\theta = 10^{-4}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}$. The obstacles are shown in green, while their predicted positions are shown in lighter color.

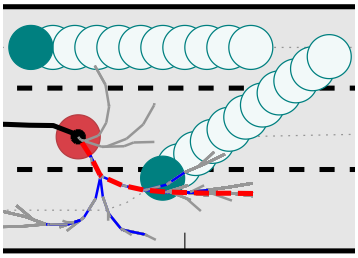
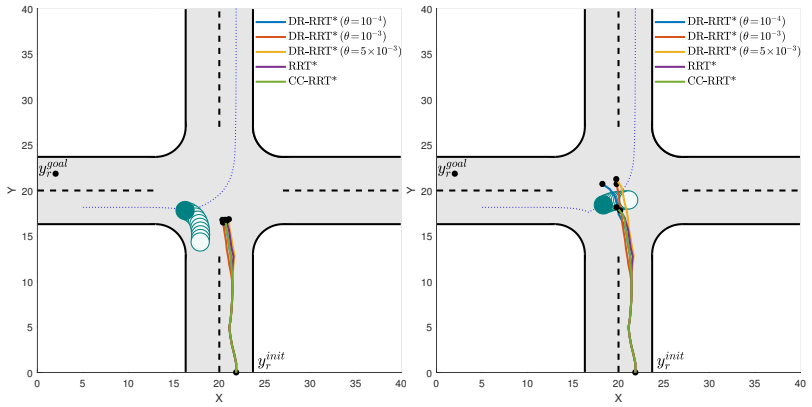
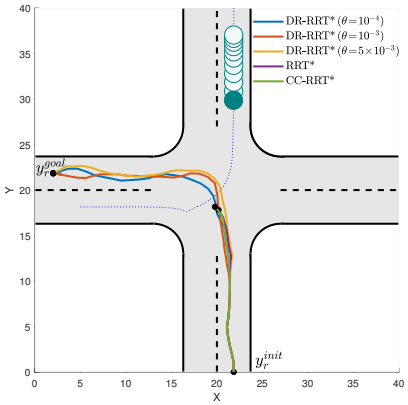


Figure 2.9: Growing process of tree $\mathcal{T}$ (grey) and safe subtree $\mathcal{T}_{\text{safe}}$ (blue) generation. The best path for execution (red) is chosen from $\mathcal{T}_{\text{safe}}$.

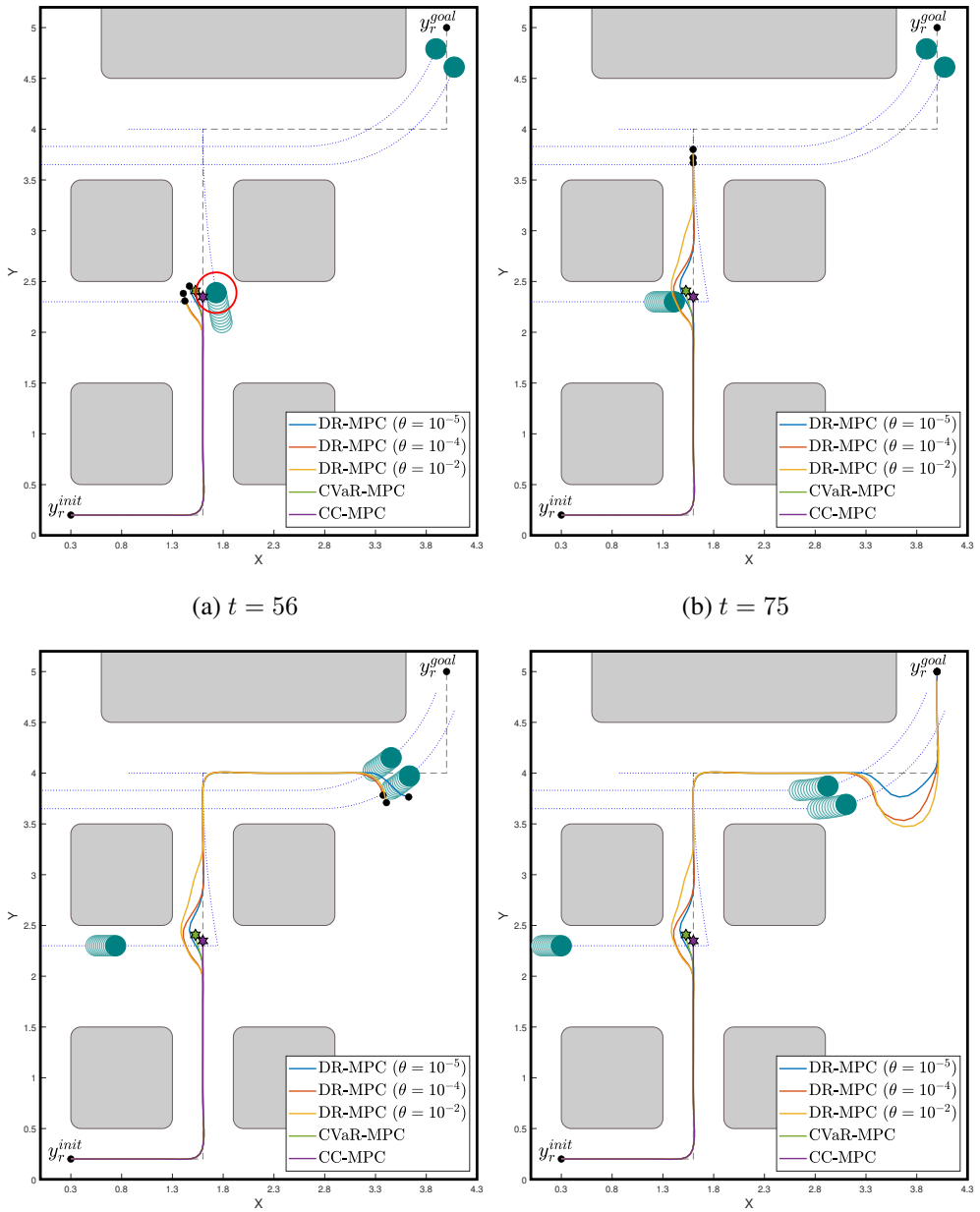


(a) $t = 18$ (b) $t = 23$



(c) $t = 43$

Figure 2.10: Application of learning-based DR-RRT* to a car-like robot in an intersection for $\theta = 10^{-4}, 10^{-3}, 5 \times 10^{-3}$ and comparison with RRT* and CC-RRT*. The obstacle is shown in green, while its predicted positions are shown in lighter color.



(a) $t = 56$ (b) $t = 75$

(c) $t = 110$ (d) $t = 133$

Figure 2.11: Application of learning-based DR-MPC to a car-like robot in a cluttered environment for $\theta = 10^{-5}, 10^{-4}, 10^{-2}$, compared against CC-MPC and CVaR-MPC with $N = 100$. The obstacles are shown in green, while predictions for the corresponding obstacle are in lighter color. Star indicates collision, while the red circle is the collision ball of radius r_ℓ.

Chapter 3

Distributionally Robust Optimization with Unscented Transform for Learning-Based Motion Control in Dynamic Environments

3.1 Introduction

Autonomous mobile robots have shown promise in many real-world applications ranging from indoor services to urban navigation. In general, information about the exact robot model and the environment dynamics is unavailable or highly limited. Learning-based control approaches are commonly used in such settings to infer unknown models and improve the overall control performance. However, the safety of learning-based controllers (e.g., collision-free navigation) remains a significant concern for the application of such methods, especially when the learned models are unreliable and inaccurate [139].

Existing learning-based control methods employ various machine learning techniques to infer the unknown dynamics of the robot and the environment. The learning models most commonly used for this purpose include deep neural networks [47–49, 140, 141], Bayesian linear regression [53, 54, 142], and Gaussian processes (GPs) [51, 52, 143–145], among others. One of the most popular approaches for learning-based

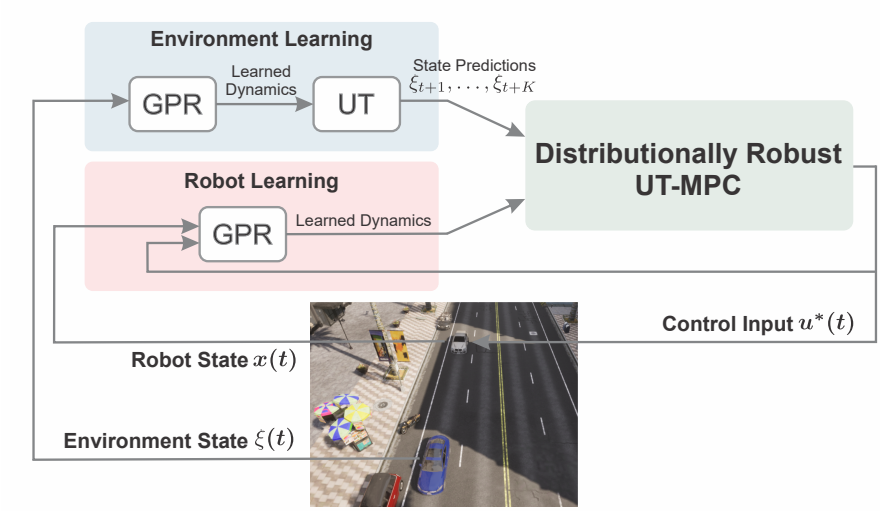


Figure 3.1: The overview of our method.

motion control of robotic systems is model predictive control (MPC), where the unknown models are substituted with the learned ones. Most research efforts in this field have focused on improving the prediction model by learning the system dynamics or fine-tuning its parameters [51, 143–145]. In contrast, a few works learn the dynamic environment model and apply the controller to a system with known dynamics [54, 74, 82, 140, 146–148]. Safety in such methods is often addressed via probabilistic constraints, such as chance constraints [81, 149–152] or conditional value-at-risk (CVaR) constraints [70, 111, 112, 153, 154]. However, most existing methods do not address the learning inaccuracies or unreliability of the models, applying them directly to the controller. Such distributional uncertainties are handled in distributionally robust optimization (DRO) methods, where a given stochastic program is solved in the face of the worst-case distribution drawn from some ambiguity set [39,40,43,155,156]. Recently, the application of DRO has been extended to learning-based control problems to account for learning errors during the control stage [31,43,75,157]. However, these methods require nontrivial computational demand when solving DRO problems.

Our paper is related to learning-based distributionally robust control in that we

learn the unknown dynamics of both the robot and the environment, as well as address the learning errors in the motion control stage by adopting tools from DRO. As shown in Fig. 3.1, our framework consists of (i) separate learning modules for inferring the unknown models of both the robot and the dynamic environment via Gaussian process regression (GPR) [108] and the unscented transform (UT), and (ii) an MPC-based control module that uses the learned models with an accurate uncertainty propagation scheme and is robust against possible learning errors. Unlike typical uncertainty propagation schemes used in GP-based MPC methods [51,143,144], we propose exploiting UT to improve computational efficiency and prediction accuracy. Prior works utilize a similar uncertainty propagation approach in stochastic MPC settings with known system dynamics [158–160]. In contrast, we apply the UT method to the learned models to predict the states of both the robot and the environment. In addition, to immunize the system against learning errors, we adopt tools from Wasserstein DRO and design a risk constraint to limit the distributionally robust CVaR (DR-CVaR) of the safety loss. This leads to a novel distributionally robust UT-based MPC algorithm (UT-MPC), which combines the advantages of both UT and DRO within a single framework. Unfortunately, the DR-CVaR constraint is intractable as it involves an infinite-dimensional optimization problem over the space of probability distributions. To overcome this challenge, we devise a simple analytical upper bound of DR-CVaR that exploits UT to estimate the safety loss distribution. As a result, we obtain a tractable distributionally robust UT-MPC algorithm that guides the robot to take cautious actions despite learning inaccuracies. Finally, the performance and the utility of our method are demonstrated through simulations in an autonomous driving scenario. Our experiments show the capability of our algorithm to promote safe motion control in a dynamic environment, even in the presence of learning errors.

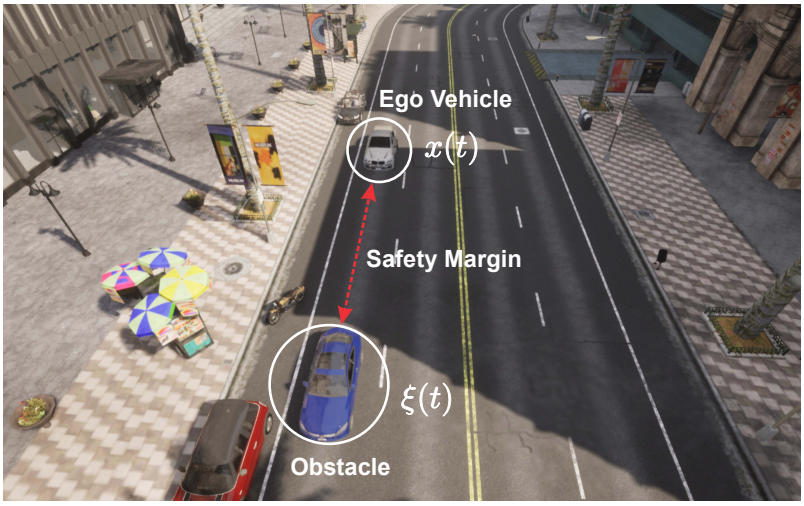


Figure 3.2: An autonomous driving scenario.

3.2 Preliminaries

3.2.1 The Setup

Consider a mobile robot modeled by the following discrete-time dynamics:

$$x(t+1) = f(x(t), u(t)) + g(x(t), u(t)), \tag{3.1}$$

where $x(t) \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $u(t) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ are the robot state and control input at time t, respectively. The dynamic model consists of a known part $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ that can be derived from the physics of the system and an unknown mismatch term $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$, often occurring due to oversimplifying complex dynamics, unexpected interactions with the environment, etc.

Example 3.1. *Consider the autonomous driving scenario in Fig. 3.2. The evolution of the ego vehicle can be described by the kinematic bicycle model, which disregards essential features, such as the slip angles, tire type, as well as driving ground. Therefore, it is reasonable to use the kinematic model with an additional mismatch term to compensate for the limited fidelity of the simple model.*

The robot operates in a dynamic environment, whose state $\xi(t) \in \mathbb{R}^{n_\xi}$ evolves according to

$$\xi(t+1) = f_{\text{env}}(\xi(t)).$$

Such a model is reasonable as the environment evolves independently of the robot. For instance, the safety of the ego vehicle in Example 3.1 depends on the behavior of the blue car with state $\xi(t)$ in Fig. 3.2.

To promote the safe operation of our robot, we introduce a *safety loss function* $\mathcal{J} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\xi} \to \mathbb{R}$ and impose the following constraint:

$$\mathcal{J}(x(t), \xi(t)) \leq 0. \tag{3.2}$$

In Fig. 3.2, the loss can be chosen to avoid collisions, e.g., $\mathcal{J}(x, \xi) = r_{\text{safe}}^2 - \|C(x - \xi)\|_2^2$, where r_{safe} is a safety radius, and C maps the states to the position vector.

In this work, *assuming that the dynamics of the robot and the environment are unknown*, we aim to design a learning-based motion controller that guides the robot to perform a specified task in a cautious manner despite learning errors.

3.2.2 Uncertainty Propagation via UT

When the dynamics (3.1) is learned as a stochastic approximator, the uncertainty in the states is propagated over time. Unfortunately, it is challenging to compute the resulting state distribution for non-Gaussian uncertainties passing through nonlinear dynamics. Linearization techniques from extended Kalman filter (EKF) [161] suffer from large estimation errors and require the computationally expensive Jacobian matrix. An alternative approach is UT, which can be applied to an arbitrary nonlinear function. The intuition behind UT is that with fixed parameters, it is easier to approximate the given distribution than it is to approximate a nonlinear transformation [162]. Therefore, UT aims to find a parameterization that completely encodes the statistics of the inputs, allowing its accurate propagation through a nonlinear function.

Consider a random variable $x \in \mathbb{R}^n$ with a mean vector μ^x and a covariance matrix Σ^x that undergoes a nonlinear transformation $d : \mathbb{R}^n \to \mathbb{R}^m$. The goal of UT is to accurately calculate the statistics of the output $y = d(x)$. For that, first, a set of vectors called sigma points are generated in a way to capture the moments of the input distribution. It has been shown that choosing $2n + 1$ points is sufficient for encoding the mean and covariance of the inputs [163]. The sigma points are selected according to the following rule:

$$\begin{aligned} \mathcal{X}^{(0)} &= \mu^x, \\ \mathcal{X}^{(i)} &= \mu^x + \left(\sqrt{(n+\lambda)\Sigma^x}\right)_i, \ i = 1, \ldots, n, \\ \mathcal{X}^{(n+i)} &= \mu^x - \left(\sqrt{(n+\lambda)\Sigma^x}\right)_i, \ i = 1, \ldots, n, \end{aligned} \tag{3.3}$$

where λ is a scaling parameter and $(\sqrt{\cdot})_i$ is the ith column of the matrix square root. Next, the sigma points are propagated through the nonlinear function to obtain the transformed points $\mathcal{Y}^{(i)} = d(\mathcal{X}^{(i)})$. The mean and covariance of the output y can then be computed as

$$\mu^y = \sum_{i=0}^{2n} W_m^{(i)} \mathcal{Y}^{(i)}, \ \Sigma^y = \sum_{i=0}^{2n} W_c^{(i)} \left(\mathcal{Y}^{(i)} - \mu^y\right) \left(\mathcal{Y}^{(i)} - \mu^y\right)^\top,$$

where $W_m^{(i)}$ and $W_c^{(i)}$ are the weights chosen according to [163].

One of the main advantages of UT is the accuracy of uncertainty propagation. For any nonlinearity, UT captures the output mean and covariance accurately to the third order of the Taylor series expansion for Gaussian inputs and to at least the second order for non-Gaussian inputs [164]. In contrast, the EKF-based method provides only first-order accuracy. Another feature of UT is the implementation simplicity, as it involves only algebraic operations without the need to evaluate the Jacobian matrix needed in EKF.

3.3 Unscented Transform and Distributionally Robust Optimization for Learning-Based Control

The overall structure of our learning-based control scheme is illustrated in Fig. 3.1. It consists of two main parts: (i) separate modules for learning the robot and environment dynamics, and (ii) a distributionally robust UT-MPC module for controlling the robot and addressing learning errors. First, the unknown dynamics are inferred via GPR using real-time observations and then used as prediction models in UT-MPC. However, due to the stochastic nature of the learned dynamics, state propagation through the GP models is not straightforward. Our algorithm mitigates this issue by exploiting UT for uncertainty propagation, achieving superior prediction accuracy and computational efficiency. Moreover, using DRO in UT-MPC immunizes the system against learning inaccuracies and promotes the robot's safety despite erroneous models.

3.3.1 Learning the Robot and Environment Dynamics

In this study, we use GPR, a non-parametric Bayesian regression method, to infer the dynamics of both the robot and the environment. A major challenge in GPR is the uncertainty propagation through the learned model, which is generally intractable. The most typical approach is linearizing the GP model around the current state mean [51, 143, 144]. However, as mentioned in Section 3.2.2, such an approach is not only computationally demanding but also degrades the prediction accuracy. Motivated by the state update equations in GP-UKF [165], we propose an uncertainty propagation scheme for GP dynamics based on the concept of UT. This approach not only improves the prediction accuracy but also involves only simple algebraic operations, relieving the computational burden. Therefore, we apply the proposed scheme to learn the dynamics of both the robot and the environment.

At stage t, GPR for the robot is performed using the training input data $\mathbf{X}_t^{\mathrm{rob}} = \{(x(t-1), u(t-1)), \ldots, (x(t-M), u(t-M))\}$ with the corresponding training output

data $\mathbf{y}_t^{\text{rob}} = \{\Delta x(t), \ldots, \Delta x(t-M+1)\}$, where $\Delta x(t) = x(t+1) - f(x(t), u(t))$ is the residual between the observed system state and the nominal model. Following the ordinary GPR procedure, the unknown dynamics of the robot is approximated by $\mathcal{GP}(\mu^g, \Sigma^g)$. The dynamics of the robot is then inferred as

$$x(t+1) = f(x(t), u(t)) + \mu^g\left(x(t), u(t)\right) + w_t^{\text{rob}}, \tag{3.4}$$

where w_t^{rob} is a zero-mean noise with covariance $\Sigma^g(x(t), u(t))$.

For state prediction, we recursively apply the UT presented in Section 3.2.2 and propagate the states along the horizon. In particular, the distribution of the state vector can be predicted starting from the current observation $\mu_0^x = x(t)$ with $\Sigma_0^x = \mathbf{0}$ according to the following rule:

$$\mathcal{X}_k = \left[\mu_k^x,\ \mu_k^x \pm \sqrt{(n_x + \lambda_x)\Sigma_k^x}\right] \tag{3.5}$$

$$\mathcal{Y}_k^{(i)} = f(\mathcal{X}_k^{(i)}, u_k) + \mu^g(\mathcal{X}_k^{(i)}, u_k),\ i = 0, \ldots, 2n_x \tag{3.6}$$

$$\mu_{k+1}^x = \sum_{i=0}^{2n_x} W_{m_{\text{rob}}}^{(i)} \mathcal{Y}_k^{(i)} \tag{3.7}$$

$$\Sigma_{k+1}^x = \sum_{i=0}^{2n_x} W_{c_{\text{rob}}}^{(i)} \left(\mathcal{Y}_k^{(i)} - \mu_{k+1}^x\right)\left(\mathcal{Y}_k^{(i)} - \mu_{k+1}^x\right)^\top + \Sigma^g(\mu_k^x, u_k). \tag{3.8}$$

Similarly, using datasets $\mathbf{X}_t^{\text{env}} = \{\xi(t-1), \ldots, \xi(t-M)\}$ and $\mathbf{y}_t^{\text{env}} = \{\xi(t), \ldots, \xi(t-M+1)\}$, the dynamics of the environment is approximated by $\mathcal{GP}(\mu^{\text{env}}, \Sigma^{\text{env}})$. Then, the environment states evolve according to

$$\xi(t+1) = \mu^{\text{env}}(\xi(t)) + w_t^{\text{env}},$$

where w_t^{env} is a zero-mean noise with covariance $\Sigma^{\text{env}}(\xi(t))$. By applying a UT scheme similar to (3.5)–(3.8) with weights $W_{m_{\text{env}}}$ and $W_{c_{\text{env}}}$, the environment states can be predicted over the horizon to obtain μ_k^ξ and Σ_k^ξ starting from $\mu_0^\xi = \xi(t)$ and $\Sigma_0^\xi = \mathbf{0}$. Unlike the robot, the environment states are independent of the control inputs. Therefore, as illustrated in Fig. 3.1, the environment state prediction can be performed outside the control loop, saving computational resources.

For our further analysis, it is convenient to denote the joint state of the robot and the environment by $\mathbf{z}_k = [x_k^\top, \xi_k^\top]^\top$. Then, assuming the independence of the states, the estimated joint distribution $\mathbb{P}_k$ of z_k at any time step $t+k$ can be represented by its mean vector $\mu_k^{\mathbf{z}} = [(\mu_k^x)^\top, (\mu_k^\xi)^\top]^\top$ and covariance matrix $\Sigma_k^{\mathbf{z}} = \mathrm{diag}(\Sigma_k^x, \Sigma_k^\xi)$.

3.3.2 Distributionally Robust UT-MPC

Since the state information is no longer deterministic due to the use of GPR, the MPC problem attains a stochastic formulation, where the deterministic constraint (3.2) is not valid anymore. Instead, a risk measure can be used to assess the risk of unsafe events using the learned joint state distribution. Among several risk measures, we use the CVaR, which is a coherent measure in the sense of Artzner et al. [110] and has been advocated as a rational risk measure in robotics [69]. CVaR of a random loss $X \sim \mathbb{P}$ is defined as

$$\mathrm{CVaR}_\epsilon^{\mathbb{P}}[X] := \min_{z\in\mathbb{R}} \mathbb{E}^{\mathbb{P}}\left[z + \frac{(X-z)^+}{1-\epsilon}\right],$$

where $\epsilon \in (0,1]$ is some confidence level. It quantifies the average loss beyond ϵ, accounting for rare but crucial events.

However, the quality of risk assessment highly depends on the accuracy of the learned safety loss distribution. Unfortunately, in our case, the learned information might be unreliable for measuring the robot's safety due to inaccuracies in GP models. To immunize the system against such distributional uncertainties, we propose evaluating the following distributionally robust version of CVaR:

$$\text{DR-CVaR}_\epsilon^{\mathbb{P}_k^{\mathcal{J}}}\left[\mathcal{J}(\mathbf{z}_k)\right] := \sup_{\mathbb{Q}_k\in\mathbb{D}_\theta(\mathbb{P}_k^{\mathcal{J}})} \mathrm{CVaR}_\epsilon^{\mathbb{Q}_k}\left[\mathcal{J}(\mathbf{z}_k)\right], \tag{3.9}$$

which evaluates the worst-case CVaR over an ambiguity set $\mathbb{D}_\theta(\mathbb{P}_k^{\mathcal{J}})$ constructed using the learned safety loss distribution $\mathbb{P}_k^{\mathcal{J}}$. In this work, we define the ambiguity set as a Wasserstein ball of radius $\theta > 0$ centered at $\mathbb{P}_k^{\mathcal{J}}$:

$$\mathbb{D}_\theta(\mathbb{P}_k^{\mathcal{J}}) = \left\{\mathbb{Q}_k \in \mathcal{P}_2(\mathbb{R}) \mid W_2(\mathbb{P}_k^{\mathcal{J}}, \mathbb{Q}_k) \le \theta\right\}, \tag{3.10}$$

where $\mathcal{P}_2(\mathcal{W})$ is the space of Borel probability measures on $\mathcal{W}$ with a finite second moment. Here, $W_2(\mathbb{P},\mathbb{Q})$ is the 2-Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$, which is defined as

$$W_2(\mathbb{P},\mathbb{Q}) := \inf_{\kappa\in\mathcal{P}(\mathcal{W}^2)} \left\{ \left(\int_{\mathcal{W}^2} \|x-y\|^2 \,\mathrm{d}\kappa(x,y) \right)^{1/2} \Big|\, \Pi^1\kappa = \mathbb{P}, \Pi^2\kappa = \mathbb{Q} \right\},$$

where κ is the *transport plan*, with $\Pi^i\kappa$ denoting its ith marginal, and $\|\cdot\|$ is the Euclidean norm quantifying the transportation cost. Wasserstein distance represents the minimum cost of transporting mass from one distribution to another using nonuniform perturbations. It has received great interest in DRO for its superior features, such as providing a finite-sample performance guarantee and addressing the closeness between two points in the support [30, 40, 41].

Combining the UT-based GP dynamics (3.5)–(3.8) and the DR-CVaR risk (3.9), we formulate the following distributionally robust UT-MPC problem:

$$\min_{\mathbf{u}} \sum_{k=0}^{K-1} \mathbb{E}^{\mathbb{P}_k}\left[c(x_k,u_k)\right] + \mathbb{E}^{\mathbb{P}_K}\left[q(x_K)\right] \tag{3.11a}$$

$$\text{s.t. } (3.5)-(3.8) \tag{3.11b}$$

$$\mathrm{DR\text{-}CVaR}_{\epsilon}^{\mathbb{P}_k^{\mathcal{J}}}\left[\mathcal{J}(\mathbf{z}_k)\right] \le 0 \tag{3.11c}$$

$$\mu_k^x \in \mathbb{X} \tag{3.11d}$$

$$u_k \in \mathbb{U} \tag{3.11e}$$

$$\mu_0^x = x(t), \Sigma_0^x = \mathbf{0}, \tag{3.11f}$$

where $c : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u} \to \mathbb{R}$ is the stage-wise cost function, and $q : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost function. Here, the constraints (3.11b) and (3.11e) hold for $k = 0,\ldots,K-1$, while the constraints (3.11c) and (3.11d) hold for $k = 0,\ldots,K$.

The constraint on UT-based GP dynamics (3.11b) plays an important role in our distributionally robust UT-MPC problem. First, the UT-based state propagation scheme provides better state prediction accuracy than the linearization technique often met in prior GP-based MPC approaches [51, 143, 144]. Second, the nonconvexities in the

equality constraints involve simple algebraic operations and, thus, are relatively easy to handle than the derivatives in linearization. Another key component of our MPC problem is the DR-CVaR constraint (3.11c). It limits the safety risk under the worst-case distribution within the ambiguity set of the learned loss distribution for all time stages. Notably, adjusting the radius θ changes the conservativeness of the constraint, as it determines the range of distributions in the neighborhood of $\mathbb{P}_k^{\mathcal{J}}$ to be included in the ambiguity set. In summary, the combination of UT-based GP dynamics and DR-CVaR risk constraint reinforces our distributionally robust UT-MPC problem with superior prediction accuracy and computational efficiency, as well as the capability of limiting the safety risk despite learning inaccuracies.

3.4 Tractable Reformulation and Algorithm

Despite the advantages of the distributionally robust UT-MPC problem (3.11), it is intractable due to the objective function (3.11a) and the safety constraint (3.11c). The objective function can be handled relatively easily by approximating it around the predicted state mean. Our primary concern is the DR-CVaR in constraint (3.11c), which is challenging to evaluate as it involves an infinite-dimensional optimization problem over the ambiguity set of probability distributions. In our method, we overcome the intractability of the MPC problem using a novel UT-based approximation scheme. Specifically, we take advantage of UT to estimate the statistics of the safety loss distribution. Then, we use the approximate distribution and modern tools from DRO to derive an upper bound of DR-CVaR. As a consequence, we arrive at a tractable distributionally robust UT-MPC algorithm.

3.4.1 UT-Based Upper Bound of DR-CVaR

The Wasserstein ambiguity set in (3.10) is built around the learned loss distribution $\mathbb{P}_k^{\mathcal{J}}$. However, in each prediction step k, we are given only the mean and covari-

ance of the joint state vector $\mathbf{z}_k$. Therefore, our first goal is to determine $\mathbb{P}_k^{\mathcal{J}}$ by propagating $\mathbf{z}_k$ through the loss function. Fortunately, we can apply the UT-based uncertainty propagation scheme in Section 3.2.2 to directly estimate the statistics of the loss distribution. For that, we first generate sigma points $\mathcal{Z}_k$ for $\mu_k^{\mathbf{z}}$ and $\Sigma_k^{\mathbf{z}}$ according to (3.3), pass them through the loss function, and obtain transformed points $\mathcal{L}_k^{(i)} = \mathcal{J}\big(\mathcal{Z}_k^{(i)}\big),\ i = 0, \dots, 2(n_x + n_\xi)$. Then, the mean and the variance of the loss can be obtained as

$$\mu_k^{\mathcal{J}} = \sum_{i=0}^{2(n_x+n_\xi)} W_{m_{\text{loss}}}^{(i)} \mathcal{L}_k^{(i)} \tag{3.12}$$

$$\big(\sigma_k^{\mathcal{J}}\big)^2 = \sum_{i=0}^{2(n_x+n_\xi)} W_{c_{\text{loss}}}^{(i)} \big(\mathcal{L}_k^{(i)} - \mu_k^{\mathcal{J}}\big)\big(\mathcal{L}_k^{(i)} - \mu_k^{\mathcal{J}}\big)^\top. \tag{3.13}$$

Though there is still no full knowledge about the distribution $\mathbb{P}_k^{\mathcal{J}}$, UT provides us with knowledge about its mean and variance. In the following proposition, we show how this statistical information can be used to obtain a tractable and simple upper bound on DR-CVaR.

Proposition 3.1. *Let $\mathbb{P}_k^{\mathcal{J}}$ be the distribution of the loss $\mathcal{J}(\mathbf{z}_k)$ with mean and variance defined in* (3.12) *and* (3.13)*, respectively. Then, the DR-CVaR* (3.9) *with a radius $\theta > 0$ has the following upper bound:*

$$\text{DR-CVaR}_\epsilon^{\mathbb{P}_k^{\mathcal{J}}}[\mathcal{J}(\mathbf{z}_k)] \leq \mu_k^\ell + \gamma\sigma_k^{\mathcal{J}} + \theta\sqrt{1+\gamma^2}, \tag{3.14}$$

where $\gamma = \sqrt{\epsilon/(1-\epsilon)}$.

Proof. We use the Gelbrich bound on Wasserstein distance, for which

$$W_2(\mathbb{P}_k^{\mathcal{J}}, \mathbb{Q}_k) \geq \sqrt{(\mu_k - \mu_k^{\mathcal{J}})^2 + (\sigma_k - \sigma_k^{\mathcal{J}})^2},$$

where $\mu_k \in \mathbb{R}$ and $\sigma_k^2 \in \mathbb{R}_+$ are the mean and variance of the loss under the distribution $\mathbb{Q}_k$ [166, Proposition 8]. The bound is exact if $\mathbb{P}_k^{\mathcal{J}}$ and $\mathbb{Q}_k$ are elliptical distributions with the same density generator. It follows that the DR-CVaR is bounded

as

$$\text{DR-CVaR}_{\epsilon}^{\mathbb{P}_k^{\mathcal{J}}}\left[\mathcal{J}(\mathbf{z}_k)\right] \leq \sup_{\mathbb{Q}_k \in \tilde{\mathbb{D}}_\theta(\mathbb{P}_k^{\mathcal{J}})} \text{CVaR}_{\epsilon}^{\mathbb{Q}_k}\left[\mathcal{J}(\mathbf{z}_k)\right],$$

where $\tilde{\mathbb{D}}_\theta(\mathbb{P}_k^{\mathcal{J}})$ is an ambiguity set with respect to the Gelbrich bound defined as

$$\tilde{\mathbb{D}}_\theta(\mathbb{P}_k^{\mathcal{J}}) := \big\{\mathbb{Q}_k \in \mathcal{P}_2(\mathbb{R}) \mid (\mu, \sigma^2) \in \mathcal{U}_\theta(\mu_k^{\mathcal{J}}, (\sigma_k^{\mathcal{J}})^2), \\ \mathbb{E}^{\mathbb{Q}_k}[\mathcal{J}(\mathbf{z}_k)] = \mu, \mathbb{E}^{\mathbb{Q}_k}\left[(\mathcal{J}(\mathbf{z}_k) - \mu)^2\right] = \sigma^2\big\},$$

and

$$\mathcal{U}_\theta(\mu_k^{\mathcal{J}}, (\sigma_k^{\mathcal{J}})^2) := \big\{(\mu, \sigma^2) \in \mathbb{R} \times \mathbb{R}_+ \mid (\mu - \mu_k^{\mathcal{J}})^2 + (\sigma - \sigma_k^{\mathcal{J}})^2 \leq \theta^2\big\}$$

is the mean-covariance uncertainty set around the estimated mean μ_k^{ℓ} and variance $(\sigma_k^{\mathcal{J}})^2$.

In order to solve the right-hand side of the inequality, also known as the *Gelbrich risk*, we decompose it into

$$\sup_{\mathbb{Q}_k \in \tilde{\mathbb{D}}_\theta(\mathbb{P}_k^{\mathcal{J}})} \text{CVaR}_{\epsilon}^{\mathbb{Q}_k}\left[\mathcal{J}(\mathbf{z}_k)\right] \\ = \sup_{(\mu_k, \sigma_k^2) \in \mathcal{U}_\theta(\mu_k^{\mathcal{J}}, (\sigma_k^{\mathcal{J}})^2)} \sup_{\mathbb{Q}_k \in \mathcal{C}(\mu_k, \sigma_k^2)} \text{CVaR}_{\epsilon}^{\mathbb{Q}_k}\left[\mathcal{J}(\mathbf{z}_k)\right], \tag{3.15}$$

where $\mathcal{C}(\mu, \sigma^2)$ is the Chebyshev uncertainty set with mean μ and variance σ^2.

To solve the inner supremum, we apply [167, Proposition 2] according to which

$$\sup_{\mathbb{Q}_k \in \mathcal{C}(\mu_k, \sigma_k^2)} \text{CVaR}_{\epsilon}^{\mathbb{Q}_k}\left[\mathcal{J}(\mathbf{z}_k)\right] = \mu_k + \gamma\sigma_k.$$

By substituting the above solution into (3.15), the problem reduces to the following convex optimization problem, which is a quadratically constrained quadratic program:

$$\max_{\mu_k, \sigma_k \geq 0} \{\mu_k + \gamma\sigma_k \mid (\mu_k - \mu_k^{\mathcal{J}})^2 + (\sigma_k - \sigma_k^{\mathcal{J}})^2 \leq \theta^2\}.$$

Using the standard duality, the solution of the above optimization problem corresponds to the right-hand side of (3.14). □

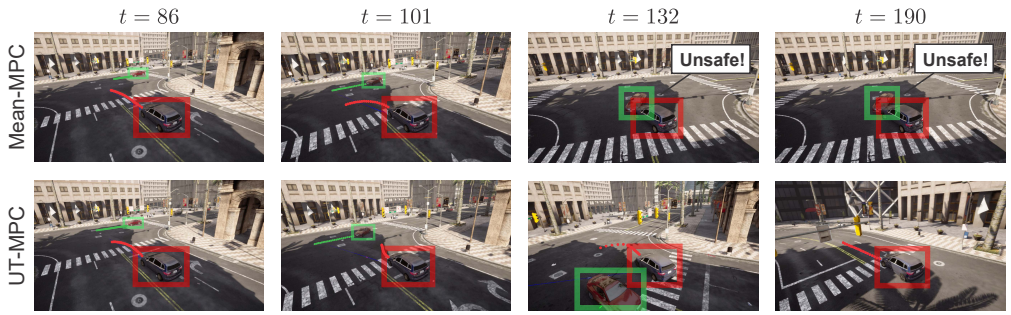


Figure 3.3: Snapshots of simulations for Mean-MPC and UT-MPC. The MPC predictions for the ego vehicle are shown in red, while the GP predictions for the obstacle are drawn in green.

The upper bound (3.14) is attained for elliptical distributions and is tight for all other distributions. Despite relying solely on the mean and variance, this bound exhibits exceptional computational properties, as it requires simple algebraic operations. Moreover, unlike the existing methods (e.g., [31, 75]), we directly estimate the loss distribution $\mathbb{P}_k^{\mathcal{J}}$, enabling the use of our approach for any safety loss function.

3.4.2 Tractable Algorithm

The UT-based upper bound of DR-CVaR can be directly incorporated into the distributionally robust UT-MPC problem (3.11) to alleviate the intractability without significantly affecting the computational complexity. For that, we replace the risk constraint (3.11c) with a constraint on the upper bound (3.14) and introduce additional equality constraints (3.12) and (3.13) for estimating the loss distribution. As a result, the reformulated MPC problem constitutes a tractable nonlinear optimization problem. Despite its nonconvexity due to the GP dynamics and the UT approximations, it can be efficiently solved using existing algorithms, such as interior-point and sequential quadratic programming methods [168].

The overall distributionally robust UT-MPC scheme is presented in Algorithm 3, given the UT weights $W_{m_i}, W_{c_i}, i = \{\text{rob}, \text{env}, \text{loss}\}$, as well as risk parameters ϵ

Algorithm 3: Distributionally Robust UT-MPC

1 **Input:** UT parameters $W_{m_i}, W_{c_i}, i = \{\text{rob}, \text{env}, \text{loss}\}$ and risk parameters ϵ, θ

2 Collect M observations to $\mathbf{X}_0^{\text{rob}}, \mathbf{y}_0^{\text{rob}}, \mathbf{X}_0^{\text{env}}, \mathbf{y}_0^{\text{env}}$

3 Observe $x(0)$ and $\xi(0)$

4 **for** $t = 0, 1, \ldots$ **do**

5 Train GPs for μ^g, Σ^g and $\mu^{\text{env}}, \Sigma^{\text{env}}$

6 Predict μ_k^ξ and Σ_k^ξ for $k = 0, \ldots, K-1$ starting from $\mu_0^\xi = \xi(t), \Sigma_0^\xi = \mathbf{0}$

7 Solve problem (3.11) with (3.14)

8 Apply $u(t) = u_0^*$ and observe $x(t+1), \xi(t+1)$

9 Update $\mathbf{X}_{t+1}^{\text{rob}}, \mathbf{y}_{t+1}^{\text{rob}}$ and $\mathbf{X}_{t+1}^{\text{env}}, \mathbf{y}_{t+1}^{\text{env}}$

and θ. First, GPR training datasets $\mathbf{X}_0^{\text{rob}}, \mathbf{y}_0^{\text{rob}}$ and $\mathbf{X}_0^{\text{env}}, \mathbf{y}_0^{\text{env}}$ are initialized by collecting M observations (line 2). Next, the states of the robot and the environment are observed to begin the main loop (line 3). In each time stage, GP models for the robot and environment are learned (line 5). Then, we predict the environment states for K time stages starting from the current state $\xi(t)$ (line 6). Using the learned models, the UT-MPC problem (3.11) is solved with the DR-CVaR upper bound (3.14) (line 7). The first element of the optimal control sequence $\mathbf{u}^*$ returned by the UT-MPC is then applied to the robot (line 8). Finally, we observe the new states and update the GPR datasets with the latest M observations (line 9).

3.5 Experiment Results

In this section, we present the simulation results of our algorithm in an autonomous driving scenario performed in an open-source traffic simulation platform CARLA [169]. The goal is to control the ego vehicle to follow the given waypoints without colliding

with the obstacle. The source code of our implementation is available online.[9]

3.5.1 Experiment Settings

In our experiments, the nominal model of the ego vehicle is chosen as the following kinematic bicycle model:

$$\begin{aligned}
\mathrm{x}(t+1) &= \mathrm{x}(t) + T_s v(t)\cos(\phi(t) + \beta_s(t)) \\
\mathrm{y}(t+1) &= \mathrm{y}(t) + T_s v(t)\sin(\phi(t) + \beta_s(t)) \\
\phi(t+1) &= \phi(t) + T_s v(t)\tan(\delta_f(t))\cos(\beta_s(t))/L \\
v(t+1) &= v(t) + T_s a(t),
\end{aligned}$$

where $x(t) = [\mathrm{x}(t), \mathrm{y}(t), \phi(t), v(t)]^\top$ is the ego vehicle's state vector, consisting of its position, heading angle and velocity, $u(t) = [a(t), \delta_f(t)]^\top$ is the control input vector, comprising acceleration and steering angle, $\beta_s(t) := \arctan\left(\frac{1}{2}\tan(\delta_f(t))\right)$ is the slipping angle, $T_s = 0.1\,\mathrm{sec.}$ is the sampling time, and $L = 4.611\,\mathrm{m.}$ is the car length. The control inputs are limited to $|a(t)| \leq 3\,\mathrm{m/sec.}^2, |\delta_f(t)| \leq 1.22\,\mathrm{rad.}$ with an additional limit on the change of front steering angle $|\Delta\delta_f(t)| \leq 0.05\,\mathrm{rad.}$ The cost function is chosen to track the waypoints p_t and penalize control input changes, i.e.,

$$c(x_k, u_k) = \|x_k - p_{t+k}\|_Q^2 + \|\Delta u_k\|_R^2, \quad q(x_K) = \|x_K - p_{t+K}\|_Q^2,$$

where $Q = \mathrm{diag}(1, 1, 0, 0.2)$ and $R = \mathrm{diag}(1.5, 3)$. We consider an MPC with a horizon of $K = 30$ and a zero-mean GPR with a radial basis function kernel trained on $M = 50$ real-time observations. The parameters for DR-CVaR are tuned to $\epsilon = 0.95$ and $\theta = 0.1$.

Due to the simulation model, the obstacle's behavior is not deterministic and varies in each execution. Therefore, for reliability, we have performed 20 simulation runs under identical conditions. We compare our method to an MPC without any learning component (Vanilla-MPC), a learning-based MPC with the safety constraint (3.2)

[9] `https://github.com/CORE-SNU/DR-UT-MPC`

Table 3.1: The total cost, average computation time per stage, and maximum safety loss value for for all algorithm computed over 20 simulations (mean $\pm$ std).

	Total Cost ($\times 10^3$)	Comp. Time (sec)	Safety Loss (m^2)
UT-MPC	28.056 ± 0.204	0.467 ± 0.013	1.864 ± 0.133
Vanilla-MPC	∞	0.031 ± 0.002	12.342 ± 0.742
Mean-MPC	∞	0.435 ± 0.001	9.314 ± 1.054
CVaR-MPC	29.356 ± 0.142	0.418 ± 0.023	2.032 ± 0.101

evaluated at the mean of the predicted state (Mean-MPC), and a non-robust version of UT-MPC with a CVaR constraint (CVaR-MPC).

3.5.2 Results

Snapshots of representative scenarios for distributionally robust UT-MPC and Mean-MPC are demonstrated in Fig. 3.3. Initially, the obstacle navigates far from the ego vehicle and plans to continue in the same lane. Therefore, in the early stages, both controllers drive the car along the reference path. However, when the vehicles approach the intersection, the obstacle suddenly steers to the left. This situation causes errors in the GPR prediction, making the learned distribution unreliable. Nevertheless, the Mean-MPC trusts the learned information even in such a situation and decides to perform a cut-in maneuver, eventually leading to a collision between the ego vehicle and the obstacle. On the contrary, UT-MPC makes the car stop at the intersection and then slowly bypass the obstacle from the right. Due to its robustness to learning errors, UT-MPC takes cautious actions and overtakes the obstacle without any collisions. Consequently, it outperforms Mean-MPC in terms of navigation quality and safety.

The statistics of our quantitative analysis for 20 simulation runs are reported in Table 3.1. In terms of safety, our algorithm outperforms all the baselines, followed

by the CVaR-MPC. Such results are expected, as UT-MPC is the only method that accounts for learning errors. On the other hand, both Vanilla-MPC and Mean-MPC become infeasible after colliding with the obstacles, making the total cost infinitely large. In terms of computation time, Vanilla-MPC surpasses all the baselines due to its simplicity. Meanwhile, all the learning-based algorithms, including our UT-MPC, require similar computation time for solving the problem. As a result, we confirm the capabilities of our algorithm for promoting safety with a comparably short computation time.

3.6 Conclusions

We have proposed a novel learning-based MPC framework for robotic systems in unknown environments. Our method exploits the learned dynamics and UT-based uncertainty propagation scheme for accurate and efficient prediction of the robot and environment states. Furthermore, it uses a DR-CVaR constraint to proactively limit the risk of unsafety even under errors in the learned models. To tackle the computational intractability of the resulting UT-MPC problem, we have approximated the safety loss distribution using UT and derived a simple upper bound of DR-CVaR. The experiment results demonstrate the computational efficiency of our method and its capability to promote safety.

Chapter 4

Wasserstein Distributionally Robust Control of Partially Observable Linear Stochastic Systems

4.1 Introduction

Optimal control of linear dynamical systems under uncertainties has a long history and is regarded as one of the most fundamental topics in control theory [170]. In various practical systems, the system states are not entirely observable, and there is only partial information available about the system coming from the noisy measurements. The theory of optimal control handles such imperfect state information either in stochastic or robust control frameworks. Robust optimal control methods address uncertainties in a pre-specified disturbance set and seek to find a controller concerning the worst-case realization of the disturbance (e.g., [15]). However, the resulting controllers are often conservative as no information other than the support of disturbances is used, and potentially useful statistical properties of the disturbances are disregarded. On the contrary, stochastic optimal control approaches design a controller using the knowledge of the disturbance distribution, which is typically modeled as Gaussian (e.g., [171]). However, it is often difficult to obtain an accurate probability distribution of disturbances. Using imperfect distributional information does not guar-

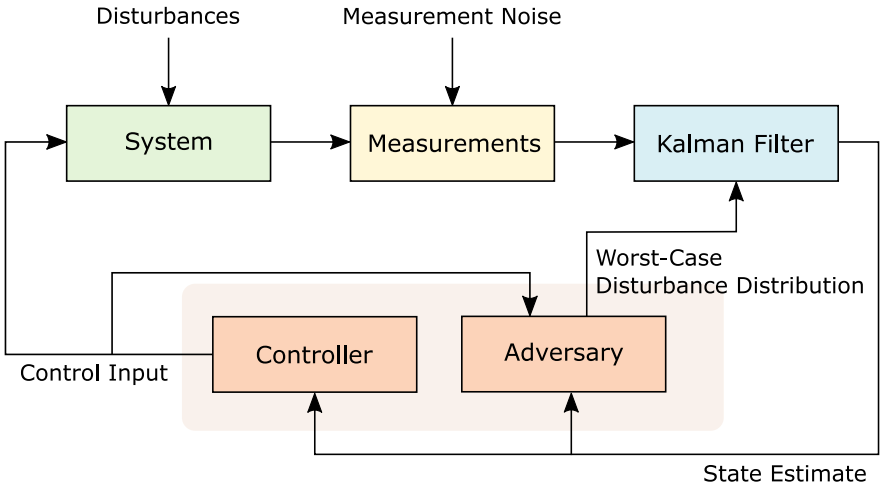


Figure 4.1: Block diagram of the proposed WDRC scheme.

antee the optimality of the resulting controller and may even cause undesirable system behaviors (e.g., [172, 173]).

To alleviate the aforementioned issues and bridge the gap between the two methods, distributionally robust control (DRC) has emerged as an alternative tool, balancing the tradeoff between required information and conservativeness [27, 28, 30, 31, 33, 42, 44, 45, 174–182]. With DRC, a controller is designed to minimize the expected cost of interest with respect to the worst-case probability distribution of disturbances in a so-called *ambiguity set*. Thus, the resulting controller proactively manages possible deviations of the true distribution from the nominal one used in the controller design.

DRC can be regarded as a dynamic or multi-stage version of distributionally robust optimization (DRO). In the literature regarding DRO, it is common to design the ambiguity set based on a nominal distribution constructed from data so that it contains the true distribution with high probability. For example, moment-based ambiguity sets are popular in DRO, which include distributions satisfying some moment constraints [36, 37, 155]. Despite outstanding tractability properties, such sets often yield conservative decisions and require accurate moment estimates. Designing the ambigu-

ity set based on statistical distances to contain distributions close to the given nominal one is another popular option. Among various distances, such as the KL-divergence and Prokhorov metric [38, 183], the Wasserstein metric attracts significant attention not only in DRO [39–41, 184] but also in DRC [29–33, 42]. The Wasserstein ambiguity set has a number of useful features, including offering a powerful finite-sample performance guarantee [39, 43]. Furthermore, it is rich enough to contain relevant distributions, thereby encouraging the DRO problem to avoid providing pathological solutions [40].

In contrast to research on fully observable settings, the literature about partially observable DRC is relatively sparse. A few works are devoted to the distributionally robust version of the linear-quadratic-Gaussian (LQG) control method. For example, [17, 44, 45] propose a minimax LQG controller that minimizes the worst-case performance by restricting the KL-divergence between the disturbance distribution and a given reference distribution. In [46], a partially observable Markov decision process is considered with finite state, action, and observation spaces. The ambiguity set is chosen to bound the moments of the joint distribution of the transition-observation probabilities. Another type of partially observable systems, namely the Markov jump linear system, is studied in [28]. The authors propose a mechanism for estimating the active mode in a receding horizon fashion and integrate this procedure with a data-driven distributionally robust controller design using the total variation distance. In [31], a data-enabled distributionally robust predictive control method is proposed and studied using noise-corrupted input and output data.

Departing from the existing literature, our particular interest is in the Wasserstein DRC (WDRC) methods for partially observable linear-quadratic optimal control in discrete time, motivated by the superior properties of Wasserstein DRO. The WDRC problem is challenging to solve due to partial observability in addition to the infinite-dimensionality of the Wasserstein DRO problem in the Bellman equation. To resolve these issues, we propose a novel approximation technique for partially observable

WDRC problems by replacing the Wasserstein ambiguity set with a special penalty term using the Gelbrich bound. The approximate problem is first solved in the finite-horizon setting by deriving a non-trivial Riccati equation alongside a closed-form expression for the optimal control policy. Then, we examine the asymptotic behavior of the controller and extend the results to the infinite-horizon average-cost setting. Consequently, we obtain optimal control and distribution policies by solving an algebraic Riccati equation (ARE) and a tractable semidefinite programming (SDP) problem. The overall scheme of the proposed WDRC method is illustrated in Fig. 4.1.

The proposed controller possesses several salient theoretical properties. First, it is shown to enjoy a guaranteed cost property for any worst-case disturbance distribution in the Wasserstein ambiguity set. This demonstrates the distributional robustness of our controller despite being constructed by solving an approximate WDRC problem. Second, the proposed controller offers a probabilistic out-of-sample performance guarantee. Last but not least, the proposed controller is shown to ensure the stability of the closed-loop mean-state system as well as its bounded-input, bounded-output (BIBO) stability when viewing the disturbances as input.

The rest of this article is organized as follows. In Section 4.2, we introduce the partially observable WDRC problem for linear systems. In Section 4.3, we introduce the tractable approximation and derive its solution in both finite- and infinite-horizon average-cost settings. In addition, we analyze the optimality of the resulting solution and describe the overall WDRC algorithm. In Section 4.4, we present the guaranteed cost property and out-of-sample performance guarantee of our controller. Section 4.5 concerns the stability properties of the closed-loop mean-state system. Finally, Section 4.6 demonstrates the performance and utility of the proposed method through numerical experiments on a power system frequency control problem.

4.2 Preliminaries

4.2.1 Notation

We let $\mathcal{P}(\mathcal{W})$ denote the set of Borel probability measures with support $\mathcal{W}$. The expected value of function $f(x)$, where x is a random variable with a probability distribution $\mathbb{P}$, is denoted by $\mathbb{E}_x[f(x)]$. We denote the space of all symmetric matrices in $\mathbb{R}^{n\times n}$ by $\mathbb{S}^n$. In addition, $\mathbb{S}^n_+$ represents the cone of all symmetric positive semidefinite (PSD) matrices in $\mathbb{S}^n$ with $\mathbb{S}^n_{++}$ denoting its subset of symmetric positive definite (PD) matrices. For any $A, B \in \mathbb{S}^n_+$, the relation $A \succeq B (A \succ B)$ means that $A - B \in \mathbb{S}^n_+ (A - B \in \mathbb{S}_{++})$.

4.2.2 Problem Setup

Consider the following discrete-time linear stochastic system:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w_t \\ y_t &= Cx_t + v_t, \end{aligned} \tag{4.1}$$

where $x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, and $y_t \in \mathbb{R}^{n_y}$ are the system state, control input, and output at stage t, respectively. Here, $w_t \in \mathbb{R}^{n_x}$ represents the system disturbance with unknown distribution, while $v_t \in \mathbb{R}^{n_y}$ is the output noise drawn from a zero-mean Gaussian distribution with covariance matrix M. The initial state x_0 is also random, drawn from a probability distribution with known mean vector m_0 and covariance matrix M_0. We assume the independence of w_s and w_t and that of v_s and v_t for any $s \neq t$. Moreover, the random vectors w_t, v_t, and x_t are assumed to be independent.

Unlike the fully observable setting, the only information available at time t is the history of noisy measurements $y_0, \ldots, y_t$ and the past control inputs $u_0, \ldots, u_{t-1}$. Therefore, the information given to the controller at time t can be represented as

$$\begin{aligned} I_t &:= (y_0, \ldots, y_t, u_0, \ldots, u_{t-1}), \quad t = 1, 2, \ldots, \\ I_0 &:= y_0, \end{aligned}$$

where I_t is called the *information vector*. Note that the information vector is updated according to the following dynamical system:

$$I_{t+1} = (I_t, y_{t+1}, u_t). \tag{4.2}$$

In the theory of stochastic optimal control, it is well-known that the information vector serves as a sufficient statistic. Thus, it suffices to consider control policies π_t that map I_t to a control input u_t for each t. The dynamics (4.2) can be viewed as describing the evolution of a system where the state is the information vector I_t and the control is u_t. The system output y_{t+1} plays the role of a stochastic disturbance due to its dependence on system disturbance w_t and measurement noise v_{t+1}, introducing randomness and impacting the dynamics of the augmented system through the measured variables.

In many practical problems, the probability distributions of output noise and initial state are given a priori (e.g., known sensor noise). In contrast, the distribution of the system disturbances is usually unknown (e.g., unmodelled dynamics). For simplicity, the disturbance distribution is often assumed to be Gaussian or estimated from data. However, when this assumption is invalid, the imperfect distributional information can deteriorate the controller's performance, especially when it has to operate for an infinite amount of time. Thus, our goal is to design a control policy that is robust against deviations of the true disturbance distribution from the given nominal one. In the literature of DRO, such distributional uncertainties are captured by a set of probability distributions $\mathcal{D}_t \subset \mathcal{P}(\mathbb{R}^{n_x})$, called the *ambiguity set*. It encompasses prior information about the underlying true distribution and includes distributions with shared structural information. As a result, we consider a distribution policy γ_t that maps I_t to a probability distribution $\mathbb{P}_t$ of w_t, chosen from the ambiguity set $\mathcal{D}_t$.

Now, consider the following finite-horizon quadratic cost function:

$$J_T(\pi, \gamma) := \mathbb{E}_{\mathbf{y}}\Big[\mathbb{E}_{x_T}[x_T^\top Q_f x_T \mid I_T] + \sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t]\Big],$$

where $\pi := (\pi_0, \ldots, \pi_{T-1})$ and $\gamma := (\gamma_0, \ldots, \gamma_{T-1})$, $Q \in \mathbb{S}_+^{n_x}, Q_f \in \mathbb{S}_+^{n_x}, R \in \mathbb{S}_{++}^{n_u}$ are the cost weights, and the outer expectation is taken with respect to the joint distri-

bution of all measurements $\mathbf{y} := (y_0, \ldots, y_T)$. Since our eventual goal is to design a controller for the infinite-horizon case, we define the following average-cost criterion:

$$J_\infty(\pi, \gamma) = \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}\left[\sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t]\right]. \tag{4.3}$$

The DRC problem can be formulated as a two-player zero-sum game, where the first player is the controller and the second player is the adversary. The controller selects a policy $\pi = (\pi_0, \pi_1, \ldots)$ to minimize the cost, while the adversary player aims to find a distribution policy $\gamma = (\gamma_0, \gamma_1, \ldots)$ to maximize the same cost. More precisely, we aim to solve the following minimax stochastic control problem:

$$\min_{\pi\in\Pi} \max_{\gamma\in\Gamma_{\mathcal{D}}} J_\infty(\pi, \gamma), \tag{4.4}$$

where $\Pi := \{\pi \mid \pi_t(I_t) = u_t, \pi_t \text{ is measurable } \forall t\}$ and $\Gamma_{\mathcal{D}} := \{\gamma \mid \gamma_t(I_t) = \mathbb{P}_t \in \mathcal{D}_t, \gamma_t \text{ is measurable } \forall t\}$ are the sets of admissible control and distribution policies. Note that the ambiguity set is embedded in the policy space for the adversary, and thus the ambiguity set plays a critical role in characterizing the distributional inaccuracies that are proactively addressed by the controller.

4.2.3 Wasserstein Ambiguity Set

Motivated by the superior properties of Wasserstein DRO mentioned in Section 4.1, we choose $\mathcal{D}_t$ as a Wasserstein ball. The Wasserstein metric of order p between two measures $\mathbb{P}$ and $\mathbb{Q}$ supported on $\mathcal{W} \subseteq \mathbb{R}^n$ quantifies the minimum cost of redistributing mass from one measure to another using non-uniform perturbations and is defined as

$$W_p(\mathbb{P}, \mathbb{Q}) := \inf_{\tau\in\mathcal{T}(\mathbb{P},\mathbb{Q})} \left\{ \left(\int_{\mathcal{W}^2} \|w - w'\|^p \mathrm{d}\tau(x, y)\right)^{1/p} \right\},$$

where $\mathcal{T}(\mathbb{P}, \mathbb{Q})$ is the set of all measures in $\mathcal{P}(\mathcal{W}^2)$ with the first and second marginals $\mathbb{P}$ and $\mathbb{Q}$, respectively. Here, τ is called the *transport plan*, which describes the amount of mass to move from w to w', and $\|\cdot\|$ is a norm on $\mathbb{R}^n$ that measures the transportation cost.

Using the Wasserstein metric of order $p = 2$ together with the standard Euclidean norm, we define the ambiguity set as a ball of radius $\theta > 0$ centered at the given nominal distribution $\mathbb{Q}_t$:

$$\mathcal{D}_t := \{\mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_x}) \mid W_2(\mathbb{P}_t, \mathbb{Q}_t) \leq \theta\}.$$

In later sections, we show that employing the Wasserstein metric is useful in partially observable LQ control, as it contributes to obtaining a tractable solution and an out-of-sample performance guarantee, among others.

4.3 Tractable Approximation and Solution

The WDRC problem (4.4) is difficult to solve for two major reasons. First, the Bellman equation for (4.4) involves an infinite-dimensional minimax optimization problem. Second, partial observability aggravates the situation because the value (or cost-to-go) function is defined over the space of the information vectors. To resolve these issues, we propose a novel approximation technique and a simple solution to the approximate WDRC problem. Our method uses a Riccati equation and a tractable SDP problem.

4.3.1 Tractable Approximation

Our approximation technique has two main steps. We first introduce an additional penalty term in the cost function, motivated by our previous work for the fully observable case [42]. However, this approximation is insufficient when the system is partially observable. Thus, the second step is to further approximate the problem using the Gelbrich bound introduced in [41].

For the first step of the proposed approximation, instead of constraining the adversary player to select a disturbance distribution from the ambiguity set, we penalize the deviation of the distribution $\mathbb{P}_t$ from the nominal distribution $\mathbb{Q}_t$. Specifically, a

Wasserstein penalty term is added to the cost function as follows:

$$\tilde{J}_{\infty}^{\lambda}(\pi,\gamma) := \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}\left[\sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t] - \lambda W_2(\mathbb{P}_t, \mathbb{Q}_t)^2\right],$$

where $\lambda > 0$ is a user-specified penalty parameter designated for adjusting the conservativeness of the control policy. Then, the following minimax control problem approximates the original WDRC problem:

$$\min_{\pi\in\Pi} \max_{\gamma\in\Gamma} \tilde{J}_{\infty}^{\lambda}(\pi,\gamma), \tag{4.5}$$

where the set of admissible distribution policies is defined as $\Gamma := \{\gamma \mid \gamma_t(I_t) = \mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_x}),\ \gamma_t \text{ is measurable } \forall t\}$. This set is different from $\Gamma_{\mathcal{D}}$ in that it does not restrict the distribution $\mathbb{P}_t$ to be selected from the ambiguity set. This would give too much freedom to the adversary if there were no penalty terms. In general, the minimax control problem with the new cost function is intractable due to partial observability and the Wasserstein penalty term. In fully observable settings, when $\mathbb{Q}_t$ is chosen as an empirical distribution, the minimax problem attains a finite-dimensional formulation. However, problem (4.5) remains intractable due to partial observability, as demonstrated in Appendix 4.8.1.

The intractability of (4.5) motivates the need for another approximation step, where we propose employing the Gelbrich bound introduced in [41]. The Gelbrich bound is lower than the Wasserstein distance and is valid for any nominal distribution with finite first- and second-order moments. Let

$$\bar{w}_t := \mathbb{E}_{w_t\sim\mathbb{P}_t}[w_t], \quad \hat{w}_t := \mathbb{E}_{w_t\sim\mathbb{Q}_t}[w_t] \tag{4.6}$$

denote the mean vectors of w_t with respect to $\mathbb{P}_t$ and $\mathbb{Q}_t$, respectively. Also, we let

$$\begin{aligned} \Sigma_t &:= \mathbb{E}_{w_t\sim\mathbb{P}_t}[(w_t - \bar{w}_t)(w_t - \bar{w}_t)^\top], \\ \hat{\Sigma}_t &:= \mathbb{E}_{w_t\sim\mathbb{Q}_t}[(w_t - \hat{w}_t)(w_t - \hat{w}_t)^\top] \end{aligned} \tag{4.7}$$

denote the covariance matrices of w_t with respect to $\mathbb{P}_t$ and $\mathbb{Q}_t$, respectively. The Gelbrich bound for Wasserstein distance can be described as follows.

Lemma 4.1. *Suppose the mean vectors and covariance matrices of $\mathbb{P}_t$ and $\mathbb{Q}_t$ are given by* (4.6) *and* (4.7)*, respectively. Then, the following lower-bound holds for the 2-Wasserstein distance:*

$$\mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t) := \sqrt{\|\bar{w}_t - \hat{w}_t\|_2^2 + \mathrm{B}^2(\Sigma_t, \hat{\Sigma}_t)} \leq W_2(\mathbb{P}_t, \mathbb{Q}_t), \tag{4.8}$$

where

$$\mathrm{B}^2(\Sigma_t, \hat{\Sigma}_t) := \mathrm{Tr}[\Sigma_t + \hat{\Sigma}_t - 2(\hat{\Sigma}_t^{1/2}\Sigma_t\hat{\Sigma}_t^{1/2})^{1/2}].$$

Furthermore, the inequality holds with equality if $\mathbb{P}_t$ and $\mathbb{Q}_t$ are elliptical with the same density-generating function.

The Gelbrich bound relies only on the mean and covariance information, which is a crucial feature for obtaining a tractable solution.

Remark 4.1. *The Gelbrich bound provides a generic lower-bound for the Wasserstein distance for distributions that are not necessarily elliptical. Thus, it is applicable to problems with non-Gaussian disturbance distributions. The bound discards information about the nominal distribution $\mathbb{Q}_t$ beyond its first- and second-order moments, thereby sacrificing possibly useful information. However, it trades available information for tractability, providing a simple strategy for evaluating the closeness of two distributions. In Sections 4.4 and 4.5, we also show that the resulting controller enjoys various useful theoretical properties despite the limited use of available information.*[10]

We leverage the Gelbrich bound and define the following cost function, replacing the Wasserstein penalty term with its lower-bound:

$$J_\infty^\lambda(\pi, \gamma) = \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}\left[\sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t] - \lambda \mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t)^2\right].$$

Using this cost function, the penalty version (4.5) of the WDRC problem can be approximated as follows:

$$\min_{\pi\in\Pi}\max_{\gamma\in\Gamma} J_\infty^\lambda(\pi, \gamma). \tag{4.9}$$

[10]The empirical performance of a Gelbrich bound-based approximation has been demonstrated through motion control problems in [75].

Having the approximate problem (4.9), a closed-form expression of its optimal solution is derived using a Riccati equation in the following subsections. We first consider the case of finite-horizon problems and then extend the obtained results to the infinite-horizon average cost setting.

4.3.2 Finite-Horizon Problem

We begin our analysis by first considering the following finite-horizon approximate WDRC problem:

$$\min_{\pi\in\Pi}\max_{\gamma\in\Gamma} J_T^{\lambda}(\pi,\gamma), \tag{4.10}$$

where the cost function is defined as

$$J_T^{\lambda}(\pi,\gamma) = \mathbb{E}_{\mathbf{y}}\Bigg[\mathbb{E}_{x_T}[x_T^\top Q_f x_T \mid I_T] + \sum_{t=0}^{T-1}\Big(\mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t] - \lambda \mathrm{G}(\mathbb{P}_t,\mathbb{Q}_t)^2\Big)\Bigg].$$

To solve the minimax problem (4.10), we apply the dynamic programming (DP) algorithm by first defining the optimal value function recursively as follows.

Let

$$V_T(I_T) := \mathbb{E}_{x_T}[x_T^\top Q_f x_T \mid I_T]$$

and

$$\begin{aligned} V_t(I_t) :=& \inf_{u_t\in\mathbb{R}^{n_u}} \sup_{\mathbb{P}_t\in\mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_{x_t,y_{t+1}}\Big[x_t^\top Q x_t + u_t^\top R u_t - \lambda \mathrm{G}(\mathbb{P}_t,\mathbb{Q}_t)^2 \\ &+ V_{t+1}(I_t, y_{t+1}, u_t) \mid I_t, u_t\Big] \qquad (4.11)\\ =& \inf_{u_t\in\mathbb{R}^{n_u}} \sup_{\substack{\bar{w}_t\in\mathbb{R}^{n_x},\\ \Sigma_t\in\mathbb{S}_+^{n_x}}} \mathbb{E}_{x_t,y_{t+1}}[x_t^\top Q x_t + u_t^\top R u_t \\ &- \lambda[\|\bar{w}_t - \hat{w}_t\|^2 + \mathrm{B}^2(\Sigma_t,\hat{\Sigma}_t)] + V_{t+1}(I_t, y_{t+1}, u_t) \mid I_t, u_t] \qquad (4.12)\end{aligned}$$

for $t = T-1,\ldots,0$. Suppose for a moment that the outer minimization problem has an optimal solution u_t^* and the value function is measurable for every t. Then, by the

DP principle (e.g., [185–188]), we have

$$\inf_{\pi\in\Pi}\sup_{\gamma\in\Gamma} J_T^{\lambda}(\pi,\gamma) = \mathbb{E}_{y_0}[V_0(I_0)],$$

and an optimal control policy π_t^* can be constructed using the optimal solutions of the outer optimization problems for all t. To this end, we inductively show that the outer minimization problem in the Bellman equation (4.12) admits an optimal solution.

Let the expected value of the state x_t conditioned on the information vector I_t under the disturbance distribution generated by the adversary's policy γ be denoted by

$$\bar{x}_t := \mathbb{E}_{x_t}[x_t \mid I_t].$$

Also, let

$$\xi_t := x_t - \bar{x}_t$$

denote the deviation of the system state from its conditional expectation, and let

$$\Phi := BR^{-1}B^\top - \frac{1}{\lambda} I \in \mathbb{S}^{n_x}.$$

As the first step for our inductive argument, we identify an optimal solution to the outer minimization problem in (4.12) for time t when V_{t+1} has the following quadratic form.

Lemma 4.2. *Fix $t \in \{0, 1, \ldots, T-1\}$, and suppose that*

$$V_{t+1}(I_{t+1}) = \mathbb{E}_{x_{t+1}}[x_{t+1}^\top P_{t+1} x_{t+1} + \xi_{t+1}^\top S_{t+1}\xi_{t+1} + 2r_{t+1}^\top x_{t+1} \mid I_{t+1}] + q_{t+1},$$

for some $P_{t+1} \in \mathbb{S}_+^{n_x}$, $S_{t+1} \in \mathbb{S}_+^{n_x}$, $r_{t+1} \in \mathbb{R}^{n_x}$, and $q_{t+1} \in \mathbb{R}$. Moreover, assume that the penalty parameter satisfies $\lambda I \succ P_{t+1}$. Then, the following results hold:

- *The outer minimization problem in* (4.12) *with respect to u_t has the following unique optimal solution:*

$$u_t^* = K_t \bar{x}_t + L_t, \tag{4.13}$$

where

$$K_t = -R^{-1}B^\top (I + P_{t+1}\Phi)^{-1} P_{t+1} A \tag{4.14}$$

$$L_t = -R^{-1}B^\top (I + P_{t+1}\Phi)^{-1} (P_{t+1}\hat{w}_t + r_{t+1}). \tag{4.15}$$

- *Given u_t^*, the inner maximization problem in* (4.12) *with respect to w_t has the following unique optimal solution:*

$$\bar{w}_t^* = H_t \bar{x}_t + G_t, \tag{4.16}$$

where

$$H_t = (\lambda I - P_{t+1})^{-1} P_{t+1}(A + BK_t) \tag{4.17}$$

$$G_t = (\lambda I - P_{t+1})^{-1} \big(P_{t+1} B L_t + r_{t+1} + \lambda \hat{w}_t\big). \tag{4.18}$$

- *The inner maximization problem in* (4.12) *with respect to $\Sigma_t \in \mathbb{S}_+^{n_x}$ reduces to the following maximization problem:*

$$\begin{aligned} \max_{\Sigma_t \in \mathbb{S}_+^{n_x}} \; & \mathbb{E}_{x_{t+1}, y_{t+1}}[\xi_{t+1}^\top S_{t+1} \xi_{t+1} \mid I_t] \\ & + \operatorname{Tr}[(P_{t+1} - \lambda I)\Sigma_t + 2\lambda (\hat{\Sigma}_t^{1/2} \Sigma_t \hat{\Sigma}_t^{1/2})^{1/2}]. \end{aligned} \tag{4.19}$$

The proof of this lemma can be found in Appendix 4.8.2. Using this lemma, we can also show that V_t has the same form as V_{t+1} whenever $\lambda I \succ P_{t+1}$. To preserve the structure of the value function through the Bellman recursion, we impose the following assumption on the penalty parameter, which is also required for the fully observable case [42].

Assumption 4.1. *The penalty parameter satisfies $\lambda I \succ P_t$ for all $t = 1, \ldots, T$.*

Under this assumption, we can use mathematical induction backward in time to recursively show that the value functions V_t's have a specific quadratic form for all t because $V_T = \mathbb{E}_{x_T}[x_T^\top Q_f x_T \mid I_T]$ is already in that form. Consequently, it follows from the DP principle that the optimal control policy can be constructed as follows.

Theorem 4.1. *Suppose that Assumption 4.1 holds and* (4.19) *attains an optimal solution. Then, the value function for all* $t = 0, \ldots, T$ *has the following form:*

$$V_t(I_t) = \mathbb{E}_{x_t}[x_t^\top P_t x_t + \xi_t^\top S_t \xi_t + 2r_t^\top x_t \mid I_t] + q_t + \sum_{s=t}^{T-1} z_t(I_t, s).$$

Here, the coefficients $P_t \in \mathbb{S}_+^{n_x}, S_t \in \mathbb{S}_+^{n_x}, r_t \in \mathbb{R}^{n_x}$, *and* $q_t \in \mathbb{R}$ *are found recursively using the following Riccati equation:*

$$P_t = Q + A^\top (I + P_{t+1}\Phi)^{-1} P_{t+1} A \tag{4.20}$$

$$S_t = Q + A^\top P_{t+1} A - P_t \tag{4.21}$$

$$r_t = A^\top (I + P_{t+1}\Phi)^{-1}(r_{t+1} + P_{t+1}\hat{w}_t) \tag{4.22}$$

$$\begin{aligned} q_t = {} & q_{t+1} + (2\hat{w}_t - \Phi r_{t+1})^\top (I + P_{t+1}\Phi)^{-1} r_{t+1} \\ & + \hat{w}_t^\top (I + P_{t+1}\Phi)^{-1} P_{t+1} \hat{w}_t - \lambda \mathrm{Tr}[\hat{\Sigma}_t] \end{aligned} \tag{4.23}$$

with the terminal conditions $P_T = Q_f, S_T = 0, r_T = 0$, *and* $q_T = 0$. *The term* $z_t(I_t, s)$ *for* $s = t, \ldots, T-1$ *is given by*

$$\begin{aligned} z_t(I_t, s) := \sup_{\Sigma_s \in \mathbb{S}_+^{n_x}} & \mathbb{E}_{x_{s+1}, y_{t+1}, \ldots, y_{s+1}}[\xi_{s+1}^\top S_{s+1} \xi_{s+1} \mid I_t] \\ & + \mathrm{Tr}[(P_{s+1} - \lambda I)\Sigma_s + 2\lambda(\hat{\Sigma}_s^{1/2} \Sigma_s \hat{\Sigma}_s^{1/2})^{1/2}]. \end{aligned} \tag{4.24}$$

Moreover, an optimal policy pair can be obtained as follows:

- *The optimal control policy is uniquely given by*

  $$\pi_t^*(I_t) = K_t \bar{x}_t + L_t,$$

  with K_t *and* L_t *defined as* (4.14) *and* (4.15)*, respectively; and*

- *For each* I_t, *let* $\gamma_t^*(I_t) = \mathbb{P}_t^*$, *where* $\mathbb{P}_t^*$ *is a probability distribution with mean vector defined as* (4.16) *and covariance matrix* Σ_t^* *obtained as the maximizer of* (4.24) *for stage* t. *Then,* γ_t^* *is an optimal policy for the adversary that generates the worst-case distribution.*

The proof of this theorem can be found in Appendix 4.8.2. In the theorem, the existence of Σ_t^* is not guaranteed in general. However, we will see that Σ_t^* exists and is obtained in a tractable way if the Kalman filter is used.

It is worth comparing our result with that of the fully observable case [42]. Due to partial observability, the optimal control policy and the mean vector of the worst-case distribution are affine in the conditional expectation $\bar{x}_t$ instead of the actual state x_t. An additional estimator, such as the Kalman filter, is required for computing the state estimates based on the information I_t collected so far. However, the Riccati recursion (4.20)–(4.23), as well as the controller parameters (4.14) and (4.15), are independent of the information vector I_t. Thus, the *separation principle* holds for our WDRC method, where the state estimation and the optimal control parts can be decoupled, allowing each component to be designed independently.

The standard Kalman filter uses the mean vector and covariance matrix of the ground-truth disturbance distribution. However, in our problem setting, it is required to estimate the states under disturbances drawn from the worst-case distribution $\mathbb{P}_t^*$. The expected value of x_{t+1} conditioned on I_t is then estimated as follows:

$$\bar{x}_{t+1} = \bar{x}_{t+1}^- + \bar{X}_{t+1} C^\top M^{-1} (y_{t+1} - C\bar{x}_{t+1}^-), \tag{4.25}$$

where $\bar{x}_{t+1}^- = A\bar{x}_t + Bu_t^* + \bar{w}_t^*$ with $\bar{x}_t^- = m_0$. Here, $\bar{X}_t$ is the covariance matrix of x_t given I_t, i.e.,

$$\bar{X}_t = \mathbb{E}_{x_t}[(x_t - \bar{x}_t)(x_t - \bar{x}_t)^\top \mid I_t],$$

which can be precomputed by applying the following recursion forward in time:

$$\bar{X}_{t+1} = \bar{X}_{t+1}^- - \bar{X}_{t+1}^- C^\top (C\bar{X}_{t+1}^- C^\top + M)^{-1} C\bar{X}_{t+1}^- \tag{4.26}$$

$$\bar{X}_{t+1}^- = A\bar{X}_t A^\top + \Sigma_t^*, \tag{4.27}$$

starting from $\bar{X}_0^- = M_0$.

It follows from Theorem 4.1 and Kalman filter equations (4.25)–(4.27) that the optimal cost $J_T^\lambda(\pi^*, \gamma^*)$ depends on the worst-case distribution $\mathbb{P}_t^* = \gamma_t^*(I_t)$ only through

its first- and second-order moments. Therefore, *any* distribution with mean vector $\bar{w}_t^*$ and covariance matrix Σ_t^* is the worst-case distribution in (4.10). If the worst-case distribution is chosen to be Gaussian, then the Kalman filter is an optimal state estimator, as it minimizes the expected mean-squared error of state estimation [189]. As stated previously, when the Kalman filter is used for state estimation, the optimization problem (4.24) attains an optimal solution and can be recast as a tractable SDP problem.

Proposition 4.1. *Suppose that the system state at time* t *is estimated using the Kalman filter given the information vector* I_t*. Then,* $z_t(I_t, t)$ *given in* (4.24) *corresponds to the optimal value of the following tractable SDP problem:*

$$\begin{aligned}
\max_{\substack{X, X^-, \\ Y, \Sigma \in \mathbb{S}_+^{n_x}}} \quad & \mathrm{Tr}[S_{t+1}X + (P_{t+1} - \lambda I)\Sigma + 2\lambda Y] \\
\textit{s.t.} \quad & \begin{bmatrix} \hat{\Sigma}_t^{1/2} \Sigma \hat{\Sigma}_t^{1/2} & Y \\ Y & I \end{bmatrix} \succeq 0 \\
& \begin{bmatrix} X^- - X & X^- C^\top \\ CX^- & CX^- C^\top + M \end{bmatrix} \succeq 0 \\
& CX^- C^\top + M \succeq 0 \\
& X^- = A\bar{X}_t A^\top + \Sigma,
\end{aligned} \tag{4.28}$$

where $\bar{X}_t$ *is the covariance matrix of* x_t *conditioned on* I_t*.*

Moreover, an optimal solution Σ^* *to the SDP problem* (4.28) *is the covariance matrix of the worst-case distribution* $\mathbb{P}_t^*$ *in Theorem 4.1.*

The proof of this proposition can be found in Appendix 4.8.2. Notably, the reformulated SDP problem (4.28) is independent of real-time data such as the measurement y_t and the control input u_t. Therefore, the covariance matrix Σ_t^* of the worst-case distribution in each time stage can be computed offline by solving the SDP problem (4.28) using existing algorithms [114, 117, 119]. Having the covariance matrix Σ_t^*, the conditional state covariance matrix $\bar{X}_t$ can also be calculated offline by applying the Kalman filter recursion (4.26) and (4.27). Finally, in order to compute the value function at time

t, it is sufficient to have $z_s(I_s, s)$ for $s = t, \ldots, T-1$ as from the law of total expectation, it follows that $z_t(I_t, s) = z_s(I_s, s), s = t \ldots, T-1$.

4.3.3 From Finite-Horizon to Infinite-Horizon Problems

The results obtained for the finite-horizon problem can be extended to the infinite-horizon average cost setting (4.9) as letting T tend to ∞. Throughout this subsection, we assume the following:

Assumption 4.2. *The nominal distribution $\mathbb{Q}_t$ has a stationary mean vector and a stationary covariance matrix, i.e., $\hat{w}_t \equiv \hat{w}$ and $\hat{\Sigma}_t \equiv \hat{\Sigma}$ for all $t = 0, 1, \ldots$.*

Assumption 4.3. *$\Phi \succeq 0$, and $(A, \Phi^{1/2})$ is stabilizable and $(A, Q^{1/2})$ is observable.*

To examine the asymptotic behavior of the recursion (4.20)–(4.23), we first show the convergence of the Riccati equation (4.20) to a steady-state solution P_{ss} of an ARE.

Proposition 4.2. *Suppose that Assumptions 4.1–4.3 hold. Then, there exists a matrix $P_{ss} \in \mathbb{S}_+^{n_x}$ such that for every $P_T \in \mathbb{S}_+^{n_x}$, we have*

$$\lim_{T \to \infty} P_t = P_{ss}. \tag{4.29}$$

Furthermore, P_{ss} is the unique symmetric PSD solution of the following ARE:

$$P_{ss} = Q + A^\top (I + P_{ss}\Phi)^{-1} P_{ss} A. \tag{4.30}$$

The proof of this proposition can be found in Appendix 4.8.2. As a direct consequence, we can show the convergence of S_t and r_t to their corresponding limits.

Lemma 4.3. *Suppose that Assumptions 4.1–4.3 hold. Then, the matrix S_t and the vector r_t computed recursively according to* (4.21) *and* (4.22) *starting from $S_T = 0$ and $r_T = 0$ converge to*

$$S_{ss} = Q + A^\top P_{ss} A - P_{ss}, \tag{4.31}$$

$$r_{ss} = [I - A^\top (I + P_{ss}\Phi)^{-1}]^{-1} A^\top (I + P_{ss}\Phi)^{-1} P_{ss} \hat{w} \tag{4.32}$$

as $T \to \infty$, respectively.

The proof of this lemma can be found in Appendix 4.8.2. Proposition 4.2 and Lemma 4.3 yield to identify the limiting behavior of the finite-horizon optimal policy as the horizon length tends to infinity.

Theorem 4.2. *Suppose that Assumptions 4.1–4.3 hold. Then, as $T \to \infty$, the optimal control policy $\pi_t^*(I_t)$ converges pointwise to the steady-state policy*

$$\pi_{ss}^*(I_t) := K_{ss}\bar{x}_t + L_{ss}, \tag{4.33}$$

where

$$K_{ss} = -R^{-1}B^\top(I + P_{ss}\Phi)^{-1}P_{ss}A, \tag{4.34}$$

$$L_{ss} = -R^{-1}B^\top(I + P_{ss}\Phi)^{-1}(P_{ss}\hat{w} + r_{ss}). \tag{4.35}$$

Furthermore, as $T \to \infty$, the mean vector of the worst-case distribution $\mathbb{P}_t^*$ generated by the adversary converges to

$$\bar{w}_{t,ss}^* = H_{ss}\bar{x}_t + G_{ss}, \tag{4.36}$$

where

$$H_{ss} = (\lambda I - P_{ss})^{-1}P_{ss}(A + BK_{ss}), \tag{4.37}$$

$$G_{ss} = (\lambda I - P_{ss})^{-1}\big(P_{ss}BL_{ss} + r_{ss} + \lambda\hat{w}\big). \tag{4.38}$$

The convergence of K_t, L_t, H_t, and G_t in (4.14)–(4.18) directly follows from the convergence of P_t and r_t. The steady-state control policy (4.33) is again affine in the conditional expectation of the system state. However, it is now stationary, making the controller more attractive for practical implementation.

Theorem 4.2 only concerns the mean vector of the worst-case distribution, which is insufficient to analyze the steady-state behavior of the policy γ_t^* of the adversary. Therefore, in the remainder of this subsection, we consider a worst-case distribution policy of a special form and show that it is, in fact, optimal to the infinite-horizon average cost problem (4.9). To this end, consider a stationary distribution policy γ_{ss}^*

that maps the information vector to a probability distribution with the mean vector $\bar{w}^*_{t,ss}$ defined as (4.36) and the stationary covariance matrix Σ^*_{ss} defined as an optimal solution to the following maximization problem:

$$\begin{aligned} \max_{\substack{X, X^-, \\ \Sigma \in \mathbb{S}^{n_x}_+}} \; & \mathrm{Tr}[S_{ss} X + (P_{ss} - \lambda I)\Sigma + 2\lambda(\hat{\Sigma}^{1/2} \Sigma \hat{\Sigma}^{1/2})^{1/2}] \\ \text{s.t. } & X^- = A X A^\top + \Sigma \\ & X = X^- - X^- C^\top (C X^- C^\top + M)^{-1} C X^-. \end{aligned} \tag{4.39}$$

For further analysis, we impose the following assumption:

Assumption 4.4. *(A, C) is detectable and $\left(A, (\Sigma^*_{ss})^{1/2}\right)$ is stabilizable.*

It is well known from filtering theory (e.g., [189]) that under the distribution policy γ^*_{ss} satisfying Assumption 4.4, the matrix $\bar{X}^-_t$ given by the recursion in (4.27) tends to a PSD matrix $\bar{X}^-_{ss}$ that solves the following filter ARE:

$$\bar{X}^-_{ss} = A(\bar{X}^-_{ss} - \bar{X}^-_{ss} C^\top (C \bar{X}^-_{ss} C^\top + M)^{-1} C \bar{X}^-_{ss}) A^\top + \Sigma^*_{ss} \tag{4.40}$$

for any initial state covariance matrix $M_0 \in \mathbb{S}^{n_x}_+$. Consequently, the covariance matrix $\bar{X}_t$ converges to the constant PSD matrix

$$\bar{X}_{ss} = \bar{X}^-_{ss} - \bar{X}^-_{ss} C^\top (C \bar{X}^-_{ss} C^\top + M)^{-1} C \bar{X}^-_{ss}, \tag{4.41}$$

with the state recursively estimated according to the following asymptotic form:

$$\bar{x}_{t+1} = \bar{x}^-_{t+1} + \bar{X}_{ss} C^\top M^{-1} (y_{t+1} - C \bar{x}^-_{t+1}), \tag{4.42}$$

where $\bar{x}^-_{t+1} = A\bar{x}_t + Bu_t + \bar{w}^*_{t,ss}$ with $\bar{x}_{0|-1} = m_0$. This property is known as the duality between estimation and control. As a result, the asymptotic performance of the filter is similar to that of the standard Riccati equation, yielding the steady-state counterpart of the Kalman filter.

Due to its constraints, the optimization problem (4.39) is intractable. Using a similar argument to Proposition 4.1, it can be reformulated as the following tractable SDP

problem:

$$\begin{aligned}\max_{X,X^-,Y,\Sigma\in\mathbb{S}_+^{n_x}} \ & \mathrm{Tr}[S_{ss}X + (P_{ss}-\lambda I)\Sigma_{ss} + 2\lambda Y] \\ \text{s.t.} \ & \begin{bmatrix} \hat{\Sigma}^{1/2}\Sigma\hat{\Sigma}^{1/2} & Y \\ Y & I \end{bmatrix} \succeq 0 \\ & \begin{bmatrix} X^- - X & X^-C^\top \\ CX^- & CX^-C^\top + M \end{bmatrix} \succeq 0 \\ & CX^-C^\top + M \succeq 0 \\ & X^- = AXA^\top + \Sigma,\end{aligned} \tag{4.43}$$

which is independent of the information vector I_t and can be solved offline.

Finally, we can build the connection between the policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$ and the solution to the infinite-horizon minimax problem (4.9). For that, let the steady-state average cost incurred by the stationary policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$ be denoted as

$$\rho := J_\infty^\lambda(\pi_{ss}^*, \gamma_{ss}^*),$$

which can be calculated by combining the results from Theorem 4.1 and the maximization problem (4.39) as follows.

Proposition 4.3. *Suppose that Assumptions 4.1–4.4 hold. Then, the steady-state average cost is given by*

$$\rho = (2\hat{w} - \Phi r_{ss})^\top (I + P_{ss}\Phi)^{-1} r_{ss} - \lambda \mathrm{Tr}[\hat{\Sigma}] + \hat{w}^\top (I + P_{ss}\Phi)^{-1} P_{ss}\hat{w} + z_{ss}, \tag{4.44}$$

where z_{ss} is the optimal value of the maximization problem (4.39)*.*

The proof of this proposition can be found in Appendix 4.8.2. Having the steady-state average cost, it remains to verify the optimality of the policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$ in the average-cost criterion. For that purpose, we introduce the following optimality condition:

Proposition 4.4. *Suppose that Assumptions 4.1–4.4 hold. Then, the following average-cost optimality equation holds:*

$$\rho + h(I_t) = \inf_{u_t \in \mathbb{R}^{n_u}} \sup_{\mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_{x_t, y_{t+1}} \Big[x_t^\top Q x_t + u_t^\top R u_t - \lambda \mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t)^2 + h(I_{t+1}) \mid I_t, u_t \Big], \tag{4.45}$$

where ρ is the steady-state average cost defined as (4.44) *and*

$$h(I_t) = \bar{x}_t^\top P_{ss} \bar{x}_t + 2 r_{ss}^\top \bar{x}_t + \mathrm{Tr}[(S_{ss} + P_{ss}) \bar{X}_{ss}].$$

In addition, $(\pi_{ss}^*(I_t), \gamma_{ss}^*(I_t))$ is an optimal solution pair to the minimax problem on the right-hand side of (4.45)*.*

The proof of this proposition can be found in Appendix 4.8.2. Here, h is called the *bias* and represents the transient cost. Using the bias term, we now consider the following extended average-cost function:

$$\bar{J}_\infty^\lambda(\pi, \gamma) := \limsup_{T \to \infty} \frac{1}{T} \bar{J}_T^\lambda(\pi, \gamma), \tag{4.46}$$

where

$$\bar{J}_T^\lambda(\pi, \gamma) = \mathbb{E}_{\mathbf{y}} \left[h(I_T) + \sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t] - \lambda \mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t)^2 \right].$$

The extended average cost (4.46) allows us to investigate the optimality of the steady-state policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$.

Proposition 4.5. *Suppose that Assumptions 4.1–4.4 hold. Then, the steady-state policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$ is optimal to*

$$\min_{\pi \in \bar{\Pi}} \max_{\gamma \in \bar{\Gamma}} J_\infty^\lambda(\pi, \gamma)$$

for any policy spaces $\bar{\Pi} \subset \Pi$ and $\bar{\Gamma} \subset \Gamma$ satisfying

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E}_{\mathbf{y}}[h(I_T) \mid \pi, \gamma_{ss}^*] = 0, \ \forall \pi \in \bar{\Pi} \tag{4.47}$$

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E}_{\mathbf{y}}[h(I_T) \mid \pi_{ss}^*, \gamma] = 0, \ \forall \gamma \in \bar{\Gamma}. \tag{4.48}$$

Moreover, the optimal value of this problem is equal to ρ.

The proof of this proposition can be found in Appendix 4.8.2. The first condition is similar to the one in the standard LQG control, with the difference that the disturbances follow the worst-case distribution policy γ^*_{ss}. If the expected value of the state with respect to all uncertainties is bounded under the policy pair (π^*_{ss}, γ) for some $\gamma \in \bar{\Gamma}$, then condition (4.48) holds. In fact, it is satisfied as long as the distribution $\mathbb{P}_t = \gamma(I_t)$ has a bounded mean vector and a stationary covariance matrix so that the pair $(A, \Sigma^{1/2})$ is stabilizable. This is due to the stability properties of the optimal control policy π^*_{ss}, which is discussed in Section 4.5.

We wrap up this subsection observing the tightness of the proposed Gelbrich bound-based approximation when the nominal distribution $\mathbb{Q}_t$ is elliptical. This is because the worst-case distribution can be chosen to be elliptical with the worst-case mean vector and covariance matrix.

Proposition 4.6. *Suppose that the nominal distribution $\mathbb{Q}_t$ is elliptical for all t. Let (π^*, γ^*) denote an optimal policy pair of the approximate minimax control problem* (4.9)*, such that the worst-case distribution $\mathbb{P}^*_t = \gamma^*_t(I_t)$ is elliptical with the same density generating function as $\mathbb{Q}_t$. Then, (π^*, γ^*) is an optimal policy pair for the minimax control problem* (4.5)*.*

The proof of this proposition can be found in Appendix 4.8.2. This property once again confirms the validity of our approximation scheme, as most LQ optimal control problems use nominal distributions as Gaussian. For general distributions, the proposed approximate controller is further shown to have performance guarantees in Section 4.4.

4.3.4 Algorithm

The results presented in previous sections lead us to a novel infinite-horizon WDRC scheme that controls the partially observable system (4.1) while continuously updating the state estimates. The block diagram of our method is depicted in Fig. 4.1, while

Algorithm 4: Infinite-horizon WDRC algorithm

1 **Input:** $\lambda, \hat{w}, \hat{\Sigma}, m_0, M$

2 Solve ARE (4.30) to obtain P_{ss}

3 Calculate K_{ss} and L_{ss} by (4.34) and (4.35)

4 Compute parameters H_{ss} and G_{ss} according to (4.37) and (4.38)

5 Solve SDP problem (4.43) to obtain Σ^*_{ss}

6 Solve filter ARE (4.40) and use (4.41) to obtain $\bar{X}_{ss}$

7 Measure y_0 and estimate $\bar{x}_0$ via (4.42)

8 **for** $t = 0, 1, \ldots$ **do**

9 Apply $u^*_t = \pi^*_{ss}(I_t) = K_{ss}\bar{x}_t + L_{ss}$ to the system (4.1)

10 Compute the worst-case mean $\bar{w}^*_{t,ss}$ according to (4.36)

11 Measure y_{t+1} and estimate $\bar{x}_{t+1}$ via (4.42)

the detailed procedure is given in Algorithm 4. The penalty parameter λ is initially given to the algorithm, chosen depending on the desired level of conservativeness and satisfying Assumption 4.1. The remaining inputs of the algorithm include the mean vector $\hat{w}$ and the covariance matrix $\hat{\Sigma}$ of the nominal distribution $\mathbb{Q}_t$, the initial state mean vector m_0, and the covariance matrix of the output noise M. Our algorithm essentially comprises two stages: offline and online, where the first stage concerns the controller and estimator design, while the second stage is for real-time deployment of the controller.

Since the separation principle applies to our method, we disentangle the controller from the state estimator. Therefore, in the first part, a stationary optimal control policy is synthesized (Lines 2 and 3), followed by the worst-case distribution policy construction (Lines 4 and 5). More specifically, in Line 2, the ARE (4.30) is solved to obtain the matrix P_{ss}, which is used in Line 3 to calculate K_{ss} and L_{ss} according to (4.34) and (4.35), respectively. Next, in Line 4, the parameters H_{ss} and G_{ss} of the mean vector of the worst-case disturbance distribution are found according to (4.37) and (4.38),

respectively. In Line 5, the SDP problem (4.43) is solved numerically using the steady-state matrices P_{ss} and S_{ss}. Next, in Line 6, we solve the filter ARE (4.40) and (4.41) to obtain the conditional state covariance matrix $\bar{X}_{ss}$ under the worst-case distribution.

The online stage for the fixed controller and estimator is presented in Lines 7–11, where the optimal policy π^*_{ss} is deployed to control the actual partially observable system. In the beginning, an initial measurement y_0 is received, and the initial state estimate $\bar{x}_0$ is obtained by the Kalman filter (Line 7). Then, in each time stage, a control input u^*_t is applied to the system leveraging the optimal policy π^*_{ss} and the current state estimate $\bar{x}_t$ (Line 9). The mean vector $w^*_{t,ss}$ of the worst-case distribution is then computed according to (4.36) using the parameters H_{ss} and G_{ss} calculated in the offline stage. Finally, in Line 11, the new measurements y_{t+1} are used to update the estimate about the state x_{t+1}.

It is worth mentioning that the infinite-horizon WDRC algorithm is applicable only to environments where the nominal distribution is stationary, in order to satisfy Assumption 4.2. However, in practice, the disturbance distribution is often non-stationary and varies over time. While the infinite-horizon formulation provides performance and stability guarantees, as explained in the following sections, the finite-horizon WDRC algorithm is more practical, especially for autonomous systems. In the finite-horizon formulation, the control policy is updated at each time step based on the current nominal disturbance information, allowing for adaptation to changing conditions.

4.4 Performance Guarantees

Though our approach yields a closed-form expression for the optimal control policy of the approximate minimax control problem (4.9), its relation to the original WDRC problem (4.4) is yet to be established. In this section, we demonstrate the capability of our method to provide distributional robustness with a guaranteed cost property and a probabilistic out-of-sample performance guarantee, which is an essential feature of

the WDRC method.

4.4.1 Guaranteed Cost Property

Fix a penalty parameter $\lambda > 0$ satisfying Assumption 4.1. The corresponding solution to ARE (4.30) will be P_{ss}. Now, consider the average cost criterion (4.3) and its extended version with the bias h being added as follows:

$$\bar{J}_\infty(\pi, \gamma) = \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}\left[h(I_T) + \sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t]\right].$$

The following theorem demonstrates the uniform bound on the average-cost criterion (4.3) under the stationary control policy computed in Theorem 4.1 for any worst-case distribution in the Wasserstein ambiguity set $\mathcal{D}$.

Theorem 4.3. *Suppose that Assumptions 4.1–4.4 hold for a fixed $\lambda > 0$. Also, let $\pi_{ss}^{\lambda,*}$ be the optimal control policy of the penalty problem* (4.9)*. For any policy space $\bar{\Gamma}_{\mathcal{D}}$ defined in Proposition 4.5, the average cost under the worst-case distribution policy in $\bar{\Gamma}_{\mathcal{D}}$ is bounded as follows:*

$$\sup_{\gamma\in\bar{\Gamma}_{\mathcal{D}}} J_\infty(\pi_{ss}^{\lambda,*}, \gamma) \leq \theta^2\lambda + \rho(\lambda). \tag{4.49}$$

The proof of this theorem can be found in Appendix 4.8.2. Theorem 4.3 demonstrates the distributional robustness of the optimal control policy $\pi_{ss}^{\lambda,*}$ to the approximate penalty problem, which can be controlled by tuning λ. The bound (4.49) suggests an intuitive approach for selecting the penalty parameter given a Wasserstein ball radius θ, as it is desirable to select a λ that minimizes the upper-bound,[11] i.e.,

$$\lambda(\theta) \in \arg\min_{\lambda>0} [\theta^2\lambda + \rho(\lambda)]. \tag{4.50}$$

This optimal penalty parameter is used in the following subsection.

[11] This approach was used to determine λ for our experiments in Section 4.6.

4.4.2 Out-of-Sample Performance Guarantee

Suppose that the standard stochastic optimal controller is constructed using an empirical disturbance distribution constructed from the training dataset $\mathbf{w} := \{\hat{w}^{(1)}, \ldots, \hat{w}^{(N)}\}$. The performance of this controller is deteriorated when evaluated under a testing dataset of w_t which is different from the training dataset. This issue arises even if the training and testing datasets are sampled from the same disturbance distribution. A substantial advantage of WDRC is to address this out-of-sample issue by providing a performance guarantee [30].

We argue that such an out-of-sample performance guarantee is achieved by the proposed method despite approximation. Specifically, we show that for a well-calibrated Wasserstein ambiguity set, our method with a nominal empirical distribution provides an upper confidence bound on the true average cost. Throughout this section, the nominal distribution is chosen as the following stationary empirical distribution $\mathbb{Q}$ constructed from a finite sample dataset $\mathbf{w}$:

$$\mathbb{Q} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\hat{w}^{(i)}}, \tag{4.51}$$

where δ_w denotes the Dirac measure concentrated at w. Here, each sample $\hat{w}^{(i)}$ is drawn from the true stationary distribution $\mathbb{P}$.

Given the optimal penalty parameter $\lambda(\theta)$ defined as (4.50), let $(\pi_{ss,\mathbf{w}}^{\lambda(\theta),*}, \gamma_{ss,\mathbf{w}}^{\lambda(\theta),*})$ denote the optimal stationary policy pair constructed in Section 4.3.3 with the sample dataset $\mathbf{w}$. Then, the out-of-sample performance (or cost) of $\pi_{ss,\mathbf{w}}^{\lambda(\theta),*}$ is defined as

$$J_\infty(\pi_{ss,\mathbf{w}}^{\lambda(\theta),*}, \gamma) = \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}_{\mathbf{y}} \left[\sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t \mid I_t, u_t] \,\middle|\, \pi_{ss,\mathbf{w}}^{\lambda(\theta),*}, \gamma \right],$$

where γ is a stationary policy mapping the information vector to the true disturbance distribution, i.e., $\gamma(I_t) = \mathbb{P}$ for all t.

However, as the true distribution $\mathbb{P}$ is unknown in practice, it is impossible to directly evaluate the out-of-sample performance. Instead, we consider the following al-

ternative probabilistic performance guarantee:

$$\mathbb{P}^N \left\{ \mathbf{w} \mid J_\infty(\pi_{ss,\mathbf{w}}^{\lambda(\theta),*}, \gamma) \leq \theta^2 \lambda(\theta) + \rho(\lambda(\theta)) \right\} \geq 1 - \beta, \tag{4.52}$$

where $\beta \in (0,1)$ represents a confidence level. Here, the dataset $\mathbf{w}$ is viewed as a random object governed by the distribution $\mathbb{P}^N$. The inequality (4.52) means that the cost incurred by the proposed policy under the true disturbance distribution is limited by $\theta^2\lambda(\theta)+\rho(\lambda(\theta))$ with probability no less than $1-\beta$. Note that the cost upper-bound $\theta^2\lambda(\theta)+\rho(\lambda(\theta))$ can be computed using the proposed method without the knowledge of the true distribution $\mathbb{P}$. The probability on the left-hand side critically depends on θ. Thus, given β, the size of the ambiguity set must be carefully determined to attain the probabilistic out-of-sample performance guarantee.

We identify the desired radius θ under the following assumption, ensuring that $\mathbb{P}$ is a light-tailed distribution:

Assumption 4.5. *Suppose there exist $c > 2$ and $B > 0$ such that*

$$\mathbb{E}_{w\sim\mathbb{P}}[\exp(\|w\|^c)] \leq B.$$

The required radius θ can then be found from the following measure concentration inequality for the Wasserstein metric [190, Theorem 2]:

$$\mathbb{P}^N\big\{\mathbf{w} \mid W_2(\mathbb{P},\mathbb{Q}) \geq \theta\big\} \leq c_1\big[b_1(N,\theta)\mathbf{1}_{\{\theta\leq 1\}} + b_2(N,\theta)\mathbf{1}_{\{\theta>1\}}\big], \tag{4.53}$$

where

$$b_1(N,\theta) := \begin{cases} \exp(-c_2 N\theta^2) & \text{if } n_x < 4 \\ \exp\big(-c_2 N\big(\frac{\theta}{\log(2+1/\theta)}\big)^2\big) & \text{if } n_x = 4 \\ \exp(-c_2 N\theta^{n_x/2}) & \text{otherwise} \end{cases}$$

and

$$b_2(N,\theta) := \exp(-c_2 N\theta^{c/2})$$

for some constants $c_1, c_2 > 0$, depending only on n_x and c. The measure concentration inequality (4.53) provides an upper-bound on the probability that the true disturbance

distribution $\mathbb{P}$ lies outside the Wasserstein ambiguity set. This inequality is essential for determining the radius θ required for ensuring the probabilistic out-of-sample performance of our control policy.

Theorem 4.4. *Suppose that Assumptions 4.1–4.5 hold. We also assume that the radius θ is chosen as*

$$\theta := \begin{cases} \left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{2/c} & \text{if } N < \frac{1}{c_2}\log(c_1/\beta) \\ \left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{1/2} & \text{if } N \geq \frac{1}{c_2}\log(c_1/\beta),\ n_x < 4 \\ \left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{2/n_x} & \text{if } N \geq \frac{1}{c_2}\log(c_1/\beta),\ n_x > 4 \\ \bar{\theta} & \text{if } N \geq \frac{(\log 3)^2}{c_2}\log(c_1/\beta),\ n_x = 4 \end{cases} \tag{4.54}$$

for $\bar{\theta}$ satisfying the condition

$$\frac{\bar{\theta}}{\log(2 + 1/\bar{\theta})} = \left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{1/2}.$$

Then, the probabilistic out-of-sample performance guarantee (4.52) *holds.*

The proof of this theorem can be found in Appendix 4.8.2.

Under an additional assumption that the disturbance distribution $\mathbb{P}$ is compactly supported, the concentration inequality suggested in [43, Proposition 3.2] can be used to further strengthen our result. Let the diameter of a set $S \in \mathbb{R}^{n_x}$ be denoted by $\mathrm{diam}(S) := \sup\{\|x - y\|_\infty \mid x, y \in S\}$, and for $\mathbb{P} \in \mathcal{P}(\mathbb{R}^{n_x})$ let $\mathrm{supp}(\mathbb{P})$ denote its support.

Corollary 4.1. *Suppose that Assumptions 4.1–4.4 hold and the true disturbance distribution $\mathbb{P}$ is compactly supported with $\xi := \frac{1}{2}\mathrm{diam}(\mathrm{supp}(\mathbb{P}))$. Suppose the radius θ is chosen as*

$$\theta := \begin{cases} \xi\left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{1/4} & \text{if } n_x < 4 \\ \xi\left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{1/n_x} & \text{if } n_x > 4 \\ \bar{\theta} & \text{if } n_x = 4 \end{cases}$$

for $\bar{\theta}$ satisfying the condition

$$\frac{\bar{\theta}^2}{\xi^2 \log(2 + \xi^2/\bar{\theta}^2)} = \left[\frac{\log(c_1/\beta)}{c_2 N}\right]^{1/2},$$

where $c_1, c_2 > 0$ are some constants depending only on n_x. Then, the probabilistic out-of-sample performance guarantee (4.52) *holds.*

4.5 Stability

This section investigates the stability properties of the closed-loop system when the proposed control policy π_{ss}^* is employed. It follows from Theorem 4.2 that the closed-loop system is expressed as

$$x_{t+1} = Ax_t + BK_{ss}\bar{x}_t + w_t + BL_{ss},$$

where $\bar{x}_t$ is the current state estimate. Assuming that the Kalman filter is chosen as the state estimator, our focus is to analyze the following mean-state system:

$$\begin{aligned}
\mathbb{E}[x_{t+1}] &= A\mathbb{E}[x_t] + BK_{ss}\mathbb{E}[\bar{x}_t] + \mathbb{E}[w_t] + BL_{ss} \\
\mathbb{E}[\bar{x}_{t+1}] &= \mathbb{E}[\bar{x}_{t+1}^-] + \bar{X}_{ss}C^\top M^{-1} C\mathbb{E}[x_{t+1} - \bar{x}_{t+1}^-] \\
\mathbb{E}[y_t] &= C\mathbb{E}[x_t] + \mathbb{E}[v_t],
\end{aligned} \tag{4.55}$$

where $\mathbb{E}[\bar{x}_{t+1}^-] = (A + BK_{ss} + H_{ss})\mathbb{E}[\bar{x}_t] + BL_{ss} + G_{ss}$. Here, the expectation is taken with respect to the joint probability distribution of all uncertainties up to time t.

Let

$$\tilde{x}_t := \mathbb{E}[x_t], \quad \bar{\tilde{x}}_t := \mathbb{E}[\bar{x}_t]$$

consist of the state of the mean-state system (4.55). We can show the stabilizing properties of the policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$ for the mean-state system when the nominal disturbance distribution $\mathbb{Q}_t$ has zero mean.

Proposition 4.7. *Suppose that Assumptions 4.1–4.4 hold. Under the policy pair $(\pi_{ss}^*, \gamma_{ss}^*)$, both $\tilde{x}_t$ and $\bar{\tilde{x}}_t$ of the mean-state system* (4.55) *converge to the following value:*

$$[I - (I + \Phi P_{ss})^{-1} A]^{-1} (I - \Phi(I + P_{ss}\Phi - A^\top)^{-1} P_{ss}]\hat{w}. \tag{4.56}$$

Moreover, if $\hat{w} = \mathbb{E}_{w_t \sim \mathbb{Q}_t}[w_t] = 0$*, the control policy* π^*_{ss} *stabilizes the system under the worst-case distribution policy* γ^*_{ss}*.*

The proof of this proposition can be found in Appendix 4.8.2. Furthermore, we can show that π^*_{ss} guarantees the BIBO stability of the closed-loop system (4.55) when viewing the disturbances as input.

Proposition 4.8. *Suppose that Assumptions 4.1–4.3 hold and the pair* (A, C) *is detectable. Then, the closed-loop gain matrix* $(A + BK_{ss})$ *is stable. Moreover, the mean-state system* (4.55) *under the control policy* π^*_{ss} *is BIBO stable when viewing the disturbances as input.*

The proof of this proposition can be found in Appendix 4.8.2. It follows from BIBO stability that as long as the mean vector of the disturbance distribution is bounded, the expected value of the closed-loop system state and the corresponding output will remain bounded.

4.6 Case Study

In this section, we demonstrate the performance of our WDRC method in both finite- and infinite-horizon settings and compare its performance with the standard LQG controller [4], which uses the estimated distribution of the disturbances. Since the true disturbance distribution is unknown, LQG directly uses the nominal distribution in both the controller and the estimator. For comparison, we test our algorithm in the presence of disturbances drawn from (i) a Gaussian distribution and (ii) uniform distribution. All algorithms were implemented in Python and run on a PC with an Intel Core i7-8700K (3.70 GHz) CPU and 32 GB RAM. The source code of our implementation is available online.[12]

[12] `https://github.com/CORE-SNU/PO-WDRC`

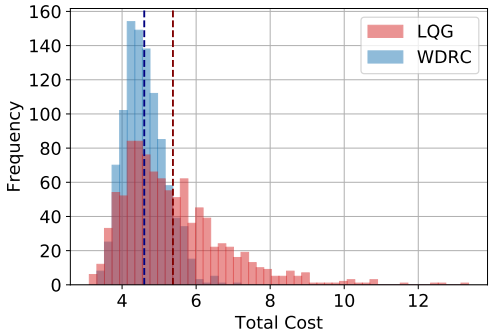


Figure 4.2: Histogram of the total costs in the case of Gaussian disturbances. The dashed lines represent the sample means of the costs returned by the two methods.

4.6.1 Finite-Horizon Settings

In these experiments, we consider a discrete-time system with the following parameters:

$$A = \begin{bmatrix} 0.518 & 0.266 \\ 0.405 & 0.806 \end{bmatrix}, B = \begin{bmatrix} -2.972 \\ -2.271 \end{bmatrix}, C = \begin{bmatrix} 1.023 & 1.955 \end{bmatrix},$$

which is unstable due to an eigenvalue outside the unit circle. The controller is required to minimize the cost with parameters $Q = Q_f = R = I$ over the time horizon of $T = 50$. The nominal disturbance distribution of w_t is estimated as a Gaussian with the empirical mean and covariance matrix constructed from $N = 5$ sample data. The states are estimated via the Kalman filter.[13]

Gaussian Case

In the first scenario, the true disturbance distribution is chosen as $\mathcal{N}([0.01, 0.02]^\top$, $[0.01,\ 0.005; 0.005,\ 0.01])$, and the disturbance data $\hat{w}_t^{(i)}$ are sampled from this distribution. The observation noise v_t follows zero-mean Gaussian distribution with covariance $M = 0.2I$, and the initial state is assumed to be distributed according to

[13] Since the actual disturbance distribution is unknown, the mean and covariance of the nominal distribution are used in the Kalman filter for the standard LQG.

Table 4.1: Total cost averaged over 1,000 simulations in the finite-horizon settings.

	Total Cost	
	WDRC	LQG
Gaussian	4.599 (0.557)	5.374 (1.398)
Uniform	0.536 (0.151)	0.781 (0.267)

$x_0 \sim \mathcal{N}([-1,-1]^\top, 0.001I)$. The penalty parameter was found according to (4.50) for $\theta = 0.1$ so that it satisfies Assumption 4.1.

Fig. 4.2 shows the distributions of the total costs over 1,000 simulations as a histogram. Overall, the cost distribution for the WDRC method has a bell shape, and thus is more favorable than that for the LQG controller. The WDR controller returns lower costs with a higher probability compared to the LQG controller. This is explained by the fact that the WDRC anticipates mismatches between the true disturbance distribution and the nominal one. Meanwhile, LQG is unable to deal with such unexpected distribution errors, causing higher total costs with a right-skewed distribution. In addition, the WDRC controller is less sensitive to the state estimates $\bar{x}_t$, unlike LQG, which relies solely on the inaccurate nominal distribution at both the control and estimation stages.

The total costs for both WDRC and LQG methods are reported in Table 4.1. The WDRC controller incurs a lower average total cost with a smaller standard deviation compared to the LQG method, confirming the superiority of our method.

Uniform Case

In the second scenario, the true disturbance distribution is assumed to be uniform, $\mathcal{U}[-0.05, 0.05]^2$, and the disturbance data $\hat{w}_t^{(i)}$ are sampled from this distribution. The observation noise v_t is drawn from a zero-mean Gaussian distribution with covariance $M = 0.1I$. The initial state is uniformly distributed with $x_0 \sim \mathcal{U}([0.1, 0.2]^\top,$

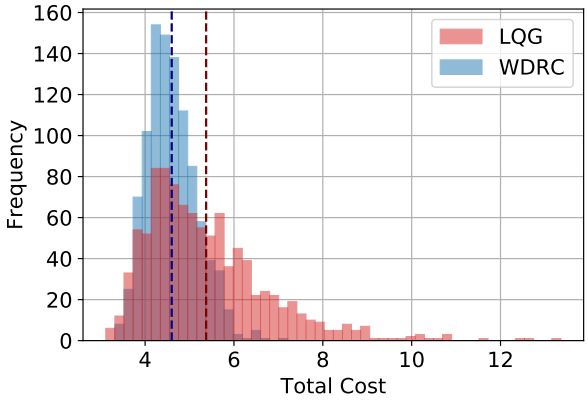


Figure 4.3: Histogram of the total costs in the case of uniform disturbances. The dashed lines represent the sample means of the costs returned by the two methods.

$[0.3, 0.5]^\top$). Since the state distribution is not Gaussian in this setting, the Kalman filter is no longer an optimal estimator. Yet, we apply the Kalman filter with the Gaussian nominal distribution of disturbances to demonstrate the capability of the WDRC to compensate for an inexact state estimator. The penalty parameter was tuned for $\theta = 0.03$ following the same procedure as in the Gaussian case.

Fig. 4.3 illustrates the cost histograms over 1,000 simulation runs. The effect of the penalty term is more pronounced here, as the difference between the cost distributions is larger compared to the Gaussian case. In particular, the total costs incurred by the WDRC method are concentrated in the low-cost regions, while those incurred by LQG are spread wider, with a right tail in the high-cost region.

Table 4.1 summarizes the total costs for both WDRC and LQG methods. Analogous to the Gaussian case, the average total cost incurred by the WDRC method is significantly lower than that obtained using LQG. Moreover, the standard deviation of the costs is considerably smaller when using the WDRC controller. The reason for this result is twofold. First, the nominal distribution is not an efficient estimator of the true uniform distribution; therefore, relying on moment estimates is insufficient. The WDRC approach alleviates this issue by considering all distributions close to the nom-

inal one, thereby enabling the system to effectively handle the distribution mismatch. Second, the state estimation for LQG is performed for Gaussian disturbances with a nominal mean and covariance, while the WDRC method uses the worst-case distribution in the state estimation, adding additional robustness to the estimation stage.

4.6.2 Infinite-Horizon Settings

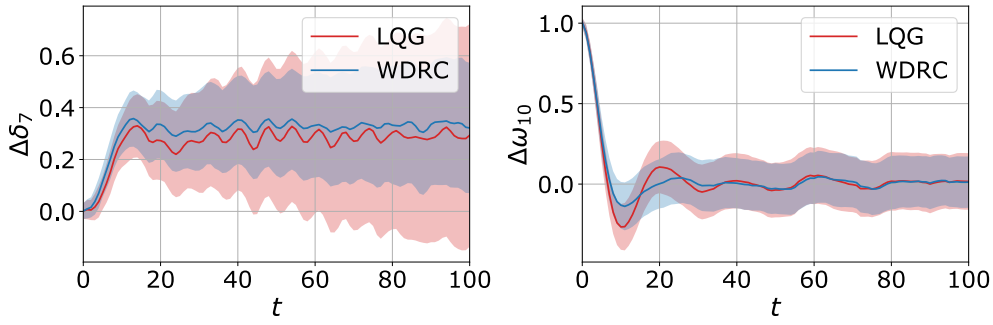
In this section, the performance of our infinite-horizon WDRC method is evaluated on a power system frequency regularization problem using the IEEE 39 bus system, which models the New England power grid [42]. The linearized second-order model for power systems has the following form:

$$\begin{bmatrix} \Delta\dot{\delta} \\ \Delta\dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\bar{M}^{-1}\bar{L} & -\bar{M}^{-1}\bar{D} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta\omega \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{M}^{-1} \end{bmatrix} \Delta P, \tag{4.57}$$

where $\bar{M}$ and $\bar{D}$ are the diagonal matrices of inertia and damping coefficients, $\bar{L}$ is the Laplacian matrix of the transmission network. The system state vector $x(t) := [\Delta\delta^\top(t), \Delta\omega^\top(t)]^\top \in \mathbb{R}^{20}$ consists of the rotor angles and frequencies for 10 generators, while the control input $u(t) := \Delta P(t) \in \mathbb{R}^{10}$ is the power injection vector of the generators. It is assumed that only the rotor angle and frequency of the first six generators are measured, i.e., $n_y = 12$ with $C = [I_{12\times 6}, \mathbf{0}_{12\times 4}, I_{12\times 6}, \mathbf{0}_{12\times 4}]$. The continuous-time system (4.57) is discretized by a zero-order hold method with sample time 0.1 seconds. This yields a discrete-time stochastic system model of the form (4.1). A disturbance $w(t)$ drawn from an unknown distribution affects the power system dynamics. Such disturbances arise from fluctuations in net demand, mechanical noise in generators, etc. The results are obtained by running the algorithms for 100 time steps.

Gaussian Case

In these experiments, the initial state distribution is Gaussian with mean $m_0 = [\mathbf{0}_{19}, 1]^\top$ and covariance matrix $M_0 = 0.01 I_{20}$. The true disturbances are drawn from a zero-

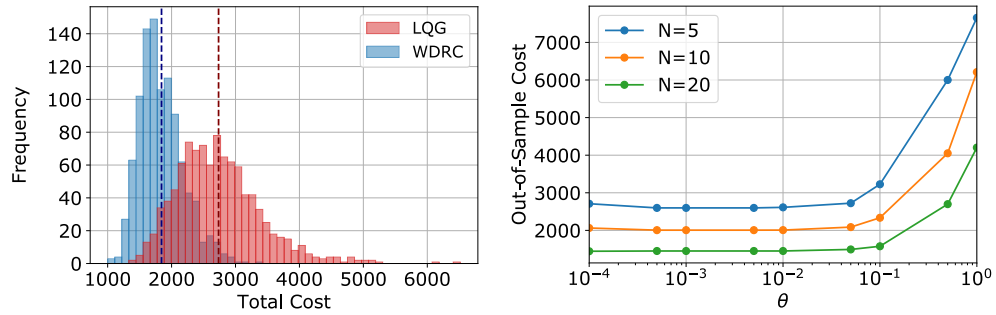


(a) (b)

Figure 4.4: Trajectories of $\Delta\delta_7$ and $\Delta\omega_{10}$ for the system controlled by the LQG and WDRC methods averaged over 1,000 simulation runs in the case of Gaussian disturbances. The shaded regions represent 25% of the standard deviation.

mean Gaussian distribution with a covariance matrix $\Sigma = 0.01I_{20}$, while the observation noise has a covariance $M = 0.01I_{12}$. The nominal distribution is constructed using $N = 5$ disturbance sample data by letting $\hat{\mu} = 0$ and $\hat{\Sigma}$ be the empirical covariance matrix. We select the penalty parameter λ by minimizing the upper-bound in (4.49) for $\theta = 10^{-3}$.

Fig. 4.4 displays the state trajectories of $\Delta\delta_7$ and $\Delta\omega_{10}$, which are both unobservable states, controlled by the WDRC and LQG methods. The results are averaged over 1,000 simulation runs. These results indicate that the WDRC method reduces the fluctuations and the large variance in the rotor angle and removes unnecessary undershoot in the frequency. Besides, our method successfully keeps the states stable despite the inaccurate nominal distribution. The total cost and the computation time for running the whole algorithm are reported in Table 4.2. The WDRC method yields a lower average total cost with a smaller variance over the simulations than the LQG method. Furthermore, the computation times for running the two methods are almost identical, as the computationally expensive SDP problem and the Riccati equations are solved in the offline stage, making the complexity of the online stage similar for both algorithms.



(a) (b)

Figure 4.5: (a) Histogram of the total costs incurred by the LQG and WDRC methods, and (b) out-of-sample performance of WDRC in the case of Gaussian disturbances. The dashed lines represent the sample means of the costs returned by the two methods.

Fig. 4.5 (a) displays the distribution of total costs computed for 1,000 simulation runs. It reveals that for WDRC, the overall distribution is concentrated in the low-cost region. In contrast, the total costs induced by the LQG controller are comparatively higher as it relies on the nominal disturbance distribution, disregarding possible inaccuracies due to the small sample size. Meanwhile, our WDRC method penalizes deviations of the true distribution from the nominal one, thereby making the controller more robust against distributional uncertainties.

Fig. 4.5 (b) shows the out-of-sample cost incurred by our method for different values of the ambiguity set radius θ and various sample sizes of the dataset $\mathbf{w}$ estimated for 10,000 disturbance samples drawn from the true distribution and averaged over 1,000 independent simulation runs. For each θ, the penalty parameter $\lambda(\theta)$ is found according to (4.50). We observe that the cost slightly decreases as the radius increases up to $\theta = 10^{-3}$. The cost starts growing for $\theta \in [10^{-3}, 10^{0}]$. This is because a large θ encourages the controller to be overly conservative, while the controller with a small θ is not sufficiently robust.

As part of these experiments, we also examine the effect of partial observability on the control performance. Specifically, Fig. 4.6 shows the total costs incurred by the

Table 4.2: Total cost and online computation time averaged over 1,000 simulations in the infinite-horizon settings.

	Total Cost		**Computation Time**	
	WDRC	LQG	WDRC	LQG
Gaussian	1842.640	2735.015	0.113	0.115
	(341.836)	(661.369)	(0.019)	(0.014)
Uniform	1891.211	2653.224	0.0184	0.0183
	(394.855)	(767.445)	(0.003)	(0.002)

WDRC and LQG methods under a varying number of observable generators. It can be seen that regardless of the number of observable generators, our method outperforms LQG. Overall, the total cost decreases as more generators become observable, resulting in smaller mean and variance values.

Uniform Case

In this scenario, the true disturbances in each dimension follow a uniform distribution $\mathcal{U}(-0.15, 0.15)$. The initial state distribution is also uniform, $\mathcal{U}(-0.05, 0.05)$ for all states, except $\Delta\omega_{10}$, for which the initial state is selected from $\mathcal{U}(0.95, 1.05)$. The nominal distribution is constructed using $N = 5$ sample data drawn from the true distribution with its mean and covariance corresponding to the empirical ones. The penalty parameter is chosen by minimizing the upper-bound in (4.49) for $\theta = 10^{-2}$.

The Kalman filter is an optimal estimator only in the Gaussian case. However, we approximate the disturbance distribution by a Gaussian, assuming $w_t^* \sim \mathcal{N}(\bar{w}_{t,ss}^*, \Sigma_{ss}^*)$, and apply the steady-state Kalman filter. Besides, unlike the usual LQG settings, where the observation noise is assumed to be zero-mean Gaussian, we draw it from a uniform distribution $\mathcal{U}(-0.4, 0.4)$ and estimate the covariance matrix from 40 samples. By doing so, we evaluate the capability of our WDRC algorithm in the presence of an

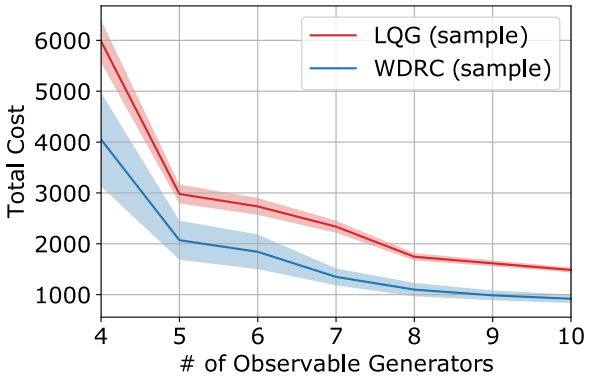
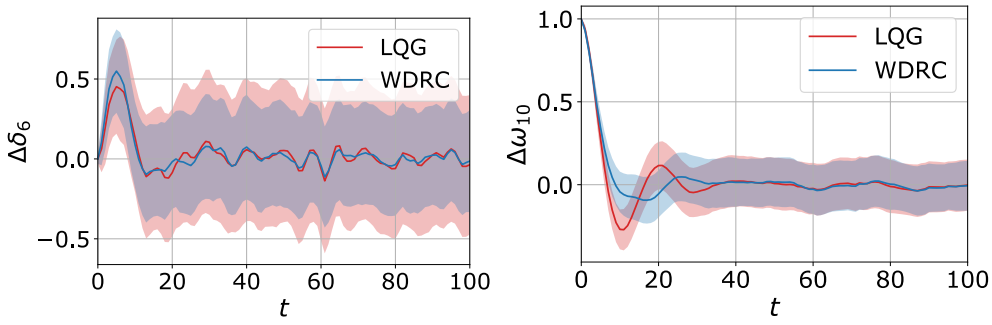


Figure 4.6: Effect of the number of observable generators on the total cost incurred by the LQG and WDRC methods averaged over 1,000 simulation runs in the case of normal disturbances. The shaded regions represent 25% of the standard deviation.

erroneous state estimator.

Fig. 4.7 displays the state trajectories for $\Delta\delta_6$ and $\Delta\omega_{10}$ for the WDRC and LQG methods averaged over 1,000 simulation runs. It shows that LQG results in a larger variance in the trajectory for $\Delta\delta_6$, which is reduced in the WDRC case. In addition, our method smooths the unwanted fluctuations in the trajectory of $\Delta\omega_{10}$ present in the LQG case. The total cost and the computation time for running the algorithm are presented in Table 4.2. Our WDRC method outperforms the LQG method in total cost, inducing a lower average cost with a smaller variance.

The distribution of total costs computed for 1,000 simulation runs is presented as a histogram in Fig. 4.8 (a). Overall, the total costs incurred by WDRC are smaller than the ones induced by the LQG method. Furthermore, the costs for applying the proposed method are concentrated in the low-cost region, whereas the cost distribution for LQR is relatively widespread, covering a large range of costs. This happens because the LQG controller is designed solely using the mean and covariance of the nominal distribution. Furthermore, the state estimation is performed for an inaccurate disturbance distribution, aggravating the situation. Our WDRC method resolves these



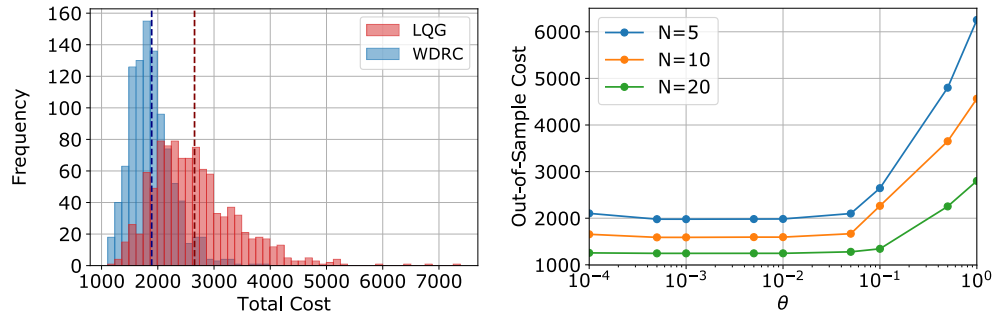
(a) (b)

Figure 4.7: Trajectories of $\Delta\delta_6$ and $\Delta\omega_{10}$ for the system controlled by the LQG and WDRC methods averaged over 1,000 simulation runs in the case of uniform disturbances. The shaded regions represent 25% of the standard deviation.

issues by considering the worst-case disturbance distribution close to the nominal one, thereby anticipating mismatches between the actual and nominal distributions during both the control and estimation stages.

Fig. 4.8 (b) illustrates the total out-of-sample cost induced by our method for different values of θ and N estimated for 10,000 disturbance samples drawn from the true distribution. The results are averaged over 1,000 independent simulation runs. Similar to the previous scenario, the cost slightly decreases as the radius increases up to $\theta = 10^{-3}$ and the cost increases thereafter.

Fig. 4.9 showcases the effect of distributional uncertainties in measurement noise. Specifically, it demonstrates the total costs incurred by the WDRC and LQG methods for measurement noise covariance matrix M estimated using different samples. It is evident that even for only 10 samples, the average performance of WDRC reaches that of LQG with fully known measurement noise distribution. These results illustrate the capabilities of our method to account for erroneous measurement noise information although the proposed controller is designed to achieve distributional robustness in terms of disturbances. Using the worst-case distribution in the state estimator in our approach



(a) (b)

Figure 4.8: (a) Histogram of the total costs incurred by the LQG and WDRC methods, and (b) out-of-sample cost of WDRC in the case of uniform disturbances.

induces additional robustness to the Kalman filter, yielding better overall performance even for a small sample size compared to the standard LQG control method.

4.7 Conclusions

In this work, we have presented a novel WDRC method for discrete-time partially observable linear systems. We have proposed an approximation scheme for reformulating the original WDRC problem into a tractable one. The approximate problem is first solved in finite-horizon settings, resulting in a closed-form expression of the optimal control policy with the corresponding Riccati equation. The mean vector of the worst-case distribution is also found in closed form, while the covariance matrix is found as the solution of a tractable SDP problem. The results for the finite-horizon problem were extended to the infinite-horizon setting by observing the asymptotic behaviors of the optimal policy pair and the cost. Consequently, we obtained a steady-state control policy by solving an ARE. The proposed method has several salient features, such as guaranteed cost property, probabilistic out-of-sample performance guarantee, and closed-loop stability. The experiment results demonstrate the capabilities of our method to immunize partially observable linear systems against distributional ambi-

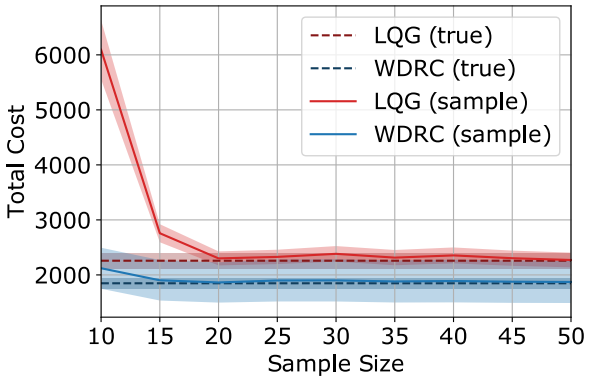


Figure 4.9: Effect of measurement noise uncertainty on the total cost incurred by the LQG and WDRC methods averaged over 1,000 simulation runs in the case of uniform disturbances. The shaded regions represent 25% of the standard deviation.

guity.

4.8 Appendix

4.8.1 Intractability of Minimax LQ Control Problems with Wasserstein Penalty under Partial Observations

Consider the partially observable system (4.1) and the corresponding minimax control problem (4.5) in a finite horizon:

$$\min_{\pi\in\Pi}\max_{\gamma\in\Gamma}\tilde{J}_T^{\lambda}(\pi,\gamma),$$

where

$$\tilde{J}_T^{\lambda}(\pi,\gamma)=\mathbb{E}_{\mathbf{y}}\left[\mathbb{E}_{x_T}[x_T^{\top}Q_f x_T\mid I_T]+\sum_{t=0}^{T-1}\mathbb{E}_{x_t}[x_t^{\top}Qx_t+u_t^{\top}Ru_t\mid I_t,u_t]-\lambda W_2(\mathbb{P}_t,\mathbb{Q}_t)^2\right].$$

To solve the minimax control problem using DP, we define the value function recursively as follows:

$$\begin{aligned}\tilde{V}_t(I_t) := \inf_{u_t \in \mathbb{R}^{n_u}} \sup_{\mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_{x_t}[x_t^\top Q x_t + u_t^\top R u_t - \lambda W_2(\mathbb{P}_t, \mathbb{Q}_t)^2 \\ + \mathbb{E}_{y_{t+1}}[\tilde{V}_{t+1}(I_t, y_{t+1}, u_t) \mid I_t, u_t]\end{aligned} \tag{4.58}$$

with

$$\tilde{V}_T(I_T) := \mathbb{E}_{x_T}[x_T^\top Q_f x_T \mid I_T].$$

In fully observable settings, a common approach to solving the inner maximization problem in (4.58) is to use Kantorovich duality [40]. The most tractable case is when the nominal distribution $\mathbb{Q}_t$ is chosen as the empirical distribution (4.51). In this case, Kantorovich duality can be expressed as

$$\begin{aligned}\sup_{\mathbb{P} \in \mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_w[f(x,w)] - \lambda W_2(\mathbb{P}, \mathbb{Q})^2 \\ = \frac{1}{N} \sum_{i=1}^{N} \sup_{w \in \mathbb{R}^{n_x}} \left\{ f(x,w) - \lambda \|\hat{w}^{(i)} - w\|^2 \right\},\end{aligned} \tag{4.59}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$ is some function depending on the disturbance w and some fixed parameters x.

However, unlike the fully observable case, the uncertainty of the system is represented by the output y_{t+1} and not w_t directly. Therefore, if we can write the value function (4.58) in a way that has the form of the left-hand side in (4.59), then Kantorovich duality can be applied analogously to the fully observable settings. To this end, we recursively solve (4.58) to check whether a specific form of the value function is preserved. For time $t = T-1$, the value function is given by

$$\begin{aligned}\tilde{V}_{T-1}(I_{T-1}) = \inf_{u_{T-1} \in \mathbb{R}^{n_u}} \mathbb{E}_{x_{T-1}}[x_{T-1}^\top Q x_{T-1} \mid I_{T-1}] + u_{T-1}^\top R u_{T-1} \\ + \sup_{\mathbb{P}_{T-1} \in \mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_{x_{T-1}, w_{T-1}}[(Ax_{T-1} + Bu_{T-1} + w_{T-1})^\top \times \\ Q_f(Ax_{T-1} + Bu_{T-1} + w_{T-1}) \mid I_{T-1}, u_{T-1}] - \lambda W_2(\mathbb{P}_{T-1}, \mathbb{Q}_{T-1})^2.\end{aligned}$$

It follows from Kantorovich duality that

$$\begin{aligned}\tilde{V}_{T-1}(I_{T-1}) = &\inf_{u_{T-1}\in\mathbb{R}^{n_u}} \mathbb{E}_{x_{T-1}}[x_{T-1}^\top Q x_{T-1} \mid I_{T-1}] + u_{T-1}^\top R u_{T-1} \\ &+ \frac{1}{N}\sum_{i=1}^{N} \sup_{w_{T-1}\in\mathbb{R}^{n_x}} \big\{\mathbb{E}_{x_{T-1}}(Ax_{T-1}+Bu_{T-1}+w_{T-1})^\top \times \\ &Q_f(Ax_{T-1}+Bu_{T-1}+w_{T-1}) \mid I_{T-1}, u_{T-1}] - \lambda\|\hat{w}_{T-1}^{(i)} - w_{T-1}\|^2\big\}.\end{aligned}$$

If the penalty parameter satisfies the condition $\lambda I \succ Q_f$, then the inner maximization problem for each $i = 1, \ldots, N$ has a unique maximizer $w^{(i),*}$, given by

$$w_{T-1}^{(i),*} := (\lambda I - Q_f)^{-1}\big[Q_f(A\mathbb{E}_{x_{T-1}}[x_{T-1} \mid I_{T-1}] + Bu_{T-1}) + \lambda\hat{w}_{T-1}^{(i)}\big].$$

Solving the outer minimization problem with respect to u_{T-1} yields the following unique minimizer:

$$\begin{aligned}u_{T-1}^* = &-R^{-1}B^\top(I + Q_f B R^{-1} B^\top - \frac{1}{\lambda}Q_f)^{-1} \\ &\times \big(A\mathbb{E}_{x_{T-1}}[x_{T-1} \mid I_{T-1}] + \frac{1}{N}\sum_{i=1}^{N}\hat{w}_{T-1}^{(i)}\big).\end{aligned}$$

Then, the value function at time $t = T-1$ has the following quadratic form:

$$\tilde{V}_{T-1} = \mathbb{E}_{x_{T-1}}[x_{T-1}^\top P_{T-1} x_{T-1} + \xi_{T-1}^\top S_{T-1}\xi_{T-1} + 2r_{T-1}^\top x_{T-1} \mid I_{T-1}] + q_{T-1},$$

where $\xi_{T-1} = x_{T-1} - \mathbb{E}_{x_{T-1}}[x_{T-1} \mid I_{T-1}]$ is the difference between the state and its estimate, while $P_{T-1}, S_{T-1} \in \mathbb{S}_+^{n_x}$, $r_{T-1} \in \mathbb{R}^{n_x}$ and $q_{T-1} \in \mathbb{R}$ are coefficients.

Continuing the recursion for $t = T-2$, the value function is written as

$$\begin{aligned}\tilde{V}_{T-2}(I_{T-2}) = &\inf_{u_{T-2}\in\mathbb{R}^{n_u}} \mathbb{E}_{x_{T-2}}[x_{T-2}^\top Q x_{T-2} \mid I_{T-2}] + u_{T-2}^\top R u_{T-2} \\ &+ \sup_{\mathbb{P}_{T-2}\in\mathcal{P}(\mathbb{R}^{n_x})} \mathbb{E}_{x_{T-2},w_{T-2}}[(Ax_{T-2}+Bu_{T-2}+w_{T-2})^\top \\ &\times P_{T-1}(Ax_{T-2}+Bu_{T-2}+w_{T-2}) \\ &+ 2r_{T-1}^\top(Ax_{T-2}+Bu_{T-2}+w_{T-2}) \mid I_{T-2}, u_{T-2}] \\ &+ \mathbb{E}_{y_{T-1},x_{T-1}}[\xi_{T-1}^\top S_{T-1}\xi_{T-1} \mid I_{T-1}] \\ &+ q_{T-1} - \lambda W_2(\mathbb{P}_{T-2}, \mathbb{Q}_{T-2})^2.\end{aligned}$$

Due to the structure of the expression inside the maximization, it is straightforward that the value function does not have the form in (4.59). This is because the term $\mathbb{E}_{y_{T-1},x_{T-1}}[\xi_{T-1}^\top S_{T-1}\xi_{T-1} \mid I_{T-1}]$ cannot be represented by an expectation with respect to w_{T-2}, though it implicitly depends on the disturbances via x_{T-1} and y_{T-1}. Consequently, the standard LQR argument is not applicable to the minimax problem with the Wasserstein penalty under partial observations.

4.8.2 Proofs

Proof of Lemma 4.2

Proof. Having the quadratic value function for time $t+1$ and plugging it into (4.12), the value function for time t is given by

$$\begin{aligned} V_t(I_t) = \inf_{u_t\in\mathbb{R}^{n_u}} \sup_{\substack{\bar{w}_t\in\mathbb{R}^{n_x},\\ \Sigma_t\in\mathbb{S}_+^{n_x}}} & \mathbb{E}_{x_t}[x_t^\top Q x_t \mid I_t] + u_t^\top R u_t \\ & + \mathbb{E}_{x_t,w_t}\big[(Ax_t+Bu_t+w_t)^\top P_{t+1}(Ax_t+Bu_t+w_t) \\ & + 2r_{t+1}^\top(Ax_t+Bu_t+w_t) \mid I_t, u_t\big] + \mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1}^\top S_{t+1}\xi_{t+1} \mid I_t] \\ & + q_{t+1} - \lambda[\|\bar{w}_t-\hat{w}_t\|^2 + \mathrm{B}^2(\Sigma_t,\hat{\Sigma}_t)]. \end{aligned}$$

Using the property that

$$\mathbb{E}[w_t^\top P_{t+1} w_t] = \bar{w}_t^\top P_{t+1}\bar{w}_t + \mathrm{Tr}[P_{t+1}\Sigma_t],$$

we further simplify the value function as

$$\begin{aligned} V_t(I_t) = \inf_{u_t\in\mathbb{R}^{n_u}} \sup_{\substack{\bar{w}_t\in\mathbb{R}^{n_x},\\ \Sigma_t\in\mathbb{S}_+^{n_x}}} & \mathbb{E}_{x_t}[x_t^\top Q x_t \mid I_t] + u_t^\top R u_t \\ & + \mathbb{E}_{x_t}\big[(Ax_t+Bu_t+\bar{w}_t)^\top P_{t+1}(Ax_t+Bu_t+\bar{w}_t) \\ & + 2r_{t+1}^\top(Ax_t+Bu_t+\bar{w}_t) \mid I_t, u_t\big] - \lambda\|\bar{w}_t-\hat{w}_t\|_2^2 + \mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1}^\top S_{t+1}\xi_{t+1} \mid I_t] \\ & + \mathrm{Tr}[(P_{t+1}-\lambda I)\Sigma_t] + 2\lambda\mathrm{Tr}[(\hat{\Sigma}_t^{1/2}\Sigma_t\hat{\Sigma}_t^{1/2})^{1/2}] - \lambda\mathrm{Tr}[\hat{\Sigma}_t] + q_{t+1}. \end{aligned}$$

Note that

$$\mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1} \mid I_t] = 0,$$

and $\mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1}\xi_{t+1}^\top \mid I_t]$ is independent of u_t and $\bar{w}_t$. Thus, the objective function for the inner maximization problem

$$\begin{aligned}\mathbb{E}_{x_t}\big[(Ax_t+Bu_t+\bar{w}_t)^\top P_{t+1}(Ax_t+Bu_t+\bar{w}_t)+2r_{t+1}^\top(Ax_t+Bu_t+\bar{w}_t)\mid I_t,u_t\big]\\ -\lambda\|\bar{w}_t-\hat{w}_t\|_2^2+\mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1}^\top S_{t+1}\xi_{t+1}\mid I_t]\\ +\mathrm{Tr}[(P_{t+1}-\lambda I)\Sigma_t]+2\lambda\mathrm{Tr}[(\hat{\Sigma}_t^{1/2}\Sigma_t\hat{\Sigma}_t^{1/2})^{1/2}]\end{aligned}$$

can be written separately in terms of $\bar{w}_t$ and Σ_t, enabling to solve two independent maximization problems. Specifically, the two problems are as follows:

$$\begin{aligned}\max_{\bar{w}_t\in\mathbb{R}^{n_x}}\ \mathbb{E}_{x_t}\big[(Ax_t+Bu_t+\bar{w}_t)^\top P_{t+1}(Ax_t+Bu_t+\bar{w}_t)\\ +2r_{t+1}^\top(Ax_t+Bu_t+\bar{w}_t)\mid I_t,u_t\big]-\lambda\|\bar{w}_t-\hat{w}_t\|_2^2\end{aligned}$$

and

$$\max_{\Sigma_t\in\mathbb{S}_+^{n_x}}\ \mathbb{E}_{x_{t+1},y_{t+1}}[\xi_{t+1}^\top S_{t+1}\xi_{t+1}\mid I_t]+\mathrm{Tr}[(P_{t+1}-\lambda I)\Sigma_t+2\lambda(\hat{\Sigma}_t^{1/2}\Sigma_t\hat{\Sigma}_t^{1/2})^{1/2}].$$

Regarding the first problem for $\bar{w}_t$, the Hessian of value function with respect to $\bar{w}_t$ is negative definite under the assumption on the penalty parameter λ. Thus, the objective is strictly concave, and its unique maximizer given control input u_t is obtained from the first-order optimality condition as

$$\bar{w}_t^*(u_t)=(\lambda I-P_{t+1})^{-1}\big(P_{t+1}[A_t\bar{x}_t+B_tu_t]+r_{t+1}\big).$$

Note that the maximizer of the second problem is independent of the control input u_t. For the outer minimization problem with respect to u_t, we first differentiate the objective function with respect to $u_t\in\mathbb{R}^{n_u}$ to obtain the following derivative:

$$\begin{aligned}&2\left[B+\frac{\partial\bar{w}_t^*(u_t)}{\partial u_t}\right]^\top[P_{t+1}(A\bar{x}_t+Bu_t+\bar{w}_t^*(u_t))+r_{t+1}]\\ &-2\lambda\frac{\partial\bar{w}_t^*(u_t)}{\partial u_t}(\bar{w}_t^*(u_t)-\hat{w}_t)+2Ru_t\\ &=2B^\top g_t(u_t)+2Ru_t,\end{aligned}$$

where

$$g_t(u_t):=P_{t+1}(A\bar{x}_t+Bu_t+\bar{w}_t^*(u_t))+r_{t+1}.$$

Differentiating the derivative with respect to u_t again, we can check that the Hessian of the objective function is positive definite under the assumption on the penalty parameter λ. Thus, the unique minimizer u_t^* can be obtained by using the first-order optimality condition:

$$u_t^* = -R^{-1}B^\top g_t^*. \tag{4.60}$$

For further simplifications, we let $\bar{w}_t^* = \bar{w}_t^*(u_t^*)$ and rewrite it as

$$\bar{w}_t^* = \frac{1}{\lambda}\big(P_{t+1}(A\bar{x}_t + Bu_t^* + \bar{w}_t^*) + r_{t+1} + \lambda\hat{w}_t\big),$$

which yields the following expression for g_t^*:

$$g_t^* = P_{t+1}\left(A\bar{x}_t - BR^{-1}B^\top g_t^* + \frac{1}{\lambda}g_t^* + \hat{w}_t\right) + r_{t+1}.$$

Finally, we have

$$g_t^* = (I + P_{t+1}\Phi)^{-1}(P_{t+1}A\bar{x}_t + P_{t+1}\hat{w}_t + r_{t+1}) \tag{4.61}$$

and

$$\bar{w}_t^* = \frac{1}{\lambda}g_t^* + \hat{w}_t.$$

We conclude the proof by replacing (4.61) into (4.60). □

Proof of Theorem 4.1

Proof. We use mathematical induction backward in time to prove the theorem. For $t = T$, by definition, the value function is in the desired form

$$V_T(I_T) = \mathbb{E}_{x_T}[x_T^\top P_T x_T | I_T].$$

Now, it suffices to show that V_t is in the required form, given that V_{t+1} is in that

form. Specifically, the value function at time t can be written as

$$\begin{aligned}
V_t(I_t) = \inf_{u_t \in \mathbb{R}^{n_u}} \sup_{\substack{\bar{w}_t \in \mathbb{R}^{n_x}, \\ \Sigma_t \in \mathbb{S}_+^{n_x}}} & \mathbb{E}_{x_t}[x_t^\top Q x_t \mid I_t] + u_t^\top R u_t \\
& + \mathbb{E}_{x_t, w_t}\big[(Ax_t + Bu_t + w_t)^\top P_{t+1}(Ax_t + Bu_t + w_t) \\
& + 2r_{t+1}^\top (Ax_t + Bu_t + w_t) \mid I_t, u_t\big] + \mathbb{E}_{x_{t+1}, y_{t+1}}[\xi_{t+1}^\top S_{t+1} \xi_{t+1} \mid I_t] \\
& - \lambda[\|\bar{w}_t - \hat{w}_t\|^2 + \mathrm{B}^2(\Sigma_t, \hat{\Sigma}_t)] + q_{t+1} + \sum_{s=t+1}^{T-1} \mathbb{E}_{y_{t+1}}[z_t(I_t, u_t, y_{t+1}, s) \mid I_t, u_t].
\end{aligned}$$

It follows from the law of total expectation that

$$\mathbb{E}_{y_{t+1}}[z_{t+1}(I_t, u_t, y_{t+1}, s) \mid I_t, u_t] = z_t(I_t, s),$$

which is independent of $\bar{w}_t$, Σ_t, and u_t. Therefore, by Lemma 4.2, the mean vector (4.16) and the covariance matrix solving (4.19) are an optimal solution pair of the inner maximization problem. Moreover, the optimal value of (4.19) corresponds to $z_t(I_t, t)$. Meanwhile, the outer minimization problem has a unique optimal solution given as (4.13). By plugging these values into the Bellman equation, we have

$$\begin{aligned}
V_t(I_t) = & \mathbb{E}_{x_t}[x_t^\top Q x_t \mid I_t] + (g_t^*)^\top B R^{-1} B^\top g_t^* - \frac{1}{\lambda}(g_t^*)^\top g_t^* - \lambda \mathrm{Tr}[\hat{\Sigma}_t] + q_{t+1} \\
& + \mathbb{E}_{x_t}\big[(Ax_t - \Phi g_t^* + \hat{w}_t)^\top P_{t+1}(Ax_t - \Phi g_t^* + \hat{w}_t) \\
& + 2r_{t+1}^\top (Ax_t - \Phi g_t^* + \hat{w}_t) \mid I_t, u_t\big] + z_t(I_t, t) + \sum_{s=t+1}^{T-1} z_t(I_t, s).
\end{aligned}$$

It remains to simplify the expression by substituting the values for r_t and q_t as in (4.22) and (4.23). Then, the value function for time t can be written as

$$V_t(I_t) = \mathbb{E}_{x_t}[x_t^\top (Q + A^\top P_{t+1} A) x_t \mid I_t] - \bar{x}_t^\top S_t \bar{x}_t + 2r_t^\top \bar{x}_t + q_t + \sum_{s=t}^{T-1} z_t(I_t, s),$$

where $S_t = A^\top P_{t+1}\Phi(I + P_{t+1}\Phi)^{-1}P_{t+1}A$. This can be expressed as

$$\begin{aligned} V_t(I_t) =& \mathbb{E}_{x_t}[x_t^\top (Q + A^\top P_{t+1}A - S_t)x_t \mid I_t] + \mathbb{E}_{x_t}[\xi_t^\top S_t \xi_t + 2r_t^\top x_t \mid I_t] \\ &+ q_t + \sum_{s=t}^{T-1} z_t(I_t, s) \\ =& \mathbb{E}_{x_t}[x_t^\top P_t x_t + \xi_t^\top S_t \xi_t + 2r_t^\top x_t \mid I_t] + q_t + \sum_{s=t}^{T-1} z_t(I_t, s), \end{aligned}$$

which is in the desired form with parameters (4.20)–(4.24). This completes our inductive argument.

So far, we have shown that the value function is measurable, and the outer minimization problem in the Bellman equation (4.12) admits an optimal solution. Thus, it follows from the DP principle that the control policy π^* constructed as that in the theorem statement is optimal. Moreover, if (4.24) admits an optimal solution Σ_t^* for all t, the policy pair (π_t^*, γ_t^*) is minimax optimal. □

Proof of Proposition 4.1

First, we notice that $\bar{X}_{t+1} = \mathbb{E}_{x_{t+1}, y_{t+1}}[\xi_{t+1}\xi_{t+1}^\top \mid I_t]$. It follows from the Kalman filter recursion (4.26) and (4.27) that $z_t(I_t, t)$ is equal to the optimal value of (4.19), which in its turn is equivalent to the following optimization problem:

$$\begin{aligned} \max_{\substack{X, X^-, \\ \Sigma \in \mathbb{S}_+^{n_x}}} \ & \mathrm{Tr}[S_{t+1}X + (P_{t+1} - \lambda I)\Sigma + 2\lambda(\hat{\Sigma}_t^{1/2}\Sigma\hat{\Sigma}_t^{1/2})^{1/2}] \\ \text{s.t. } & X = X^- - X^- C^\top (CX^- C^\top + M)^{-1} CX^- \\ & X^- = A\bar{X}_t A^\top + \Sigma, \end{aligned}$$

where $\bar{X}_t$ is the state covariance matrix conditioned on the information vector I_t. The objective function here is continuous and jointly concave in Σ, X^- and X due to the positive semidefiniteness of S_{t+1} and Assumption 4.1. Therefore, the problem has an optimal solution and we can obtain optimal (Σ^*, X^*), corresponding to Σ_t^* and $\bar{X}_{t+1}$. The reformulation into the SDP form (4.28) is performed by using the property that

$\mathrm{Tr}[S_{t+1}X] \leq \mathrm{Tr}[S_{t+1}X']$ for any $X \preceq X'$ and then applying the Schur complement lemma to replace the inequality constraints with the corresponding linear matrix inequalities.

Proof of Proposition 4.2

Proof. The proof follows from the asymptotic property of the Riccati equation for the standard LQ control. Specifically, we rewrite the Riccati equation (4.20) as follows:

$$\begin{aligned} P_t &= Q + A^\top (I + P_{t+1}\Phi^{1/2} I^{-1} (\Phi^{1/2})^\top)^{-1} P_{t+1} Q \\ &= Q + A^\top (P_{t+1} - P_{t+1}\tilde{B}(\tilde{R} + \tilde{B}^\top P_{t+1}\tilde{B})^{-1}\tilde{B}^\top P_{t+1}) A, \end{aligned} \tag{4.62}$$

where $\tilde{R} = I$, $\tilde{B} = \Phi^{1/2}$. Consider a hypothetical linear system $(A, \tilde{B})$ with a quadratic cost function replacing R with $\tilde{R}$. It is evident that (4.62) has the form of the standard Riccati equation for this hypothetical LQ control problem. It follows from the standard LQ control theory that if the pair $(A, \tilde{B})$ is stabilizable and $(A, Q^{1/2})$ is detectable, then there exists a $P_{ss} \succeq 0$ such that (4.29) holds for any $P_T \succeq 0$. Furthermore, it is the unique solution of the ARE (4.30) [4, Proposition 3.1.1]. □

Proof of Lemma 4.3

Proof. It follows from Proposition 4.2 that $P_t \to P_{ss}$ as $T \to \infty$, and thus the convergence of $\{S_t\}$ to S_{ss} is straightforward. Moreover, r_t is updated according to

$$r_t = A^\top (I + P_{ss}\Phi)^{-1}(r_{t+1} + P_{ss}\hat{w})$$

as $T \to \infty$. Thus, to ensure the convergence of $\{r_t\}$, it suffices to show that $A^\top (I + P_{ss}\Phi)^{-1}$. For this, we revisit the proof of Proposition 4.2 and notice that the ARE can be expressed as

$$P_{ss} = Q + A^\top (P_{ss} - P_{ss}\tilde{B}(\tilde{R} + \tilde{B}^\top P_{ss}\tilde{B})^{-1}\tilde{B}^\top P_{ss}) A,$$

where $\tilde{R} = I$ and $\tilde{B} = \Phi^{1/2}$. Then, the optimal control gain matrix for the hypothetical LQ control problem for the linear system $(A, \tilde{B})$ with a quadratic cost function

replacing R with $\tilde{R}$ is given by

$$\tilde{K} = (\tilde{R} + \tilde{B}^\top P_{ss} \tilde{B})^{-1} \tilde{B}^\top P_{ss} A,$$

and the closed-loop "A" matrix is

$$A + \tilde{B}\tilde{K} = A - \tilde{B}(\tilde{R} + \tilde{B}^\top P_{ss} \tilde{B})^{-1} \tilde{B}^\top P_{ss} A,$$

which is stable because $(A, \tilde{B})$ is stabilizable. Since $A^\top (I + P_{ss}\Phi)^{-1} = (A + \tilde{B}\tilde{K})^\top$, it is also a stable matrix. Therefore, $\{r_t\}$ converges to its limit, which is obtained as (4.32). □

Proof of Proposition 4.3

Proof. It follows from Theorem 4.1 that the finite-horizon cost incurred by the policy pair $(\pi^*_{ss}, \gamma^*_{ss})$ is given by

$$\begin{aligned} J^\lambda_T(\pi^*_{ss}, \gamma^*_{ss}) &= \mathbb{E}_{y_0}\big[\mathbb{E}_{x_0}[x_0^\top P_0 x_0 + \xi_0^\top S_0 \xi_0 + 2 r_0^\top x_0 \mid I_0]\big] + q_0 \\ &\quad + \sum_{t=0}^{T-1} \Big(\mathrm{Tr}[S_{t+1} \bar{X}_{t+1} + (P_{t+1} - \lambda I)\Sigma^*_{ss}] + 2\lambda \mathrm{Tr}[(\hat{\Sigma}^{1/2} \Sigma^*_{ss} \hat{\Sigma}^{1/2})^{1/2}] \Big), \end{aligned}$$

where $\bar{X}_{t+1}$ is the state covariance matrix computed using Σ^*_{ss}. It follows from (4.41) that $\{\bar{X}_{t+1}\}$ converges to $\bar{X}_{ss}$ as $T \to \infty$. By the convergence of P_t, S_t, and r_t, as well as the recursion for q_t, the steady-state average cost is given by

$$\begin{aligned} \rho &= \limsup_{T \to \infty} \frac{1}{T} J^\lambda_T(\pi^*_{ss}, \gamma^*_{ss}) \\ &= \mathrm{Tr}[S_{ss} \bar{X}_{ss} + (P_{ss} - \lambda I)\Sigma^*_{ss} + 2\lambda(\hat{\Sigma}^{1/2} \Sigma^*_{ss} \hat{\Sigma}^{1/2})^{1/2}] \\ &\quad + (2\hat{w} - \Phi r_{ss})^\top (I + P_{ss}\Phi)^{-1} r_{ss} - \lambda \mathrm{Tr}[\hat{\Sigma}] + \hat{w}^\top (I + P_{ss}\Phi)^{-1} P_{ss} \hat{w}. \end{aligned}$$

The first term in the last equation corresponds to the optimal value z_{ss} of the maximization problem (4.39). Therefore, the result follows. □

Proof of Proposition 4.4

Proof. We first rewrite h as

$$h(I_t) = \mathbb{E}_{x_t}[x_t^\top P_{ss} x_t + \xi_t^\top S_{ss} \xi_t + 2 r_{ss}^\top x_t \mid I_t]$$

with $\mathbb{E}_{x_t}[\xi_t\xi_t^\top \mid I_t] = X_t \equiv \bar{X}_{ss}$. Next, we apply Lemma 4.2 by letting $V_{t+1} \equiv h$, or, by setting $P_{t+1} = P_{ss}, S_{t+1} = S_{ss}, r_{t+1} = r_{ss}$, and $q_{t+1} = 0$. Then, the minimax problem on the right-hand side of (4.45) has the optimal value of

$$\mathbb{E}_{x_t}[x_t^\top P_t x_t + \xi_t^\top S_t \xi_t + 2r_t^\top x_t \mid I_t] + q_t + z_t(I_t, t),$$

where

$$\begin{aligned}
P_t &= Q + A^\top (I + P_{ss}\Phi)^{-1} P_{ss} A \\
S_t &= Q + A^\top P_{ss} A - P_{ss} \\
r_t &= A^\top (I + P_{ss}\Phi)^{-1}(r_{ss} + P_{ss}\hat{w}) \\
q_t &= (2\hat{w} - \Phi r_{ss})^\top (I + P_{ss}\Phi)^{-1} r_{ss} + \hat{w}^\top (I + P_{ss}\Phi)^{-1} P_{ss}\hat{w} - \lambda \mathrm{Tr}[\hat{\Sigma}],
\end{aligned}$$

and

$$z_t(I_t, t) = \sup_{\Sigma_t \in \mathbb{S}_+^{n_x}} \mathrm{Tr}[S_{ss}\bar{X}_{t+1}] + \mathrm{Tr}[(P_{ss} - \lambda I)\Sigma_t + 2\lambda(\hat{\Sigma}^{1/2}\Sigma_t\hat{\Sigma}^{1/2})].$$

It follows from the ARE (4.30) that $P_t = P_{ss}$, while from (4.31) and (4.32) we have $S_t = S_{ss}$ and $r_t = r_{ss}$, respectively. Since $\bar{X}_{t+1} = \bar{X}_{ss}$ is stationary, the maximization problem (4.39) yields $z_t(I_t, t) = z_{ss}$ with its maximizer corresponding to the stationary covariance matrix Σ_{ss}^*. Moreover, we have

$$\begin{aligned}
\bar{X}_{ss} &= \bar{X}_{t+1}^- - \bar{X}_{t+1}^- C^\top (C\bar{X}_{t+1}^- C^\top + M)^{-1} C\bar{X}_{t+1}^- \\
\bar{X}_{t+1}^- &= A\bar{X}_t A^\top + \Sigma_{ss}^*,
\end{aligned}$$

which is valid only if $\bar{X}_t = \bar{X}_{ss}$. As a result, the optimal value of the minimax problem is equal to

$$\bar{x}_t^\top P_{ss}\bar{x}_t + 2r_{ss}^\top \bar{x}_t + \mathrm{Tr}[(S_{ss} + P_{ss})\bar{X}_{ss}] + q_t + z_{ss}.$$

Thus, the equality in (4.45) holds. The optimality of the solution pair $(\pi_{ss}^*(I_t), \gamma_{ss}^*(I_t))$ follows directly from Lemma 4.2. □

Proof of Proposition 4.5

Proof. Fix an arbitrary control policy $\pi := (\pi_0, \pi_1, \dots) \in \Pi$. We first show that

$$\bar{J}_T^{\lambda}(\pi, \gamma_{ss}^*) \geq T\rho + \mathbb{E}_{y_0}[h(I_0)] \tag{4.63}$$

using mathematical induction. For $T = 0$, $\bar{J}_0^{\lambda}(\pi, \gamma_{ss}^*) = \mathbb{E}_{y_0}[h(I_0)]$. Suppose that the induction hypothesis is true for $T = k$. When $T = k + 1$, it follows from Proposition 4.4 that

$$\begin{aligned}\bar{J}_{k+1}^{\lambda}(\pi, \gamma_{ss}^*) &\geq \bar{J}_k^{\lambda}(\pi, \gamma_{ss}^*) - \mathbb{E}_{y_{0:k}}[h(I_k)] + \rho + \mathbb{E}_{y_{0:k}}[h(I_k)] \\ &\geq (k+1)\rho + \mathbb{E}_{y_0}[h(I_0)].\end{aligned}$$

This completes our inductive argument.

Dividing both sides of (4.63) by T and taking $\limsup$, we obtain that

$$\bar{J}_{\infty}^{\lambda}(\pi, \gamma_{ss}^*) \geq \rho, \tag{4.64}$$

which holds for any control policy $\pi \in \Pi$.

Now, for any $\pi \in \bar{\Pi}$, the left-hand side of (4.64) is equivalent to

$$\begin{aligned}\bar{J}_{\infty}^{\lambda}(\pi, \gamma_{ss}^*) &= \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}[h(I_T) \mid \pi, \gamma_{ss}^*] \\ &\quad + \limsup_{T\to\infty} \frac{1}{T}\mathbb{E}_{\mathbf{y}}\left[\sum_{t=0}^{T-1} \mathbb{E}_{x_t}[x_t^\top Q x_t \mid I_t] + u_t^\top R u_t - \lambda \mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t)^2 \mid \pi, \gamma_{ss}^*\right] \\ &= J_{\infty}^{\lambda}(\pi, \gamma_{ss}^*),\end{aligned} \tag{4.65}$$

with the last equality following from the condition (4.47). Combining (4.64) and (4.65) yields

$$J_{\infty}^{\lambda}(\pi, \gamma_{ss}^*) \geq \rho \quad \forall \pi \in \bar{\Pi}.$$

Using a similar argument, we can show that

$$J_{\infty}^{\lambda}(\pi_{ss}^*, \gamma) \leq \rho \quad \forall \gamma \in \bar{\Gamma}.$$

Therefore, $(\pi_{ss}^*, \gamma_{ss}^*)$ is minimax optimal, and the optimal value corresponds to ρ. $\square$

Proof of Proposition 4.6

Proof. Since $\gamma^* \in \Gamma$, it is admissible to the original minimax control problem (4.5). Also, by Lemma 4.1, if the nominal distribution $\mathbb{Q}_t$ is elliptical, then (4.8) holds with equality, yielding

$$J_\infty^\lambda(\pi, \gamma^*) = \tilde{J}_\infty^\lambda(\pi, \gamma^*) \quad \forall \pi \in \Pi.$$

Therefore,

$$J_\infty^\lambda(\pi^*, \gamma^*) = \inf_{\pi \in \Pi} J_\infty^\lambda(\pi, \gamma^*) \leq \tilde{J}_\infty^\lambda(\pi, \gamma^*) \quad \forall \pi \in \Pi.$$

On the other hand, Lemma 4.1 implies that

$$\begin{aligned} J_\infty^\lambda(\pi^*, \gamma^*) &= \sup_{\gamma \in \Gamma} J_\infty^\lambda(\pi^*, \gamma) \\ &\geq \sup_{\gamma \in \Gamma} \tilde{J}_\infty^\lambda(\pi^*, \gamma) \geq \tilde{J}_\infty^\lambda(\pi^*, \gamma) \quad \forall \gamma \in \Gamma. \end{aligned}$$

Finally, we obtain that

$$\tilde{J}_\infty^\lambda(\pi^*, \gamma) \leq J_\infty^\lambda(\pi^*, \gamma^*) \leq \tilde{J}_\infty^\lambda(\pi, \gamma^*) \quad \forall (\pi, \gamma) \in \Pi \times \Gamma.$$

This implies that (π^*, γ^*) is minimax optimal to the original problem (4.5). □

Proof of Theorem 4.3

Proof. Fix $\lambda > 0$. Let $\mathrm{LHS} := \sup_{\gamma \in \bar{\Gamma}_\mathcal{D}} J_\infty(\pi_{ss}^{\lambda,\star}, \gamma)$ and $\mathrm{RHS} := \theta^2\lambda + \rho(\lambda)$ with $\bar{\Gamma}_\mathcal{D} := \bar{\Gamma} \cap \Gamma_\mathcal{D}$. For any $\varepsilon > 0$, there exists $\gamma^\varepsilon \in \bar{\Gamma}_\mathcal{D}$ such that

$$\mathrm{LHS} - \epsilon < J_\infty(\pi_{ss}^{\lambda,\star}, \gamma^\varepsilon).$$

By Lemma 4.1 and the definition of the Wasserstein ambiguity set $\mathcal{D}_t$, we have

$$\mathrm{G}(\mathbb{P}_t, \mathbb{Q}_t)^2 \leq W_2(\mathbb{P}_t, \mathbb{Q}_t)^2 \leq \theta^2 \quad \forall \mathbb{P}_t \in \mathcal{D}_t.$$

Thus, it follows from $\gamma^\epsilon \in \bar{\Gamma}_\mathcal{D}$ and the definitions of J_∞ and J_∞^λ that

$$\begin{aligned} J_\infty(\pi_{ss}^{\lambda,\star}, \gamma^\varepsilon) &\leq \theta^2\lambda + J_\infty^\lambda(\pi_{ss}^{\lambda,\star}, \gamma^\varepsilon) \\ &\leq \theta^2\lambda + \sup_{\gamma \in \bar{\Gamma}} J_\infty^\lambda(\pi_{ss}^{\lambda,\star}, \gamma) = \theta^2\lambda + \rho(\lambda). \end{aligned}$$

Since ϵ was arbitrarily chosen, $\mathrm{LHS} \leq \mathrm{RHS}$ as desired. □

Proof of Theorem 4.4

Proof. It follows from the measure concentration inequality (4.53) that for a Wasserstein ambiguity set with radius θ chosen according to (4.54), the following probabilistic bound holds:

$$\mathbb{P}^N\{\mathbf{w} \mid W_2(\mathbb{P}, \mathbb{Q}) \leq \theta\} \geq 1 - \beta,$$

meaning that the true distribution $\mathbb{P}$ lies in the ambiguity set with a probability no less than $(1 - \beta)$.

Moreover, Theorem 4.3 suggests

$$J_\infty(\pi_{ss,\mathbf{w}}^{\lambda(\theta),*}, \gamma) \leq \theta^2\lambda(\theta) + \rho(\lambda(\theta)) \quad \forall \gamma \in \bar{\Gamma}_{\mathcal{D}}.$$

Finally, the true distribution $\mathbb{P}$ belongs to the ambiguity set $\mathcal{D}$ with a probability no less than $(1 - \beta)$, the inequality holds with the same probability, thereby concluding the proof. □

Proof of Proposition 4.7

Proof. The mean-state system under the optimal policy $(\pi_{ss}^*, \gamma_{ss}^*)$ can be written as

$$\begin{aligned}
\tilde{x}_{t+1} &= A\tilde{x}_t + (BK_{ss} + H_{ss})\bar{\tilde{x}}_t + BL_{ss} + G_{ss} \\
\bar{\tilde{x}}_{t+1} &= (A + BK_{ss} + H_{ss} - \bar{X}_{ss}C^\top M^{-1}CA)\bar{\tilde{x}}_t \\
&\quad + BL_{ss} + G_{ss} + \bar{X}_{ss}C^\top M^{-1}CA\tilde{x}_t.
\end{aligned} \tag{4.66}$$

Let $e_t := \tilde{x}_t - \bar{\tilde{x}}_t$ be the *error state*, representing the difference between the expected values of the true state and its estimate. Then, the error state evolves according to

$$\begin{aligned}
e_{t+1} &= (A - \bar{X}_{ss}C^\top M^{-1}CA)(\tilde{x}_t - \bar{\tilde{x}}_t) \\
&= (A - \bar{X}_{ss}^- C^\top (C\bar{X}_{ss}^- C^\top + M)^{-1}CA)e_t,
\end{aligned}$$

where the last equation follows from the identity

$$\bar{X}_{ss}C^\top M^{-1} = \bar{X}_{ss}^- C^\top (C\bar{X}_{ss}^- C^\top + M)^{-1}.$$

For the steady-state Kalman filter, it is known that under Assumption 4.4 the PSD matrix $\bar{X}_{ss}^{-}$ solves the filter ARE (4.40). Therefore, the corresponding closed-loop gain matrix $A-\bar{X}_{ss}^{-}C^{\top}(C\bar{X}_{ss}^{-}C^{\top}+M)^{-1}CA$ has eigenvalues strictly within the unit circle, yielding

$$\lim_{t\to\infty} e_t = 0.$$

On the other hand, it follows from (4.66) that

$$\tilde{x}_{t+1} = (A + BK_{ss} + H_{ss})\tilde{x}_t - (BK_{ss} + H_{ss})e_t + BL_{ss} + G_{ss}. \tag{4.67}$$

To show the convergence of $\{\tilde{x}_{t+1}\}$, we rewrite H_{ss} and G_{ss} as

$$\begin{aligned} H_{ss} &= \frac{1}{\lambda}(I + P_{ss}\Phi)^{-1}P_{ss}A \\ G_{ss} &= \frac{1}{\lambda}(I + P_{ss}\Phi)^{-1}(P_{ss}\hat{w} + r_{ss}) + \hat{w}. \end{aligned}$$

Substituting the above expressions and those for K_{ss} and L_{ss} into (4.67), we obtain

$$\tilde{x}_{t+1} = (I+\Phi P_{ss})^{-1}A\tilde{x}_t+\Phi(I+P_{ss}\Phi)^{-1}P_{ss}Ae_t+(I-\Phi(I+P_{ss}\Phi-A^{\top})^{-1}P_{ss})\hat{w}.$$

In the proof of Lemma 4.3, we have shown that $(I + \Phi P_{ss})^{-1}A$ is stable. Thuse, $\{\tilde{x}_t\}$ converges to (4.56) as t tends to infinity. Since $\bar{\tilde{x}}_t = \tilde{x}_t - e_t$, $\{\bar{\tilde{x}}_t\}$ also converges to (4.56).

Moreover, if $\hat{w} = 0$, then $\lim_{t\to\infty} \tilde{x}_t = 0$ and $\lim_{t\to\infty} \bar{\tilde{x}}_t = 0$ as desired. □

Proof of Proposition 4.8

Proof. Consider an adversarial policy $\gamma' \in \Gamma$ that maps the information vector to some distribution with a mean vector $\bar{w}_t$ and a covariance matrix Σ, such that the pair $(A, \Sigma^{1/2})$ is stabilizable. When the policy pair (π^*_{ss}, γ') is applied to the mean-state system, the error state defined in the proof of Proposition 4.7 has the following form:

$$e_{t+1} = (A - \bar{X}^{-}_{ss,\gamma'}C^{\top}(C\bar{X}^{-}_{ss,\gamma'}C^{\top} + M)^{-1}CA)e_t,$$

where $\bar{X}^{-}_{ss,\gamma'}$ is the solution to the filter ARE (4.40) with disturbance distribution $\mathbb{P}_t = \gamma'(I_t)$. Analogous to the proof of Proposition 4.7, the error state e_t converges to the

origin regardless of the control gain matrix K_{ss} since $(A - \bar{X}^{-}_{ss,\gamma'}C^\top(C\bar{X}^{-}_{ss,\gamma'}C^\top + M)^{-1}CA)$ has eigenvalues strictly within the unit circle. The expected value of the state estimate for the mean-state system can now be written as

$$\bar{\bar{x}}_{t+1} = \tilde{A}\bar{\bar{x}}_t + BL_{ss} + \mathbb{E}[w_t] + \bar{X}_{ss,\gamma'}C^\top M^{-1}CAe_t, \tag{4.68}$$

where $\tilde{A} := A + BK_{ss}$ is the closed-loop gain matrix and $\bar{X}_{ss,\gamma'}$ is the conditional state covariance matrix under the adversary's policy γ'. When viewing the disturbances w_t as input, the above system is BIBO stable as long as $\mathbb{E}[w_t]$ is bounded and the matrix $\tilde{A}$ has eigenvalues strictly within the unit circle. Therefore, it is sufficient to show that for the system

$$\bar{\bar{x}}_{t+1} = \tilde{A}\bar{\bar{x}}_t$$

with an arbitrary initial state $\bar{\bar{x}}_0$, the expected value of the estimated state converges to the origin, i.e, $\bar{\bar{x}}_t \to 0$ as $t \to \infty$.

Using the closed-loop system matrix $\tilde{A}$, the ARE (4.30) is equivalent to

$$P_{ss} = Q + \tilde{A}^\top P_{ss}\tilde{A} + K_{ss}^\top R K_{ss} + \tilde{A}^\top P_{ss}(\lambda I - P_{ss})^{-1}P_{ss}\tilde{A}.$$

Therefore, we have

$$\begin{aligned}
\bar{\bar{x}}_{t+1}^\top P_{ss}\bar{\bar{x}}_{t+1} - \bar{\bar{x}}_t^\top P_{ss}\bar{\bar{x}}_t &= \bar{\bar{x}}_t^\top(\tilde{A}^\top P_{ss}\tilde{A} - P_{ss})\bar{\bar{x}}_t \\
&= -\bar{\bar{x}}_t^\top(Q + K_{ss}^\top R K_{ss} + \tilde{A}^\top P_{ss}(\lambda I - P_{ss})^{-1}P_{ss}\tilde{A})\bar{\bar{x}}_t \\
&\leq 0,
\end{aligned}$$

where the last inequality follows from $Q \succeq 0$, $R \succ 0$ and $(\lambda I - P_{ss})^{-1} \succ 0$ under Assumption 4.1. We also deduce that

$$\bar{\bar{x}}_{t+1}^\top P_{ss}\bar{\bar{x}}_{t+1} = \bar{\bar{x}}_0^\top P\bar{\bar{x}}_0 - \sum_{k=0}^{t}\bar{\bar{x}}_k^\top(Q + K_{ss}^\top R K_{ss} + \tilde{A}^\top P_{ss}(\lambda I - P_{ss})^{-1}P_{ss}\tilde{A})\bar{\bar{x}}_k.$$

However, as $P_{ss} \succeq 0$, the left-hand side of the above inequality is no less than zero. Since we have already shown that $\bar{\bar{x}}_t^\top(Q + K_{ss}^\top R K_{ss} + \tilde{A}^\top P_{ss}(\lambda I - P_{ss})^{-1}P_{ss}\tilde{A})\bar{\bar{x}}_t \geq 0$ for each t,

$$\lim_{t\to\infty}\bar{\bar{x}}_t^\top(Q + K_{ss}^\top R K_{ss} + \tilde{A}^\top P_{ss}(\lambda I - P_{ss})^{-1}P_{ss}\tilde{A})\bar{\bar{x}}_t = 0.$$

This implies that

$$\lim_{t\to\infty} Q^{1/2}\bar{\bar{x}}_t = 0, \quad \lim_{t\to\infty} K_{ss}\bar{\bar{x}}_t = 0. \tag{4.69}$$

Recall that $(A, Q^{1/2})$ is observable under Assumption 4.3. Furthermore, the relation $\bar{\bar{x}}_{t+1} = (A + BK_{ss})\bar{\bar{x}}_t$ yields

$$\begin{bmatrix} Q^{1/2}(\bar{\bar{x}}_{t+n_x-1} - \sum_{i=1}^{n_x-1} A^{i-1}BK_{ss}\bar{\bar{x}}_{t+n_x-i-1}) \\ Q^{1/2}(\bar{\bar{x}}_{t+n_x-2} - \sum_{i=1}^{n_x-2} A^{i-1}BK_{ss}\bar{\bar{x}}_{t+n_x-i-2}) \\ \vdots \\ Q^{1/2}(\bar{\bar{x}}_{t+1} - BK_{ss}\bar{\bar{x}}_t) \\ Q^{1/2}\bar{\bar{x}}_t \end{bmatrix} = \begin{bmatrix} Q^{1/2}A^{n_x-1} \\ Q^{1/2}A^{n_x-2} \\ \vdots \\ Q^{1/2}A \\ Q^{1/2} \end{bmatrix} \bar{\bar{x}}_t.$$

From (4.69) the left-hand side tends to zero and hence the right-hand side also tends to zero. However, by the observability assumption the matrix on the right-hand side has full rank, implying that $\bar{\bar{x}}_t \to 0$. Therefore, the eigenvalues of $\tilde{A}$ lie strictly within the unit circle, and the system (4.68) is BIBO stable. Since $\tilde{x}_t = e_t - \bar{\bar{x}}_t$ and $\mathbb{E}[y_t] = C\tilde{x}_t$, we conclude that the mean-state system is also BIBO stable. □

Chapter 5

Distributionally Robust Differential Dynamic Programming with Wasserstein Distance

5.1 Introduction

Nonlinear optimal control problems are difficult to solve exactly, particularly when the state space dimension is high. Differential dynamic programming (DDP) alleviates this issue using locally-quadratic approximations of the system dynamics and cost function [191–196]. It efficiently computes an approximate solution with superior scalability compared to the standard dynamic programming (DP) approach. However, it is generally challenging to apply DDP to systems with random disturbances without any means to counteract them.

Although various works have extended DDP to handle stochastic systems, existing methods often rely on either the ground truth or potentially inaccurate approximate probability distributions of disturbances. For example, the DDP algorithms introduced in [197–200] either consider Gaussian multiplicative noise or model the uncertain system dynamics as Gaussian processes. Another line of research is devoted to the minimax formulation of the DDP problem (e.g., [201, 202]), where the optimal control problem is solved in the face of the worst-case disturbances. However, such methods

often lead to overly conservative solutions.

To address the limitations of stochastic DDP methods and handle systems with unknown disturbance distributions, we propose a novel approach inspired by distributionally robust control (DRC). The objective of DRC is to design control policies that maximize the worst-case performance over a set of candidate distributions without assuming a specific distribution of disturbances. Several techniques have been proposed for hedging against distributional uncertainties in DRC problems, including moment-based and statistical distance-based approaches [31, 33, 174, 177, 179, 181, 182, 203]. While moment-based approaches rely on accurate moment estimates and may not effectively capture the full distributional information about the uncertainties, distance-based methods consider distributions that are close to a given nominal one in terms of a statistical distance measure. Many recent works have focused on Wasserstein DRC (WDRC) [30, 42, 204–206], where the ambiguity set is designed as a statistical ball with the distance between two distributions measured by the Wasserstein metric. The Wasserstein ambiguity set has salient features, including a finite-sample performance guarantee and the ability to avoid pathological solutions to distributionally robust optimization (DRO) problems [39, 40, 43].

Despite numerous attempts, existing WDRC methods still face challenges in terms of tractability and scalability. For instance, the DP-based approach introduced in [30] for solving the WDRC problem results in a semi-infinite program, requiring computationally expensive state-space discretization or sampling. To overcome this limitation, both [30] and [42] propose a relaxation technique with a penalty on the Wasserstein distance, which leads to an explicit solution in the linear-quadratic (LQ) setting. While these works focus on the theoretical analysis of the obtained policies, this study aims to design a practical and computationally efficient algorithm for solving the nonlinear WDRC problem.

In particular, a novel DDP method is developed through a locally quadratic approximation of a nonlinear WDRC problem, where the true disturbance distribution

is unknown but a disturbance sample dataset is given. By construction, the proposed distributionally robust DDP (DR-DDP) algorithm provides control policies that are robust against inevitable inaccuracies in empirical distributions of the disturbance. To make the method tractable, we first approximate the WDRC problem with its penalty version and then apply the Kantorovich duality principle. We show that the proposed approximation provides a suboptimal solution to the original WDRC problem. The value function is then decomposed in a novel way that enables deriving computationally tractable and efficient backward and forward passes. This allows us to obtain closed-form expressions for the distributionally robust control and worst-case distribution policies in each iteration of the DR-DDP algorithm. By avoiding the need to numerically solve minimax optimization problems, our approach makes the algorithm not only tractable but also scalable. The scalability of our DDP method is a remarkable advantage because the computational complexity of the standard DP algorithm in [30] for nonlinear WDRC increases exponentially with the dimension of the state space. The experiment results on kinematic car navigation and coupled oscillator problems indicate that our algorithm outperforms existing methods in terms of out-of-sample performance and provides scalable solutions for high-dimensional nonlinear optimal control problems.

5.2 Preliminaries

In this section, we introduce the WDRC problem used in our development of the DR-DDP algorithm in Section 5.3.

5.2.1 Distributionally Robust Control

Consider the following discrete-time stochastic system:

$$x_{t+1} = f(x_t, u_t, w_t), \tag{5.1}$$

where $x_t \in \mathbb{R}^{n_x}$ and $u_t \in \mathbb{R}^{n_u}$ are the system states and control inputs, respectively. Here, $w_t \in \mathbb{R}^{n_w}$ is a random disturbance with an unknown (true) distribution $\mathbb{Q}_t^{\text{true}} \in \mathcal{P}(\mathbb{R}^{n_w})$, where $\mathcal{P}(\mathbb{R}^{n_w})$ is the family of all Borel probability measures supported on $\mathbb{R}^{n_w}$. The nonlinear function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x}$ is assumed to be twice continuously differentiable.

In practice, it is restrictive to assume that the true probability distribution $\mathbb{Q}_t^{\text{true}}$ is known. Instead, we are often given a sample dataset $\mathcal{D}_t := \{\hat{w}_t^{(1)}, \hat{w}_t^{(2)}, \ldots, \hat{w}_t^{(N)}\}$ drawn from the true distribution, which can be used to construct an empirical estimate about the distribution of w_t as

$$\mathbb{Q}_t := \frac{1}{N} \sum_{i=1}^{N} \delta_{\hat{w}_t^{(i)}},$$

where $\delta_{\hat{w}_t^{(i)}}$ denotes the Dirac measure concentrated at $\hat{w}_t^{(i)}$. It is well-known that as $N \to \infty$, the empirical distribution asymptotically converges to the true distribution. However, if an inaccurate empirical estimate is used in the controller design, the resulting control performance will deteriorate due to a mismatch between the true and empirical distributions.

To hedge against such distributional uncertainties, we adopt a game-theoretic approach and consider a two-player zero-sum game in which Player I is the controller and Player II is a hypothetical adversary. Let $\pi := (\pi_0, \ldots, \pi_{T-1})$ denote the control policy, where π_t maps the state x_t to a control input u_t. The adversary player selects a policy $\gamma := (\gamma_0, \ldots, \gamma_{T-1})$, where γ_t maps the current state to a probability distribution $\mathbb{P}_t$ chosen from an *ambiguity set* $\mathbb{D}_t \subset \mathcal{P}(\mathbb{R}^{n_w})$. The ambiguity set is a family of distributions that possess certain properties to be described.

Throughout this paper, our goal is to design an optimal finite-horizon controller with the following cost functional:

$$J(\pi, \gamma) := \mathbb{E}^{\pi,\gamma}\big[\ell_f(x_T) + \sum_{t=0}^{T-1} \ell(x_t, u_t)\big],$$

where $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and $\ell_f : \mathbb{R}^{n_x} \to \mathbb{R}$ are the twice continuously differentiable

running and terminal costs, respectively, and T is the time horizon. In our problem, the controller seeks a policy π^* minimizing the cost function, while the adversary aims to find a policy γ^* to maximize the same cost, which can be obtained by solving the following DRC problem:

$$\min_{\pi\in\Pi}\max_{\gamma\in\Gamma_{\mathbb{D}}} J(\pi,\gamma), \tag{5.2}$$

where $\Pi := \{\pi \mid \pi_t(x_t) = u_t \in \mathbb{R}^{n_u},\ \forall t\}$ and $\Gamma_{\mathbb{D}} := \{\gamma \mid \gamma_t(x_t) = \mathbb{P}_t \in \mathbb{D}_t,\ \forall t\}$ are the sets of admissible control and distribution policies, respectively.

5.2.2 Wasserstein Ambiguity Set

In problem (5.2), the adversary player is restricted to select a distribution from the ambiguity set $\mathbb{D}_t$, which determines the characteristics of the worst-case distribution. Therefore, it is necessary to design the ambiguity set to appropriately characterize distributional errors. Motivated by its advantages mentioned in Section 5.1, we use the Wasserstein ambiguity set constructed around the given empirical distribution. The Wasserstein metric of order p between two distributions $\mathbb{P}$ and $\mathbb{Q}$ supported on $\mathcal{W} \subseteq \mathbb{R}^n$ represents the minimum cost of redistributing mass from one distribution to another using a small non-uniform perturbation and is defined as

$$W_p(\mathbb{P},\mathbb{Q}) := \inf_{\tau\in\mathcal{P}(\mathcal{W}^2)} \left\{ \Big(\int_{\mathcal{W}^2} \|x-y\|^p \,\mathrm{d}\tau(x,y)\Big)^{1/p} \Big|\, \Pi^1\tau = \mathbb{P}, \Pi^2\tau = \mathbb{Q} \right\},$$

where τ is the *transport plan* with $\Pi^i\tau$ denoting its ith marginal distribution, and $\|\cdot\|$ is a norm on $\mathbb{R}^n$ which quantifies the transportation cost.

In this work, we consider the Wasserstein metric of order $p = 2$ with the transportation cost represented by the standard Euclidean norm. We design the ambiguity set as follows:

$$\mathbb{D}_t := \{\mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_w}) \mid W_2(\mathbb{P}_t, \mathbb{Q}_t) \le \theta\}, \tag{5.3}$$

where $\theta > 0$ determines the size of $\mathbb{D}_t$. The ambiguity set (5.3) is a statistical ball centered at the empirical distribution $\mathbb{Q}_t$ and contains all distributions whose Wasserstein distance from the empirical distribution is no greater than radius θ.

5.3 Distributionally Robust Differential Dynamic Programming

In this section, we present our main result, called DR-DDP, which efficiently finds an approximate solution to the WDRC problem. Our method exploits the Kantorovich duality principle to decompose the value function in a novel way and devise a computationally tractable algorithm.

5.3.1 Approximation with Wasserstein Penalty

In [42], the tractability and effectiveness of a penalty version of the WDRC problem are studied. Motivated by this work, we begin our reformulations by replacing the Wasserstein ambiguity set constraint with a penalty term in the cost function as follows:

$$J_\lambda(\pi,\gamma) := \mathbb{E}^{\pi,\gamma}\Big[\ell_f(x_T) + \sum_{t=0}^{T-1} \ell(x_t,u_t) - \lambda W_2(\mathbb{P}_t,\mathbb{Q}_t)^2\Big],$$

where $\lambda > 0$ is the penalty parameter adjusting the conservativeness of the controller.

Then, the following minimax control problem approximates the original WDRC problem (5.2):

$$\min_{\pi\in\Pi}\max_{\gamma\in\Gamma} J_\lambda(\pi,\gamma), \tag{5.4}$$

where the adversary player selects policies from $\Gamma := \{\gamma := (\gamma_0,\ldots,\gamma_{T-1}) \mid \gamma_t(x_t) = \mathbb{P}_t \in \mathcal{P}(\mathbb{R}^{n_w})\}$. Note that the adversary is not restricted to select distributions from the ambiguity set. Instead, we penalize large deviations from the empirical distribution via the penalty term, thus limiting the freedom of the adversary player.

We demonstrate in the following proposition that the cost incurred by an arbitrary policy $\pi \in \Pi$ under the worst-case distributions within the Wasserstein ambiguity set has a guaranteed cost property with respect to the worst-case penalized cost. Hence, the penalty problem (5.4) is a reasonable approximation as it yields a suboptimal solution to the WDRC problem (5.2).

Proposition 5.1. *Given $\lambda > 0$, let $\pi \in \Pi$ be any arbitrary policy. Then, the cost incurred by π under the worst-case distribution policy in $\Gamma_{\mathbb{D}}$ is upper-bounded as follows:*

$$\sup_{\gamma \in \Gamma_{\mathbb{D}}} J(\pi, \gamma) \leq \lambda T \theta^2 + \sup_{\gamma \in \Gamma} J_\lambda(\pi, \gamma). \tag{5.5}$$

Its proof can be found in Appendix 5.6.1. The guaranteed cost property indicates the role of the penalty parameter λ in adjusting the robustness of the control policy, thereby providing a guideline on its selection. Specifically, the penalty parameter can be chosen to yield the least upper bound in (5.5) under the given control policy.[14]

To formalize our algorithm, we recursively define the optimal value function for problem (5.4) as follows:

$$V_t(\boldsymbol{x}) := \inf_{\pi \in \Pi} \sup_{\gamma \in \Gamma} \mathbb{E}^{\pi,\gamma} \left[\ell_f(x_T) + \sum_{s=t}^{T-1} \ell(x_s, u_s) - \lambda W_2(\mathbb{P}_s, \mathbb{Q}_s)^2 \mid x_t = \boldsymbol{x} \right]$$

for $t = T-1, \ldots, 0$, with the terminal condition $V_T(\boldsymbol{x}) = \ell_f(\boldsymbol{x})$. Then, the DP principle yields

$$V_t(\boldsymbol{x}) = \inf_{\boldsymbol{u} \in \mathbb{R}^{n_u}} \sup_{\mathbb{P} \in \mathcal{P}(\mathbb{R}^{n_w})} \ell(\boldsymbol{x}, \boldsymbol{u}) + \mathbb{E}^{w \sim \mathbb{P}} \left[V_{t+1}(f(\boldsymbol{x}, \boldsymbol{u}, w)) - \lambda W_2(\mathbb{P}, \mathbb{Q}_t)^2 \right] \tag{5.6}$$

with the optimal cost given by

$$J_\lambda^* := \inf_{\pi \in \Pi} \sup_{\gamma \in \Gamma} J_\lambda(\pi, \gamma) = V_0(x_0).$$

Unfortunately, the standard procedure for DDP cannot be applied to the value function (5.6) as it constitutes an infinite-dimensional optimization problem over $\mathcal{P}(\mathbb{R}^{n_w})$. For tractability, we employ a modern DRO technique based on the Kantorovich duality principle [30, 207] and reformulate the value function as follows.

[14] The value of λ heavily depends on the choice of the Wasserstein ambiguity set radius θ, which is typically chosen to attain a probabilistic out-of-sample performance guarantee, given a finite dataset of disturbance samples (e.g., [39, 43]).

Proposition 5.2. *Suppose that for each* $(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$*, the value function is measurable and that the outer minimization problem in* (5.6) *has an optimal solution. Then, for any* $\lambda > 0$*, we have that*

$$V_t(\boldsymbol{x}) = \inf_{\boldsymbol{u} \in \mathbb{R}^{n_u}} \ell(\boldsymbol{x}, \boldsymbol{u}) + \mathbb{E}^{\hat{w}_t \sim \mathbb{Q}_t} \Big[\sup_{\boldsymbol{w} \in \mathbb{R}^{n_w}} V_{t+1}(f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})) - \lambda \|\hat{w}_t - \boldsymbol{w}\|^2 \Big], \quad (5.7)$$

for all $\boldsymbol{x} \in \mathbb{R}^{n_x}$.

Its proof can be found in Appendix 5.6.2. While previous works (e.g., [30]) use similar approaches to reformulate and analyze the solution to the WDRC problem, our focus is on designing a practical and efficient method for obtaining tractable solutions. For that, we let

$$Q_t^{(i)}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) := \ell(\boldsymbol{x}, \boldsymbol{u}) + V_{t+1}(f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})) - \lambda \|\hat{w}_t^{(i)} - \boldsymbol{w}\|^2$$

denote the state-action-disturbance value function or the Q-function for each sample index $i = 1, \ldots, N$ and

$$Q_t^{*,(i)}(\boldsymbol{x}, \boldsymbol{u}) = \sup_{\boldsymbol{w} \in \mathbb{R}^{n_w}} Q_t^{(i)}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w})$$

denote the corresponding "worst-case" state-action value function. Then, we obtain that

$$V_t(\boldsymbol{x}) = \inf_{\boldsymbol{u} \in \mathbb{R}^{n_u}} \frac{1}{N} \sum_{i=1}^{N} Q_t^{*,(i)}(\boldsymbol{x}, \boldsymbol{u}). \quad (5.8)$$

It is worth emphasizing that the Kantorovich duality principle enables us to obtain this novel decomposition of the value function, which can be used to design a computationally tractable DR-DDP solution in the following subsection.

5.3.2 Solution via DDP

In each iteration of the original DDP algorithm, a backward pass is performed on the current estimate of the state and control trajectories, called the *nominal trajectories*, followed by a forward pass. In the backward pass, the cost function and the system

dynamics are quadratically approximated around the nominal trajectories to update the policy, while in the forward pass, the nominal trajectories are recomputed by executing the latest policy to the system. We adopt this technique for our problem and derive the backward and forward passes for the value function (5.7). The proposed DR-DDP method is presented in Algorithm 5.

Backward Pass

In each backward pass, we are given nominal state, control input, and disturbance trajectories $\bar{\boldsymbol{x}}_{\rm nom} = (\bar{\boldsymbol{x}}_0, \ldots, \bar{\boldsymbol{x}}_T)$, $\bar{\boldsymbol{u}}_{\rm nom} = (\bar{\boldsymbol{u}}_0, \ldots, \bar{\boldsymbol{u}}_{T-1})$ and $\bar{\boldsymbol{w}}_{\rm nom} = (\bar{\boldsymbol{w}}_0, \ldots, \bar{\boldsymbol{w}}_{T-1})$, respectively. For quadratic approximations, DDP considers the following deviations of the system state, control input, and disturbance, i.e., $\delta x_t := x_t - \bar{\boldsymbol{x}}_t$, $\delta u_t := u_t - \bar{\boldsymbol{u}}_t$, $\delta w_t := w_t - \bar{\boldsymbol{w}}_t$.

We first consider the following second-order approximation of $V_{t+1}(x_{t+1})$:

$$\boldsymbol{V}_{t+1} + V_{t+1,x}^\top \delta x_{t+1} + \frac{1}{2} \delta x_{t+1}^\top V_{t+1,xx} \delta x_{t+1}, \tag{5.9}$$

for some $(\boldsymbol{V}_{t+1}, V_{t+1,x}, V_{t+1,xx}) \in \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_x \times n_x}$ to be determined.[15] Let $\hat{Q}_t^{(i)}$ be an approximate Q-function, defined by replacing V_{t+1} in the definition of $Q_t^{(i)}$ with the approximate value function (5.9). Then, $\hat{Q}_t^{(i)}(x_t, u_t, w_t)$ is twice differentiable and its second-order Taylor expansion is given by

$$\boldsymbol{Q}_t^{(i)} + \delta Q_t^{(i)}(\delta x_t, \delta u_t, \delta w_t), \tag{5.10}$$

where

$$\delta Q_t^{(i)}(\delta x_t, \delta u_t, \delta w_t) = Q_{t,x}^\top \delta x_t + Q_{t,u}^\top \delta u_t + {Q_{t,w}^{(i)}}^\top \delta w_t + \frac{1}{2} \Delta Q_t(\delta x_t, \delta u_t, \delta w_t)$$

with

$$\Delta Q_t(\delta x, \delta u, \delta w) := \begin{bmatrix} \delta x \\ \delta u \\ \delta w \end{bmatrix}^\top \begin{bmatrix} Q_{t,xx} & Q_{t,xu} & Q_{t,xw} \\ Q_{t,xu}^\top & Q_{t,uu} & Q_{t,uw} \\ Q_{t,xw}^\top & Q_{t,uw}^\top & Q_{t,ww} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \\ \delta w \end{bmatrix}$$

[15] If V_{t+1} is twice differentiable, the parameters $(\boldsymbol{V}_{t+1}, V_{t+1,x}, V_{t+1,xx})$ can be simply determined using the second-order Taylor expansion.

and

$$\begin{cases} \boldsymbol{Q}_t^{(i)} = \mathcal{J}(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{u}}_t) + \boldsymbol{V}_{t+1} - \lambda \|\bar{\boldsymbol{w}}_t - \hat{w}_t^{(i)}\|^2 \\ Q_{t,xx} = \ell_{t,xx} + f_{t,x}^\top V_{t+1,xx} f_{t,x} + V_{t+1,x}^\top f_{t,xx} \\ Q_{t,uu} = \ell_{t,uu} + f_{t,u}^\top V_{t+1,xx} f_{t,u} + V_{t+1,x}^\top f_{t,uu} \\ Q_{t,ww} = f_{t,w}^\top V_{t+1,xx} f_{t,w} - 2\lambda I + V_{t+1,x}^\top f_{t,ww} \\ Q_{t,xu} = \ell_{t,xu} + f_{t,x}^\top V_{t+1,xx} f_{t,u} \\ Q_{t,xw} = f_{t,x}^\top V_{t+1,xx} f_{t,w}, \quad Q_{t,uw} = f_{t,u}^\top V_{t+1,xx} f_{t,w} \\ Q_{t,x} = \ell_{t,x} + f_{t,x}^\top V_{t+1,x}, \quad Q_{t,u} = \ell_{t,u} + f_{t,u}^\top V_{t+1,x} \\ Q_{t,w}^{(i)} = f_{t,w}^\top V_{t+1,x} - 2\lambda(\bar{\boldsymbol{w}}_t - \hat{w}_t^{(i)}). \end{cases}$$

Here, $f_{t,\cdot}$ and $\ell_{t,\cdot}$ denote the partial derivatives of f and ℓ evaluated at $(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{u}}_t, \bar{\boldsymbol{w}}_t)$.

Let $\hat{\bar{w}}_t := \mathbb{E}^{\hat{w}_t \sim \mathbb{Q}_t}[\hat{w}_t]$ and $\hat{\Sigma}_t := \mathbb{E}^{\hat{w}_t \sim \mathbb{Q}_t}[(\hat{w}_t - \hat{\bar{w}}_t)(\hat{w}_t - \hat{\bar{w}}_t)^\top]$ denote the empirical mean vector and covariance matrix of disturbance w_t, respectively. The above approximation transforms the problem (5.8) into a quadratic form similar to those addressed in [30,42]. This approximation enables us to explicitly solve the problem with respect to δu_t and δw_t, as presented in the following theorem.

Theorem 5.1. *Let $Q_{t,ww} \prec 0$ and $\ell_{t,uu} \succ 0$. Suppose the value function at time $t+1$ is approximated as* (5.9)*. Then, the outer minimization problem in* (5.8) *with $Q_t^{(i)}(x_t, u_t, w_t)$ replaced by the approximation* (5.10) *has the following unique minimizer:*

$$\delta u_t^* = K_t \delta x_t + k_t, \tag{5.11}$$

where

$$\begin{aligned} K_t &= -\tilde{Q}_t (Q_{t,xu}^\top - Q_{t,uw} Q_{t,ww}^{-1} Q_{t,xw}^\top) \\ k_t &= -\tilde{Q}_t (Q_{t,u} - Q_{t,uw} Q_{t,ww}^{-1} \bar{Q}_{t,w}) \end{aligned} \tag{5.12}$$

with $\tilde{Q}_t := (Q_{t,uu} - Q_{t,uw} Q_{t,ww}^{-1} Q_{t,uw}^\top)^{-1}$ and $\bar{Q}_{t,w} := f_{t,w}^\top V_{t+1,x} - 2\lambda(\bar{\boldsymbol{w}}_t - \hat{\bar{w}}_t)$.

Moreover, for each $i = 1, \ldots, N$, the maximization problem in (5.8) *with $Q_t^{(i)}(x_t, u_t, w_t)$ replaced by the approximation* (5.10) *has the following unique solution:*

$$\delta w_t^{*,(i)} = H_t \delta x_t + h_t^{(i)}, \tag{5.13}$$

where

$$\begin{aligned} H_t &= -Q_{t,ww}^{-1}[Q_{t,uw}^\top K_t + Q_{t,xw}^\top] \\ h_t^{(i)} &= -Q_{t,ww}^{-1}[Q_{t,uw}^\top k_t + Q_{t,w}^{(i)}]. \end{aligned} \tag{5.14}$$

Proof. Let $\delta w_t^{(i)} := w_t^{(i)} - \bar{\boldsymbol{w}}_t$. Evaluating the approximate Q-function (5.10) for $\delta w_t^{(i)}$, we see that it is strictly concave in $\delta w_t^{(i)}$ as $Q_{t,ww} \prec 0$. Then, the first-order optimality condition yields the following unique maximizer:

$$\delta w_t^{*,(i)} = -Q_{t,ww}^{-1}(Q_{t,xw}^\top \delta x_t + Q_{t,uw}^\top \delta u_t + Q_{t,w}^{(i)}). \tag{5.15}$$

Replacing $Q_t^{(i)}(x_t, u_t, w_t)$ with the approximation (5.10), the objective function in (5.8) is quadratically approximated as

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^N \left[\boldsymbol{Q}_t^{(i)} + \delta Q_t^{(i)}(\delta x_t, \delta u_t, \delta w_t^{*,(i)})\right] = \bar{\boldsymbol{Q}}_t + Q_{t,x}^\top \delta x_t + Q_{t,u}^\top \delta u_t + \bar{Q}_{t,w}^\top \overline{\delta w_t}^* \\ + \frac{1}{2}\Delta Q_t(\delta x_t, \delta u_t, \overline{\delta w_t}^*), \end{aligned}$$

where

$$\overline{\delta w_t}^* := \frac{1}{N}\sum_{i=1}^N \delta w_t^{*,(i)} = -Q_{t,ww}^{-1}(Q_{t,xw}^\top \delta x_t + Q_{t,uw}^\top \delta u_t + \bar{Q}_{t,w})$$

and

$$\bar{\boldsymbol{Q}}_t := \ell_t(\bar{\boldsymbol{x}}_t, \bar{\boldsymbol{u}}_t) + \boldsymbol{V}_{t+1} - \lambda\|\bar{\boldsymbol{w}}_t - \hat{\bar{w}}_t\|^2 - \lambda \mathrm{Tr}[\hat{\Sigma}_t] - 2\lambda^2 \mathrm{Tr}[Q_{t,ww}^{-1}\hat{\Sigma}_t].$$

To minimize this approximated objective function with respect to δu_t, the following first-order optimality condition can be used:

$$\begin{aligned} 0 = Q_{t,u} + Q_{t,uu}\delta u_t + Q_{t,xu}^\top \delta x_t + Q_{t,uw}\overline{\delta w_t}^* \\ + \frac{\partial \overline{\delta w_t}^*}{\partial \delta u_t}^\top (\bar{Q}_{t,w} + Q_{t,xw}^\top \delta x_t + Q_{t,uw}^\top \delta u_t + Q_{t,ww}\overline{\delta w_t}^*). \end{aligned}$$

By the strong convexity of the quadratic approximation, its minimizer is uniquely given by

$$\delta u_t^* = -\tilde{Q}_t\big(Q_{t,u} - Q_{t,uw}Q_{t,ww}^{-1}\bar{Q}_{t,w} + [Q_{t,xu}^\top - Q_{t,uw}Q_{t,ww}^{-1}Q_{t,xw}^\top]\delta x_t\big),$$

which is equivalent to (5.11). By substituting δu_t^* into (5.15), we obtain the maximizer defined in (5.13). □

Theorem 5.1 provides the remarkable advantage that a DR-DDP policy pair $(\bar{\pi}^*, \bar{\gamma}^*)$ is constructed in the following closed-form without numerically solving any infinite-dimensional minimax optimization problems:

$$\bar{\pi}_t^*(x_t) = \bar{\boldsymbol{u}}_t + K_t(x_t - \bar{\boldsymbol{x}}_t) + k_t \tag{5.16a}$$

$$\bar{\gamma}_t^*(x_t) = \frac{1}{N}\sum_{i=1}^{N} \delta_{(\bar{\boldsymbol{w}}_t + h_t^{(i)} + H_t(x_t - \bar{\boldsymbol{x}}_t))}. \tag{5.16b}$$

As a result of the backward pass, we also obtain the following equations for updating the parameters of the approximate value function (5.9):

$$\begin{aligned}
\boldsymbol{V}_t =& \bar{\boldsymbol{Q}}_t + Q_{t,u}^\top k_t + \bar{Q}_{t,w}^\top h_t + \frac{1}{2} k_t^\top Q_{t,uu} k_t + \frac{1}{2} h_t^\top Q_{t,ww} h_t + k_t^\top Q_{t,uw} h_t \\
V_{t,x} =& Q_{t,x} + Q_{t,xu} k_t + K_t^\top (Q_{t,u} + Q_{t,uu} k_t + Q_{uw} h_t) \\
&+ Q_{xw} h_t + H_t^\top (\bar{Q}_{t,w} + Q_{t,ww} h_t + Q_{t,uw}^\top k_t) \\
V_{t,xx} =& Q_{t,xx} + K_t^\top Q_{t,uu} K_t + H_t^\top Q_{t,ww} H_t + 2 Q_{t,xu} K_t \\
&+ 2 K_t^\top Q_{t,uw} H_t + 2 Q_{t,xw} H_t,
\end{aligned} \tag{5.17}$$

where $h_t := \frac{1}{N}\sum_{i=1}^{N} h_t^{(i)}$.

In practice, it is not common to assume that control inputs are unrestricted. Often, the control inputs are limited to some box constraints $\overline{\mathbf{u}} \leq u_t \leq \underline{\mathbf{u}}$. Taking into account such control limits requires a careful design of the backward pass as it is required to minimize the approximate state-action value function subject to the constraints. To find a closed-form solution to the constrained problem for all δx_t, we use the projected Newton-based approach proposed in [192], where the control gains are found by solving a quadratic program.

In the next step, the nominal trajectories have to be reconstructed using the DR-DDP policy pair $(\bar{\pi}^*, \bar{\gamma}^*)$ to update the quadratically approximated models, which is performed during the forward pass introduced in what follows.

Forward Pass

In the original DDP algorithm, the forward pass is performed by executing the control policy to the system. However, due to the disturbance term in the system dynamics and lack of knowledge about its true distribution, it is not trivial to perform forward rollouts for the ambiguous stochastic system (5.1). Instead, we choose to execute the control and distribution policy pair $(\bar{\pi}^*, \bar{\gamma}^*)$ in the following manner.

First, using (5.16a) and (5.16b), we construct a control input $u_t = \bar{\boldsymbol{u}}_t + \alpha k_t + K_t(x_t - \bar{\boldsymbol{x}}_t)$ and sample a disturbance realization as $w_t \sim \frac{1}{N}\sum_{i=1}^{N} \delta_{\bar{\boldsymbol{w}}_t + \alpha h_t^{(i)} + H_t(x_t - \bar{\boldsymbol{x}}_t)}$, where $\alpha \in (0, 1)$ is a line-search parameter. Since DDP is a second-order method and potentially takes large steps, regularization is required to prevent the blow-up of the value. Therefore, we multiply k_t and $h_t^{(i)}$ by scaling a parameter $\alpha \in (0, 1)$ and perform a line-search. In particular, the line-search parameter alpha is iteratively reduced to improve the total cost. Then, both the control input and the disturbance sample are executed to the system for $t = 0, \ldots, T-1$ starting from the initial state x_0.[16]

5.4 Numerical Experiments

In this section, we compare the empirical performance of our DR-DDP method with three baseline algorithms: *GT-DDP* [201], which uses a minimax approach to consider the worst-case disturbances, *box-DDP* [192], a deterministic DDP algorithm that ignores uncertainties in the controller design but considers box constraints on control inputs, and *NR-DDP*, the non-robust version of our DR-DDP algorithm that utilizes the empirical distribution.

In our experiments, we choose the penalty parameter λ to minimize the cost upper bound (5.5) for $\theta = 0.1$ under the DR-DDP policy pair $(\bar{\pi}^*, \bar{\gamma}^*)$. We estimate the upper

[16] The complexity of a single iteration of our algorithm is bounded by $O\big(T(n_x^3 + n_u^3 + (N + n_w)n_w^2)\big)$, which is polynomial in state, input and disturbance dimensions and linear in the time horizon and sample size.

Algorithm 5: DR-DDP algorithm

Input: $x_0, \pi_{\text{init}}, \gamma_{\text{init}}, T, \lambda$

Apply $(\pi_{\text{init}}, \gamma_{\text{init}})$ to generate $(\bar{\boldsymbol{x}}_{\text{nom}}, \bar{\boldsymbol{u}}_{\text{nom}}, \bar{\boldsymbol{w}}_{\text{nom}})$

while *not converged* **do**

`// Backward Pass`

$\boldsymbol{V}_T \leftarrow \ell_f(\bar{\boldsymbol{x}}_T), V_{T,x} \leftarrow \ell_{f,x}, V_{T,xx} \leftarrow \ell_{f,xx}$

for $t = T-1$ **to** 0 **do**

Construct $(\bar{\pi}_t^*, \bar{\gamma}_t^*)$ using (5.16a) and (5.16b)

Update $\boldsymbol{V}_t, V_{t,x}, V_{t,xx}$ according to (5.17)

`// Forward Pass`

Perform line-search to update α

for $t = 0$ **to** $T-1$ **do**

Compute $u_t = \bar{\boldsymbol{u}}_t + \alpha k_t + K_t(x_t - \bar{\boldsymbol{x}}_t)$

Sample $w_t \sim \frac{1}{N}\sum_{i=1}^{N} \delta_{\bar{\boldsymbol{w}}_t + \alpha h_t^{(i)} + H_t(x_t - \bar{\boldsymbol{x}}_t)}$

Execute u_t and w_t to (5.1) and observe x_{t+1}

$\bar{\boldsymbol{x}}_{\text{nom}} \leftarrow x_{0:T}, \bar{\boldsymbol{u}}_{\text{nom}} \leftarrow u_{0:T-1}, \bar{\boldsymbol{w}}_{\text{nom}} \leftarrow w_{0:T-1}$

return $(\bar{\pi}^*, \bar{\gamma}^*)$

bound by conducting 1,000 independent Monte Carlo simulations and computing the Wasserstein distance via a linear program. The optimal penalty parameter is then found via numerical optimization. Note that this procedure does not require knowledge of the true disturbance distribution.

All simulations were performed on a PC with a 3.70 GHz Intel Core i7-8700K processor and 32 GB RAM. The source code of our DR-DDP implementation is available online.[17]

[17] `https://github.com/CORE-SNU/DR-DDP`

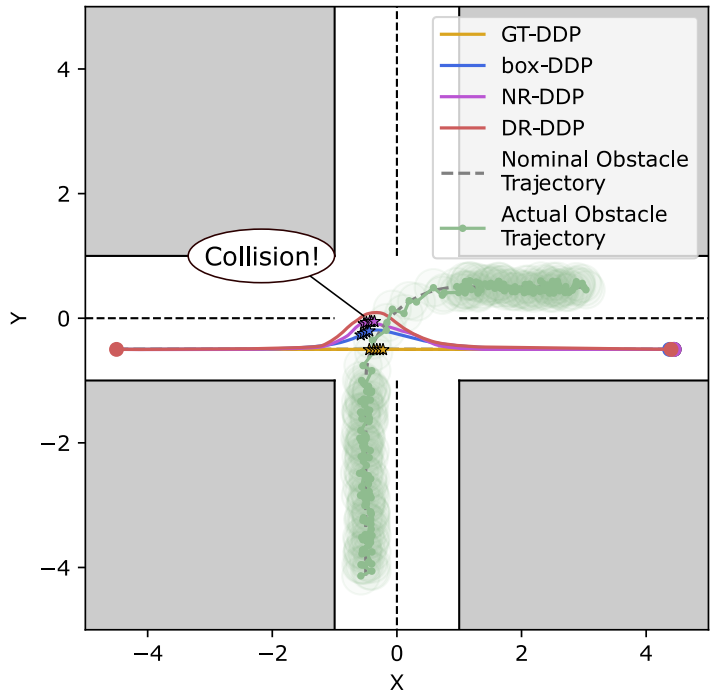


Figure 5.1: Trajectories of the kinematic car, controlled by GT-DDP, box-DDP, NR-DDP, and DR-DDP, in the presence of a randomly moving obstacle. Star marks represent collisions.

5.4.1 Kinematic Car Navigation

In the first experiment, we consider an autonomous navigation task for a kinematic car in an intersection where a randomly moving obstacle obstructs navigation. Consider the following system:

$$x_{t+1} = \begin{bmatrix} x_{t+1}^{\text{car}} \\ \boldsymbol{p}_{t+1}^{\text{obs}} \end{bmatrix} = \begin{bmatrix} f_{\text{car}}(x_t^{\text{car}}, u_t) \\ \boldsymbol{p}_t^{\text{obs}} + \Delta \boldsymbol{p}_t^{\text{obs}} + w_t \end{bmatrix}$$

with system state $x_t \in \mathbb{R}^5$ and control input $u_t \in \mathbb{R}^2$. Here, $x_t^{\text{car}} \in \mathbb{R}^3$ represents the car's state evolving according to the differential-drive kinematics $f_{\text{car}} : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ and consists of the car's center position $\boldsymbol{p}$ and its heading angle ϕ. The control input vector comprises the velocity and steering angle of the car and has a lower limit of $\underline{\mathbf{u}} = [0, -0.6]^\top$ and an upper limit of $\overline{\mathbf{u}} = [10, 0.6]^\top$. The state component $\boldsymbol{p}_t^{\text{obs}}$ represents

the position vector of a random circular obstacle with radius $r_{\text{obs}} = 0.2$. It is assumed that in each time instance, the obstacle has a nominal deterministic motion represented by $\Delta \boldsymbol{p}_t^{\text{obs}} \in \mathbb{R}^2$, which is obstructed with a positional disturbance vector $w_t \in \mathbb{R}^2$. Each component of the disturbances follows a uniform distribution $\mathcal{U}(-0.1, 0.1)$. Our DR-DDP algorithm uses only $N = 10$ samples drawn from the true distribution. The goal is to safely pass the intersection by tracking the given reference trajectory x^{ref} and avoiding the obstacle in $T = 800$ steps. For this purpose, we design a time-varying cost function as

$$\ell_t(x, u) := \|x^{\text{car}} - x_t^{\text{ref}}\|_Q^2 + \|u\|_R^2 + Q_{\text{obs}} \exp\left(-0.5 \frac{\|\boldsymbol{p} - \boldsymbol{p}^{\text{obs}}\|^2}{(r_{\text{obs}} + r_{\text{safe}})^2}\right),$$

where the last term is a soft constraint for avoiding the obstacle with a safe margin of $r_{\text{safe}} = 0.2$. The weights are chosen as $Q = 10I, R = 0.1I$ and $Q_{\text{obs}} = 20$. The terminal cost is similar to the running cost with no control cost. The penalty parameter is set to $\lambda = 9000$ found as the minimizer of the upper bound in (5.5).

Fig. 5.1 shows the trajectories of the kinematic car for a single realization of the disturbances, while Table 5.1 summarizes the computational requirements of each algorithm. Only DR-DDP successfully avoids the obstacle and accomplishes the task, resulting in the lowest total cost. Even though both box-DDP and NR-DDP drive the car away from the reference path, they collide with the obstacle, leading to increased total costs due to the soft constraint for collision avoidance. This is because box-DDP completely disregards uncertainties, while NR-DDP relies solely on inaccurate disturbance information. Meanwhile, GT-DDP incurs extremely high costs as it fails to drive the car away from the obstacle. Despite the distinct behaviors exhibited by the two algorithms, the average total computation times for DR-DDP and GT-DDP are quite similar (less than $25\,sec.$), indicating their comparable computational efficiency. To validate our results, we conducted 1,000 independent simulation runs to measure the *out-of-sample performance* of each method, which are reported in Table 5.1.[18] The proposed

[18]The out-of-sample performance of the controller is defined as $\mathbb{E}^{w_t \sim \mathbb{Q}_t^{\text{true}}}[\ell_f(x_T) +$

Table 5.1: Out-of-sample cost, total computation time, and average computation time per iteration for all algorithms computed over 1,000 simulations.

	DR-DDP	GT-DDP	box-DDP	NR-DDP
Out-of-sample cost	176.713	198.611	225.335	211.461
Total comp. time (sec.)	24.133	23.357	9.642	18.241
Comp. time per. iter. (sec.)	0.203	0.342	0.092	0.125

DR-DDP algorithm achieves an out-of-sample cost as low as 176.713, while box-DDP, NR-DDP, and GT-DDP demonstrate worse out-of-sample performance costs of 225.335, 211,461, and 198.611, respectively. These findings demonstrate the effectiveness of our algorithm in addressing distributional uncertainties in nonlinear stochastic systems.

5.4.2 Synchronization of Coupled Oscillators

In the second experiment, we demonstrate the scalability of our algorithm through a synchronization problem with L coupled noisy oscillators using the following discrete-time Kuramoto model [208]:

$$\eta_{t+1}^{(i)} = \eta_t^{(i)} + \Delta t\Big(\omega_i + \mathcal{K} u_t \sum_{j=1}^{L} \sin(\eta_t^{(j)} - \eta_t^{(i)})\Big) + w_t^{(i)},$$

where $i = 1, \ldots, L$. Here, $x_t = [\eta_t^{(1)}, \ldots, \eta_t^{(L)}]^\top \in \mathbb{R}^L$ is the system state, and $u_t \in \mathbb{R}$ is the control input. For each i-th oscillator, $\eta_t^{(i)}$ represents its phase, $\omega^{(i)}$ is its natural frequency, $\mathcal{K}$ is the coupling strength, and $\Delta t = 0.03\, sec.$ is the discretization step. We assume the frequencies $\omega^{(i)}$ and disturbances $w_t^{(i)}$ follow Gaussian distributions

$\sum_{t=0}^{T-1} \ell(x_t, \pi_t^*(x_t))]$, which is evaluated using 10,000 disturbance samples drawn from the true distribution $\mathbb{Q}_t^{\text{true}}$ and averaged over 200 simulations. It represents the expected total cost under a new disturbance sample generated according to the true disturbance distribution $\mathbb{Q}_t^{\text{true}}$ independent of the sample dataset used in DR-DDP.

$\mathcal{N}(0, 0.004)$ and $\mathcal{N}(0.001, 0.001)$, respectively. We aim to synchronize the oscillators within a finite horizon of $T = 100$, assuming that only $N = 50$ disturbance samples are available. For that, the cost function is designed as

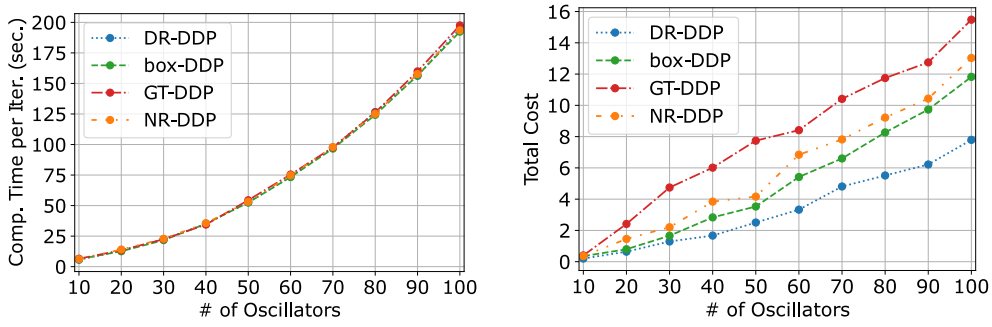
$$\ell(x_t, u_t) := \sum_{i,j=1}^{L} \sin^2(\eta_t^{(j)} - \eta_t^{(i)}) + 0.0001 u_t^2,$$

and the penalty parameter is chosen as $\lambda = 10000$ to minimize the upper bound (5.5).

To assess the scalability of our method, we evaluate the computation time to perform a single iteration of our DR-DDP algorithm depending on the number of oscillators. The computation times required for our method and the three baselines, along with the corresponding total costs, are presented in Fig. 5.2. As expected, the computation time increases with the number of oscillators. However, consistent with the theoretical complexity, the computation time grows as a polynomial function of the state dimension, showing the superiority of our method over the DP algorithm. Notably, the computation time required to perform a single iteration of DR-DDP is almost identical to the computation times required by box-DDP, NR-DDP, and GT-DDP. Furthermore, our DR-DDP algorithm consistently returns the lowest out-of-sample cost for any number of oscillators considered, successfully synchronizing the oscillators despite the disturbances.

5.5 Conclusions

In this work, we have proposed a practical DR-DDP algorithm for solving nonlinear stochastic optimal control problems with unknown disturbance distributions. Our approach leverages WDRC to address limited distributional information. We reformulated the quadratic approximation of value functions for WDRC using the Kantorovich duality principle and then solved it in a DDP fashion to obtain closed-form expressions of the distributionally robust control and distribution policies in each iteration. Our simulation results demonstrate the superior out-of-sample performance of



(a) (b)

Figure 5.2: (a) Computation time per iteration (in seconds) and (b) out-of-sample cost depending on the number of oscillators calculated over 1,000 simulations.

the proposed method compared to existing DDP methods, as well as its outstanding scalability to high-dimensional state spaces. In the future, we plan to investigate the theoretical properties of our algorithm, including its convergence rate and performance guarantees.

5.6 Appendix

5.6.1 Proof of Proposition 5.1

Proof. The proof is based on the arguments used in [42, Lemma 4.1] for the LQ case. Specifically, fix $\lambda > 0$. For any $\varepsilon > 0$, there exists $\gamma^{\varepsilon} \in \Gamma_{\mathbb{D}}$ such that

$$\sup_{\gamma \in \Gamma_{\mathbb{D}}} J(\pi, \gamma) - \epsilon < J(\pi, \gamma^{\epsilon}).$$

Since $\gamma^{\epsilon} \in \Gamma_{\mathbb{D}}$, it follows that $\gamma_t^{\varepsilon}(x_t) = \mathbb{P}_t \in \mathbb{D}_t$. Thus,

$$\begin{aligned} J(\pi, \gamma^{\epsilon}) &\leq \lambda T \theta^2 + J_{\lambda}(\pi, \gamma^{\epsilon}) \\ &\leq \lambda T \theta^2 + \sup_{\gamma \in \Gamma} J_{\lambda}(\pi, \gamma). \end{aligned}$$

Since this inequality holds for any $\varepsilon > 0$, we conclude that (5.5) holds. □

5.6.2 Proof of Proposition 5.2

Proof. We first note that by the definition of the Wasserstein distance, the inner supremum in (5.6) is equivalent to

$$\begin{aligned}
&\sup_{\mathbb{P}\in\mathcal{P}(\mathbb{R}^{n_w})} \mathbb{E}^{w\sim\mathbb{P}}\Big[V_{t+1}(f(\boldsymbol{x},\boldsymbol{u},w)) - \lambda W_2(\mathbb{P},\mathbb{Q}_t)^2\Big] \\
&= \sup_{\mathbb{P}\in\mathcal{P}(\mathbb{R}^{n_w})} \int_{\mathcal{W}} V_{t+1}(f(\boldsymbol{x},\boldsymbol{u},w))\mathrm{d}\mathbb{P}(w) \\
&\quad - \lambda \inf_{\substack{\tau\in\mathcal{P}(\mathcal{W}^2):\\ \Pi^1\tau=\mathbb{P},\Pi^2\tau=\mathbb{Q}_t}} \int_{\mathcal{W}^2} \|w-\hat{w}\|^2 \,\mathrm{d}\tau(w,\hat{w}) \\
&= \sup_{\substack{\tau\in\mathcal{P}(\mathcal{W}^2):\\ \Pi^2\tau=\mathbb{Q}_t}} \int_{\mathcal{W}^2} \Big[V_{t+1}(f(\boldsymbol{x},\boldsymbol{u},w)) - \lambda\|w-\hat{w}\|^2\Big]\,\mathrm{d}\tau(w,\hat{w})
\end{aligned} \tag{5.18}$$

According to the Kantorovich duality principle [39, 40],

$$W_2(\mathbb{P},\mathbb{Q})^2 = \sup_{\varphi,\psi\in\Phi}\left[\int_{\mathcal{W}}\varphi(w)\mathrm{d}\mathbb{P}(w) + \int_{\mathcal{W}}\psi(\hat{w})\mathrm{d}\mathbb{Q}_t(\hat{w})\right],$$

where $\Phi := \{(\varphi,\psi)\in L^1(\mathrm{d}w)\times L^2(\mathrm{d}\hat{w}) \mid \varphi(w)+\psi(\hat{w}) \le \|w-\hat{w}\|^2, \forall w,\hat{w}\in\mathcal{W}\}$. Thus, for any $(\varphi,\psi)\in\Phi$, we have that

$$\psi(\hat{w}) \le \inf_{\boldsymbol{w}\in\mathcal{W}} \|\hat{w}-\boldsymbol{w}\|^2 - \varphi(\boldsymbol{w})$$

for each $\hat{w}\in\mathcal{W}$. Consequently, for any $\lambda > 0$, *weak duality* holds for the inner problem as follows:

$$\begin{aligned}
&\sup_{\substack{\tau\in\mathcal{P}(\mathcal{W}^2):\\ \Pi^2\tau=\mathbb{Q}_t}} \int_{\mathcal{W}^2} \Big[V_{t+1}(f(\boldsymbol{x},\boldsymbol{u},w)) - \lambda\|\hat{w}-w\|^2\Big]\,\mathrm{d}\tau(w,\hat{w}) \\
&\le \int_{\mathcal{W}} \sup_{\boldsymbol{w}\in\mathcal{W}} \Big[V_{t+1}(f(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})) - \lambda\|\hat{w}-\boldsymbol{w}\|^2\Big]\,\mathrm{d}\mathbb{Q}_t(w').
\end{aligned} \tag{5.19}$$

Using Proposition 1 in [207], we can further show that strong duality holds for the inner problem for any $\lambda > 0$. We conclude the proof by substituting the expression for the inner supremum into (5.6). □

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we have introduced several novel approaches to address the WDRC problem in situations where the controller has limited information about the uncertainty distribution. Firstly, we have proposed a new DR-risk map tool for mobile robots operating in learning-enabled environments, which evaluates the risk of system unsafety in a distributionally robust manner, despite errors in the learning process. To evaluate the DR-risk map, we have introduced a computationally tractable SDP formulation with probabilistic guarantees on the loss of safety. We have demonstrated the effectiveness of this risk map for motion planning and control of mobile robots.

Next, we have improved the accuracy and efficiency of the learning-based motion controller by employing a UT-based uncertainty propagation scheme. We have also introduced a simple upper-bound replacement for the risk constraint, which proactively limits the risk of unsafety even under learning errors.

Furthermore, we have addressed the WDRC problem for discrete-time partially observable linear systems and proposed a novel approximation scheme to obtain tractable solutions. We have derived closed-form expressions of the control policy in both finite- and infinite-horizon settings. The proposed method features several important properties, including guaranteed cost, probabilistic out-of-sample performance guarantees, and closed-loop stability.

Finally, we have presented a novel DR-DDP algorithm for solving the nonlinear WDRC problem in situations where there is limited distributional information. Our approach unifies the previously mentioned methods and provides an explicit controller for nonlinear systems that can be readily applied in learning-enabled environments. We have demonstrated the effectiveness of the proposed frameworks through numerical experiments in various environments.

In conclusion, the contributions of this thesis provide novel approaches to address the WDRC problem in situations where the controller has limited information about the uncertainty distribution. These contributions have practical applications in various fields, such as safe learning and risk-aware control of autonomous systems and robotic decision-making with partial observations.

Future work includes applying the proposed methods to physical robots to validate their effectiveness in real-world scenarios. Furthermore, enhancing the adaptivity of the algorithms by updating the conservativeness in an online manner based on the observed safety margin would be a promising direction for future research. Additionally, extending the obtained results for partially observable settings to the case where the probability distribution of measurement noise is unknown would further increase the robustness of the proposed methods.

Bibliography

[1] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.

[2] P. Whittle, "Risk-sensitive linear/quadratic/Gaussian control," *Advances in Applied Probability*, vol. 13, no. 4, pp. 764–777, 1981.

[3] P. Kumar, "Optimal adaptive control of linear-quadratic-gaussian systems," *SIAM Journal on Control and Optimization*, vol. 21, no. 2, pp. 163–178, 1983.

[4] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, 2012, vol. 1.

[5] H. Mania, S. Tu, and B. Recht, "Certainty equivalence is efficient for linear quadratic control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[7] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[8] M. Farina, L. Giulioni, and R. Scattolini, "Stochastic linear model predictive control with chance constraints–a review," *Journal of Process Control*, vol. 44, pp. 53–67, 2016.

[9] G. Ripaccioli, D. Bernardini, S. Di Cairano, A. Bemporad, and I. Kolmanovsky, "A stochastic model predictive control approach for series hybrid electric vehicle power management," in *EEE American control Conference*, 2010, pp. 5844–5849.

[10] M. Bichi, G. Ripaccioli, S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic model predictive control with driver behavior learning for improved powertrain control," in *IEEE Conference on Decision and Control*, 2010, pp. 6077–6082.

[11] G. Schildbach, G. C. Calafiore, L. Fagiano, and M. Morari, "Randomized model predictive control for stochastic linear systems," in *IEEE American Control Conference*, 2012, pp. 417–422.

[12] L. Hewing and M. N. Zeilinger, "Scenario-based probabilistic reachable sets for recursively feasible stochastic model predictive control," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 450–455, 2019.

[13] A. L. Visintini, W. Glover, J. Lygeros, and J. Maciejowski, "Monte Carlo optimization for conflict resolution in air traffic control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 470–482, 2006.

[14] N. Kantas, J. Maciejowski, and A. Lecchini-Visintini, "Sequential Monte Carlo for model predictive control," *Nonlinear model predictive control: Towards new challenging applications*, pp. 263–273, 2009.

[15] I. Khalil, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.

[16] I. Tzortzis, C. D. Charalambous, T. Charalambous, C. K. Kourtellaris, and C. N. Hadjicostis, "Robust linear quadratic regulator for uncertain systems," in *IEEE Conference on Decision and Control*, 2016, pp. 1515–1520.

[17] V. A. Ugrinovskii and I. R. Petersen, "Finite horizon minimax optimal control of stochastic partially observed time varying uncertain systems," *Mathematics of Control, Signals, and Systems*, vol. 12, no. 1, pp. 1–23, 1999.

[18] Y. Shi and B. Yu, "Robust mixed H_2/H_∞ control of networked control systems with random time delays in both forward and backward communication links," *Automatica*, vol. 47, no. 4, pp. 754–760, 2011.

[19] T. Başar and P. Bernhard, *H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer Science & Business Media, 2008.

[20] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard H_2 and H_∞ control problems," in *IEEE American Control Conference*. IEEE, 1988, pp. 1691–1696.

[21] L. Xie, "Output feedback H_∞ control of systems with parameter uncertainty," *International Journal of Control*, vol. 63, no. 4, pp. 741–750, 1996.

[22] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 2007, pp. 207–226.

[23] Z. Q. Zheng and M. Morari, "Robust stability of constrained model predictive control," in *IEEE American Control Conference*, 1993, pp. 379–383.

[24] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[25] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based mpc for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[26] D. Q. Mayne, E. C. Kerrigan, E. Van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[27] B. P. Van Parys, D. Kuhn, P. J. Goulart, and M. Morari, "Distributionally robust control of constrained stochastic systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 430–442, 2015.

[28] M. Schuurmans and P. Patrinos, "Data-driven distributionally robust control of partially observable jump linear systems," in *IEEE Conference on Decision and Control*, 2021, pp. 4332–4337.

[29] P. Coppens, M. Schuurmans, and P. Patrinos, "Data-driven distributionally robust LQR with multiplicative noise," in *Learning for Dynamics and Control*, 2020, pp. 521–530.

[30] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3863–3870, 2020.

[31] J. Coulson, J. Lygeros, and F. Dorfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, 2021.

[32] C. Mark and S. Liu, "Stochastic MPC with distributionally robust chance constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7136–7141, 2020.

[33] A. Hakobyan and I. Yang, "Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 939–957, 2021.

[34] K. Kim and I. Yang, "Minimax control of ambiguous linear stochastic systems using the Wasserstein metric," in *IEEE Conference on Decision and Control*, 2020, pp. 1777–1784.

[35] H. Xu and S. Mannor, "Distributionally robust Markov decision processes," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[36] G. C. Calafiore, "Ambiguous risk measures and optimal robust portfolios," *SIAM Journal on Optimization*, vol. 18, no. 3, pp. 853–877, 2007.

[37] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.

[38] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.

[39] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1-2, pp. 115–166, 2018.

[40] R. Gao and A. Kleywegt, "Distributionally robust stochastic optimization with Wasserstein distance," *Mathematics of Operations Research*, 2022.

[41] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*. INFORMS, 2019, pp. 130–166.

[42] K. Kim and I. Yang, "Distributional robustness in minimax linear quadratic control with Wasserstein distance," *SIAM Journal on Control and Optimization*, 2022.

[43] D. Boskos, J. Cortés, and S. Martínez, "Data-driven ambiguity sets with probabilistic guarantees for dynamic processes," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 2991–3006, 2020.

[44] I. R. Petersen, M. R. James, and P. Dupuis, "Minimax optimal control of stochastic uncertain systems with relative entropy constraints," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 398–412, 2000.

[45] V. A. Ugrinovskii and I. R. Petersen, "Minimax LQG control of stochastic partially observed uncertain systems," *SIAM Journal on Control and Optimization*, vol. 40, no. 4, pp. 1189–1226, 2002.

[46] H. Nakao, R. Jiang, and S. Shen, "Distributionally robust partially observable Markov decision process with moment-based ambiguity," *SIAM Journal on Optimization*, vol. 31, no. 1, pp. 461–488, 2021.

[47] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[48] T. Wei and C. Liu, "Safe control with neural network dynamic models," in *Learning for Dynamics and Control Conference*, 2022, pp. 739–750.

[49] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for Lidar 3D vehicle detection," in *IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 3266–3273.

[50] A. Lederer, A. J. O. Conejo, K. A. Maier, W. Xiao, J. Umlauft, and S. Hirche, "Gaussian process-based real-time learning for safety critical applications," in *International Conference on Machine Learning*, 2021, pp. 6055–6064.

[51] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[52] J. Fan and W. Li, "Safety-guided deep reinforcement learning via online Gaussian process estimation," *arXiv preprint arXiv:1903.02526*, 2019.

[53] L. Brunke, S. Zhou, and A. P. Schoellig, "Barrier Bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems," in *Learning for Dynamics and Control Conference*, 2022, pp. 881–892.

[54] C. D. McKinnon and A. P. Schoellig, "Learn fast, forget slow: Safe predictive learning control for systems with unknown and changing dynamics performing repetitive tasks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2180–2187, 2019.

[55] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *IEEE Conference on Decision and Control*, 2014, pp. 1424–1431.

[56] Y. S. Shao, C. Chen, S. Kousik, and R. Vasudevan, "Reachability-based trajectory safeguard (RTS): A safe and fast reinforcement learning safety layer for continuous control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3663–3670, 2021.

[57] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in un-

known environments," in *IEEE Conference on Decision and Control*, 2019, pp. 1758–1765.

[58] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[59] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[60] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2460–2465.

[61] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[62] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 279–284.

[63] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *IEEE Conference on Decision and Control*, 2018, pp. 6059–6066.

[64] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *IEEE Conference on Decision and Control*, 2018, pp. 7130–7135.

[65] ——, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[66] A. Jain, T. Nghiem, M. Morari, and R. Mangharam, "Learning and control using Gaussian processes," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2018, pp. 140–149.

[67] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.

[68] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[69] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Robotics Research*. Springer, 2020, pp. 75–84.

[70] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using CVaR-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019.

[71] N. A. Urpí, S. Curi, and A. Krause, "Risk-averse offline reinforcement learning," *arXiv preprint arXiv:2102.05371*, 2021.

[72] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 6518–6523.

[73] V. Renganathan, I. Shames, and T. H. Summers, "Towards integrated perception and motion planning with distributionally robust risk constraints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 530–15 536, 2020.

[74] A. Hakobyan and I. Yang, "Learning-based distributionally robust motion control with Gaussian processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 7667–7674.

[75] ——, "Distributionally robust risk map for learning-based motion planning and control: A semidefinite programming approach," *IEEE Transactsions on Robotics*, 2022.

[76] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[77] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *Conference on Robot Learning*, 2018, pp. 466–476.

[78] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control Lyapunov functions for uncertain robotic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 6878–6884.

[79] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 3387–3395.

[80] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control Conference*, 2020, pp. 708–717.

[81] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

[82] C. Richter, J. Ware, and N. Roy, "High-speed autonomous navigation of unknown environments using learned probabilities of collision," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 6114–6121.

[83] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 1, pp. 137–147, 2008.

[84] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk assessment at road intersections: Comparing intention and expectation," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 165–171.

[85] A. Dixit, M. Ahmadi, and J. W. Burdick, "Risk-sensitive motion planning using entropic value-at-risk," in *IEEE European Control Conference*, 2021, pp. 1726–1732.

[86] D. Di Castro, A. Tamar, and S. Mannor, "Policy gradients with variance related risk criteria," *arXiv preprint arXiv:1206.6404*, 2012.

[87] S. Mannor and J. Tsitsiklis, "Mean-variance optimization in Markov decision processes," *arXiv preprint arXiv:1104.5601*, 2011.

[88] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety analysis using conditional value-at-risk," *IEEE Transactions on Automatic Control*, 2021.

[89] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 723–730.

[90] W. Liu and M. H. Ang, "Incremental sampling-based algorithm for risk-aware planning under motion uncertainty," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 2051–2058.

[91] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[92] T. Hester, M. Quinlan, and P. Stone, "RTMBA: A real-time model-based reinforcement learning architecture for robot control," in *IEEE International Conference on Robotics and Automation*, 2012.

[93] A. Venkatraman, R. Capobianco, L. Pinto, M. Hebert, D. Nardi, and J. A. Bagnell, "Improved learning of dynamics models for control," in *International Symposium on Experimental Robotics*, 2016.

[94] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 2, pp. 153–173, 2017.

[95] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *IEEE International Conference on Robotics and Automation*, 2015.

[96] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *IEEE International Conference on Robotics and Automation*, 2015.

[97] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *The International Journal of Robotic Research*, vol. 36, no. 10, pp. 1073–1087, 2017.

[98] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium*, 2017.

[99] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE International Conference on Robotics and Automation*, 2018.

[100] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using Gaussian mixture models," in *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[101] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, "Deep neural networks for Markovian interactive scene prediction in highway scenarios," in *IEEE Intelligent Vehicles Symposium*, 2017.

[102] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Probabilistic motion planning among moving obstacles following typical motion patterns," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4027–4033.

[103] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 902–907.

[104] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robotss*, vol. 35, no. 1, pp. 51–76, 2013.

[105] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[106] W. Chi and M. Q.-H. Meng, "Risk-RRT*: A robot motion planning algorithm for the human robot coexisting environment," in *International Conference on Advanced Robotics*, 2017, pp. 583–588.

[107] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letter*, vol. 4, no. 4, pp. 4459–4466, 2019.

[108] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[109] S. Sarykalin, G. Serraino, and S. Uryasev, "Value-at-risk vs. conditional value-at-risk in risk management and optimization," in *State-of-the-art decision-making tools in the information-intensive age*. Informs, 2008, pp. 270–294.

[110] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[111] V. D. Sharma, M. Toubeh, L. Zhou, and P. Tokekar, "Risk-aware planning and assignment for ground vehicles using uncertain perception from aerial vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 11 763–11 769.

[112] S. Singh, J. Lacotte, A. Majumdar, and M. Pavone, "Risk-sensitive inverse reinforcement learning via semi- and non-parametric methods," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1713–1740, 2018.

[113] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-sensitive path planning via CVaR barrier functions: Application to bipedal locomotion," *arXiv preprint arXiv:2011.01578*, 2020.

[114] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.

[115] K.-C. Toh, M. J. Todd, and R. H. T
"ut

”unc

”u, “SDPT3—a MATLAB software package for semidefinite programming, version 1.3,” *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[116] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[117] B. O’donoghue, E. Chu, N. Parikh, and S. Boyd, “Conic optimization via operator splitting and homogeneous self-dual embedding,” *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.

[118] M. Kočvara and M. Stingl, “PENNON: A code for convex nonlinear and semidefinite programming,” *Optimization Methods and Software*, vol. 18, no. 3, pp. 317–333, 2003.

[119] M. ApS, “Mosek optimization suite,” 2019.

[120] A. Lederer, J. Umlauft, and S. Hirche, “Uniform error bounds for Gaussian process regression with application to safe control,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[121] ——, “Uniform error and posterior variance bounds for gaussian process regression with application to safe control,” *arXiv preprint arXiv:2101.05328*, 2021.

[122] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotic Research*, vol. 30, no. 7, pp. 846–894, 2011.

[123] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 1478–1483.

[124] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *AIAA Guidance, Navigation, and Control Conference*, 2013, p. 5097.

[125] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 8160.

[126] W. Chi, J. Wang, and M. Q.-H. Meng, "Risk-Informed-RRT*: A sampling-based human-friendly motion planning algorithm for mobile service robots in indoor environments," in *IEEE International Conference on Information and Aautomation*, 2018, pp. 1101–1106.

[127] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[128] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[129] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[130] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[131] L. Liberti, "Introduction to global optimization," *Ecole Polytechnique*, 2008.

[132] E. M. Smith and C. C. Pantelides, "A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 457–478, 1999.

[133] H. S. Ryoo and N. V. Sahinidis, "Global optimization of nonconvex NLPs and MINLPs with applications in process design," *Computers & Chemical Engineering*, vol. 19, no. 5, pp. 551–566, 1995.

[134] I. Quesada and I. E. Grossmann, "A global optimization algorithm for linear fractional and bilinear programs," *Journal of Global Optimization*, vol. 6, no. 1, pp. 39–76, 1995.

[135] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, no. 1, pp. 13–29, 2020.

[136] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for MAVs in dynamic environments," *IEEE Robotics and Automation Letter*, vol. 4, no. 2, pp. 776–783, 2019.

[137] M. Gelbrich, "On a formula for the L2 Wasserstein metric between measures on Euclidean and Hilbert spaces," *Mathematische Nachrichten*, vol. 147, no. 1, pp. 185–203, 1990.

[138] S. Zymler, D. Kuhn, and B. Rustem, "Distributionally robust joint chance constraints with second-order moment information," *Mathematical Programming*, vol. 137, no. 1, pp. 167–198, 2013.

[139] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[140] A. Elhafsi, B. Ivanovic, L. Janson, and M. Pavone, "Map-predictive motion planning in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 8552–8558.

[141] T. Salzmann, E. Kaufmann, M. Pavone, D. Scaramuzza, and M. Ryll, "Neural-MPC: Deep learning model predictive control for quadrotors and agile robotic platforms," *arXiv preprint arXiv:2203.07747*, 2022.

[142] K. Seel, M. Haring, E. I. Grøtli, K. Y. Pettersen, and J. T. Gravdahl, "Learning-based robust model predictive control for sector-bounded lur'e systems," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 46–52, 2021.

[143] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[144] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.

[145] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[146] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: a data-driven risk-aware approach," *IEEE Control Systems Letters*, 2022.

[147] D. Li, D. Fooladivanda, and S. Martínez, "Online learning of parameterized uncertain dynamical environments with finite-sample guarantees," in *IEEE American Control Conference*, 2021, pp. 2005–2010.

[148] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 11 205–11 212.

[149] Y. K. Nakka, A. Liu, G. Shi, A. Anandkumar, Y. Yue, and S.-J. Chung, "Chance-constrained trajectory optimization for safe exploration and learning of nonlinear systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 389–396, 2020.

[150] J. Ma, Z. Cheng, X. Zhang, A. A. Mamun, C. W. de Silva, and T. H. Lee, "Non-parametric behavior learning for multi-agent decision making with chance constraints: A data-driven predictive control framework," *arXiv preprint arXiv:2011.03213*, 2020.

[151] A. Wang, A. Jasour, and B. C. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, 2020.

[152] A. Thorpe, T. Lew, M. Oishi, and M. Pavone, "Data-driven chance constrained control using kernel distribution embeddings," in *Learning for Dynamics and Control Conference*, 2022, pp. 790–802.

[153] I. G. Jin, B. Sch
"urmann, R. M. Murray, and M. Althoff, "Risk-aware motion planning for automated vehicle among human-driven cars," in *American Control Conference*, 2019, pp. 3987–3993.

[154] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-averse control via CVaR barrier functions: Application to bipedal robot locomotion," *IEEE Control Systems Letters*, vol. 6, pp. 878–883, 2021.

[155] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[156] A. R. Hota, A. Cherukuri, and J. Lygeros, "Data-driven chance constrained optimization under Wasserstein ambiguity sets," in *2019 American Control Conference (ACC)*, 2019, pp. 1501–1506.

[157] A. Zolanvari and A. Cherukuri, "Data-driven distributionally robust iterative risk-constrained model predictive control," in *IEEE European Control Conference*, 2022, pp. 1578–1583.

[158] C. Liu, A. Gray, C. Lee, J. K. Hedrick, and J. Pan, "Nonlinear stochastic predictive control with unscented transformation for semi-autonomous vehicles," in *IEEE American Control Conference*, 2014, pp. 5574–5579.

[159] E. Bradford and L. Imsland, "Stochastic nonlinear model predictive control with state estimation by incorporation of the unscented Kalman filter," *arXiv preprint arXiv:1709.01201*, 2017.

[160] T. Knudsen and J. Leth, "Stochastic MPC using the unscented transform," in *IEEE Annual American Control Conference*, 2018, pp. 4718–4724.

[161] M. I. Ribeiro, "Kalman and extended Kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics*, vol. 43, p. 46, 2004.

[162] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, 1997, pp. 182–193.

[163] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[164] S. J. Julier, "The scaled unscented transformation," in *IEEE American Control Conference*, vol. 6, 2002, pp. 4555–4559.

[165] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1901–1907.

[166] V. A. Nguyen, S. S. Abadeh, D. Filipović, and D. Kuhn, "Mean-covariance robust risk measurement," *arXiv preprint arXiv:2112.09959*, 2021.

[167] Y.-L. Yu, Y. Li, D. Schuurmans, and C. Szepesvári, "A general projection property for distribution families," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[168] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[169] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017, pp. 1–16.

[170] K. J. Åström, *Introduction to Stochastic Control Theory*. Courier Corporation, 2012.

[171] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. SIAM, 2015.

[172] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[173] S. Samuelson and I. Yang, "Data-driven distributionally robust control of energy storage to manage wind power fluctuations," in *IEEE Conference on Control Technology and Applications*, 2017.

[174] I. Yang, "A dynamic game approach to distributionally robust safety specifications for stochastic systems," *Automatica*, vol. 94, pp. 94–101, 2018.

[175] I. Tzortzis, C. D. Charalambous, and T. Charalambous, "Infinite horizon average cost dynamic programming subject to total variation distance ambiguity," *SIAM Journal on Control and Optimization*, vol. 57, no. 4, pp. 2843–2872, 2019.

[176] P. Coppens and P. Patrinos, "Data-driven distributionally robust MPC for constrained stochastic systems," *IEEE Control Systems Letters*, vol. 6, no. 1274–1279, 2021.

[177] C. Mark and S. Liu, "Data-driven distributionally robust MPC: An indirect feedback approach," *arXiv preprint arXiv:2109.09558*, 2021.

[178] I. Tzortzis, C. D. Charalambous, and C. N. Hadjicostis, "A distributionally robust LQR for systems with multiple uncertain players," in *IEEE Conference on Decision and Control*, 2021.

[179] A. Zolanvari and A. Cherukuri, "Data-driven distributionally robust iterative risk-constrained model predictive control," in *IEEE European Control Conference*, 2022.

[180] Z. Zhong, E. A. del Rio-Chanona, and P. Petsagkourakis, "Distributionally robust MPC for nonlinear systems," in *IFAC-PapersOnLine*, vol. 55, no. 7, 2022, pp. 606–613.

[181] A. Dixit, M. Ahmadi, and J. W. Burdick, "Distributionally robust model predictive control with total variation distance," *arXiv preprint arXiv:2203.12062*, 2022.

[182] F. Micheli, T. Summers, and J. Lygeros, "Data-driven distributionally robust MPC for systems with uncertain dynamics," in *IEEE Conference on Decision and Control*, 2022.

[183] G. Bayraksan and D. K. Love, "Data-driven stochastic programming using phi-divergences," in *Tutorials in Operations Research*, 2015, pp. 1–19.

[184] C. Zhao and Y. Guan, "Data-driven risk-averse stochastic optimization with Wasserstein metric," *Operations Research Letters*, vol. 46, no. 2, pp. 262–267, 2018.

[185] T. Osogami, "Robust partially observable Markov decision process," in *International Conference on Machine Learning*, 2015, pp. 106–115.

[186] S. Saghafian, "Ambiguous partially observable Markov decision processes: Structural results and applications," *Journal of Economic Theory*, vol. 178, pp. 1–35, 2018.

[187] J. I. González-Trejo, O. Hernández-Lerma, and L. F. Hoyos-Reyes, "Minimax control of discrete-time stochastic systems," *SIAM Journal on Control and Optimization*, vol. 41, no. 5, pp. 1626–1659, 2003.

[188] O. Hernández-Lerma and J. B. Lasserre, *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Springer, 2012.

[189] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Courier Corporation, 2012.

[190] N. Fournier and A. Guillin, "On the rate of convergence in Wasserstein distance of the empirical measure," *Probability Theory and Related Fields*, vol. 162, no. 3–4, pp. 707–738, 2015.

[191] L.-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 36, no. 6, pp. 692–706, 1991.

[192] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation*, 2014.

[193] A. Pavlov, I. Shames, and C. Manzie, "Interior point differential dynamic programming," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2720–2727, 2021.

[194] W. Jallet, N. Mansard, and J. Carpentier, "Implicit differential dynamic programming," in *IEEE International Conference on Robotics and Automation*, 2022.

[195] O. So, Z. Wang, and E. A. Theodorou, "Maximum entropy differential dynamic programming," in *IEEE International Conference on Robotics and Automation*, 2022.

[196] V. Roulet, S. Srinivasa, M. Fazel, and Z. Harchaoui, "Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates," *arXiv preprint arXiv:2207.06362*, 2022.

[197] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *IEEE American Control Conference*, 2005.

[198] E. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *IEEE American Control Conference*, 2010.

[199] Y. Pan and E. A. Theodorou, "Data-driven differential dynamic programming using Gaussian processes," in *IEEE American Control Conference*, 2015.

[200] Y. Pan, G. I. Boutselis, and E. A. Theodorou, "Efficient reinforcement learning via probabilistic trajectory optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5459–5474, 2018.

[201] W. Sun, Y. Pan, J. Lim, E. A. Theodorou, and P. Tsiotras, "Min-max differential dynamic programming: Continuous and discrete time formulations," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 12, pp. 2568–2580, 2018.

[202] J. Morimoto, G. Zeglin, and C. G. Atkeson, "Minimax differential dynamic programming: Application to a biped walking robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[203] M. Schuurmans and P. Patrinos, "Learning-based distributionally robust model predictive control of Markovian switching systems with guaranteed stability and recursive feasibility," in *IEEE Conference on Decision and Control*, 2020.

[204] Z. Zhong, E. A. del Rio-Chanona, and P. Petsagkourakis, "Data-driven distributionally robust MPC using the Wasserstein metric," *arXiv preprint arXiv:2105.08414*, 2021.

[205] A. B. Kordabad, R. Wisniewski, and S. Gros, "Safe reinforcement learning using Wasserstein distributionally robust MPC and chance constraint," *IEEE Access*, vol. 10, pp. 130 058–130 067, 2022.

[206] A. Hakobyan and I. Yang, "Wasserstein distributionally robust control of partially observable linear systems: Tractable approximation and performance guarantee," in *IEEE Conference on Decision and Control*, 2022.

[207] A. Sinha, H. Namkoong, R. Volpi, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," *arXiv preprint arXiv:1710.10571*, 2017.

[208] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. Springer Science & Business Media, 2012, vol. 19.

초 록

분포 강건 제어(Distributionally robust control, DRC)와 분포 강건 최적화(Distributionally robust optimization, DRO)는 최근에 스토캐스틱 시스템에서 부정확한 분포 정보를 처리하는 효과적인 방법으로 등장하였다. 본 연구에서는 시스템 또는 환경 모델의 불확실성에 대한 제한된 정보만이 주어진 자율 시스템에 대한 새로운 제어 방법을 개발한다. 이를 위해, 주어진 데이터를 이용하여 불확실성 분포를 추정하고, 해당 분포를 중심으로 ambiguity set을 구성한다. Ambiguity set은 추정된 분포로부터 Wasserstein 거리가 주어진 반지름보다 작은 모든 분포를 포함한다. 추정 결과의 불확실성을 고려하기 위해 ambiguity set 내에서 최악의 경우 분포에 대한 최적 제어 문제를 푼다. 그러나 이 문제는 무한 차원 최적화 문제이기 때문에, DRO 분야의 최신 도구를 적용하여 Wasserstein DRC(WDRC) 문제를 계산 가능한 형태로 바꾸는 범용적인 여러 방법을 개발한다. 이 방법들은 다양한 이론적 특성을 가지며, 여러 응용 분야에서 탁월한 성능을 보인다.

본 연구에서 제안하는 첫 번째 방법은 학습 가능한 환경에서 이동 로봇의 동작 계획과 제어를 위한 분포적으로 강건한 위험 함수(DR-risk map)이라는 새로운 안전 평가 방법을 제안한다. DR-risk map은 Gaussian process regression(GPR)에 의해 움직임이 추론되는 장애물과의 충돌 위험성을 안정적으로 계산한다. 이 방법은 추론된 분포의 오류를 고려하기 위해 ambiguity set 내 최악의 분포에 대한 위험을 측정한다. 무한차원 특성으로 인한 문제를 해결하기 위해, DR-risk map의 상한을 해로 갖는 semidefinite programming(SDP) 문제를 유도한다. 더 나아가 DR-risk map을 학습 가능 환경에서 자율 시스템의 동작 계획 및 제어를 수행하기 위해 적용한다.

본 논문에서 제안하는 두번 째 방법은 unscented transform을 사용하는 새로운

학습 기반의 동작 제어 도구이다, 이 방법은 GPR에서 이루어지는 불확실성 전파에 unscented transform을 적용함으로써 분포 추정 정확성과 계산 효율성을 향상시킨다. 또한, 임의의 안전 손실 함수에 대한 DR-risk 제약 조건을 대체하는 새로운 상한을 제시한다.

분포 강건 제어의 아이디어는 완전 관찰 가능 시스템보다 현실적인 부분적 관찰 가능한 스토캐스틱 시스템에도 적용 가능하다. 특히 본 논문에서는 부분적 관찰 가능한 선형 스토캐스틱 시스템을 위한 WDRC 문제를 고려하고, Wasserstein 거리의 Gelbrich 상한을 이용한 새로운 근사 문제를 제안하고, 최적 제어 정책의 closed-form 표현과, 최악의 분포 정책을 찾는 SDP 문제를 finite-horizon 및 infinite-horizon 설정에서 모두 유도한다. 제안한 방법은 out-of-sample performance에 대한 보장과 안정성 등 여러 가지 중요한 이론적 특성을 가지며, 제어 정책의 분포 강건성을 보장한다.

마지막으로, 일반적인 비선형 WDRC 문제를 해결하기 위한 새로운 분포 강건한 differential dynamic programming 방법을 제시한다. 이 방법은 비선형 스토캐스틱 시스템에 대한 closed-loop 제어 정책을 제공하며, 학습 가능한 환경에도 적용 가능하다는 측면에서 열거한 방법론들을 포괄한다. 이 접근법은 value function의 분해와 국소적 이차 근사를 특징으로 하여, 최소-최대 최적화 문제를 수치적으로 풀 필요 없이 효율적이고 고차원 시스템에도 쉽게 적용 가능하다.

다양한 시스템에서 실증적 연구를 통해 본 논문에서 소개된 방법론들의 성능과 효율성을 분석하고 입증한다. 결론적으로, 본 연구에서 제안한 방법들을 통하여 시스템 및 환경, 그리고 추론 결과의 분포적 불확실성을 체계적으로 다룰 수 있는 제어 정책을 제공한다.

주요어: 분포 강인 최적화, 분포 강인 최적제어, 동작 계획, 동작 제어, 로봇 안전
학번: 2021-37761

ACKNOWLEDGEMENT

With profound gratitude, I extend my heartfelt appreciation to all those who have supported me throughout this challenging and rewarding journey of pursuing my Ph.D. Their unwavering encouragement, guidance, and assistance have been the cornerstone in transforming this thesis from a dream into a reality.

I am deeply indebted to my esteemed supervisor, Dr. Yang, for his exceptional support and invaluable mentorship. His wealth of knowledge, patience, and unwavering belief in my potential have been instrumental in shaping the direction of my research and honing my academic abilities. I am immensely grateful for his constant motivation and dedicated guidance throughout this academic pursuit.

I would like to express my sincere appreciation to the members of my thesis committee, Dr. Park, Dr. Kim, Dr. Jung, and Dr. Hovakimyan, for their invaluable insights, critical feedback, and thoughtful suggestions. Their constructive comments have played a crucial role in elevating the quality and depth of this work.

I want to extend my gratitude to my colleagues and fellow researchers at the CORE lab, with whom I have shared insightful discussions and memorable experiences. Their camaraderie and unwavering support have made the challenges more manageable and the victories more meaningful.

To my dear parents, words cannot adequately express the depth of my gratitude. Your unconditional love, unwavering belief in me, and the countless sacrifices you've made have been the driving force behind my pursuit of knowledge. Despite being separated by thousands of miles, your constant encouragement and understanding during

the highs and lows of this journey have been my guiding light, and I cherish the values you have instilled in me.

To my dear siblings, Seda and Davit, thank you for always standing by my side and cheering me on. Your unwavering support and belief in my abilities have given me the strength to overcome obstacles and strive for excellence. I am grateful for the bond we share and the love that continues to inspire me.

I also want to express my heartfelt appreciation to my beloved grandparents and Aunt for their continuous support and encouragement. Your belief in my potential has motivated me to persevere and reach new heights in my academic journey. Your wisdom and love have been a source of inspiration, and I am grateful for the cherished memories we've shared.

Additionally, I extend my heartfelt appreciation to my friends who have been pillars of emotional support throughout this demanding endeavor. Your unwavering friendship, laughter, and encouragement have brought joy and balance to the five years in Korea. Your presence in my life has made this journey memorable and meaningful.

In conclusion, I extend my deepest appreciation to everyone who has been a part of this academic pursuit. Your contributions, whether big or small, have played a vital role in shaping this work and my growth as a researcher.

Thank you all for being an integral part of this fulfilling journey, and for being my unwavering support system, even from afar.

Seoul, 21 July 2023 Astghik Hakobyan