M.S. THESIS

RISK-AWARE DISTRIBUTIONALLY ROBUST OPTIMIZATION FOR LEARNING-BASED AUTONOMOUS SYSTEMS

학습 기반 자율시스템의 리스크를
고려하는 분포적 강인 최적화

BY

ASTGHIK HAKOBYAN

AUGUST 2020

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

M.S. THESIS

RISK-AWARE DISTRIBUTIONALLY ROBUST OPTIMIZATION FOR LEARNING-BASED AUTONOMOUS SYSTEMS

학습 기반 자율시스템의 리스크를
고려하는 분포적 강인 최적화

BY

ASTGHIK HAKOBYAN
AUGUST 2020

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

RISK-AWARE DISTRIBUTIONALLY ROBUST OPTIMIZATION FOR LEARNING-BASED AUTONOMOUS SYSTEMS

학습 기반 자율시스템의 리스크를
고려하는 분포적 강인 최적화

지도교수 양 인 순
이 논문을 공학석사 학위논문으로 제출함

2020년 5월

서울대학교 대학원

전기 · 정보공학부

아스트힉

아스트힉의 공학석사 학위 논문을 인준함

2020년 5월

위 원 장: _____

부위원장: _____

위 원: _____

Abstract

In this thesis, a risk-aware motion control scheme is considered for autonomous systems to avoid randomly moving obstacles when the true probability distribution of uncertainty is unknown. We propose a novel model predictive control (MPC) method for motion planning and decision-making that systematically adjusts the safety and conservativeness in an environment with randomly moving obstacles. The key component is the Conditional Value-at-Risk (CVaR), employed to limit the safety risk in the MPC problem. Having the empirical distribution obtained using a limited amount of sample data, Sample Average Approximation (SAA) is applied to compute the safety risk. Furthermore, we propose a method, which limits the risk of unsafety even when the true distribution of the obstacles' movements deviates, within an *ambiguity set*, from the empirical one. By choosing the ambiguity set as a statistical ball with its radius measured by the *Wasserstein metric*, we achieve a probabilistic guarantee of the *out-of-sample risk*, evaluated using new sample data generated independently of the training data. A set of reformulations are applied on both SAA-based MPC (SAA-MPC) and Wasserstein Distributionally Robust MPC (DR-MPC) to make them tractable.

In addition, we combine the DR-MPC method with Gaussian Process (GP) to predict the future motion of the obstacles from past observations of the environment.

The performance of the proposed methods is demonstrated and analyzed through simulation studies using a nonlinear vehicle model and a linearized quadrotor model.

keywords: Distributionally Robust Optimization, Optimal Control, Motion Control, Collision Avoidance, Robot Safety

student number: 2018-26905

Contents

List of Tables

List of Figures

Chapter 1

BACKGROUND AND OBJECTIVES

1.1 Motivation and Objectives

Safety is one of the most fundamental challenges in the operation of robots and autonomous systems in practical environments, which are uncertain and dynamic. In particular, the unpredicted motion of objects and agents often risks the collision-free navigation of mobile robots, as for examples in Fig. 1.1. One of the simplest and most popular ways to estimate the probability distribution of the obstacle's motion is to use data to construct an empirical distribution. Then, one might use well-known Sample Average Approximation (SAA) to limit the risk of unsafety. However, even though SAA is quite effective with asymptotic optimality, it does not ensure the satisfaction of risk constraints, particularly with a small sample size.

The research question to be addressed in this work is as follows: *Can a robot make a safe decision using an unreliable distribution estimated from small samples?*

To answer this question, we develop an optimization-based motion control method that uses a limited amount of data for making a risk-aware decision and improving its performance to ensure a finite-sample probabilistic guarantee of collision avoidance.

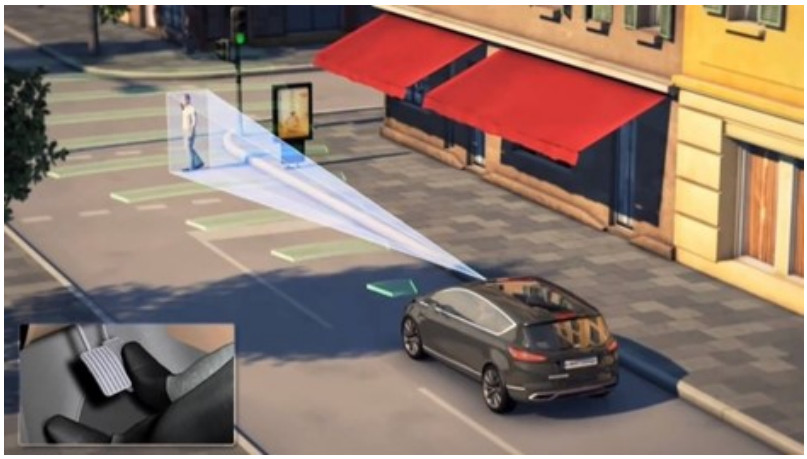


Figure 1.1: A person with unknown motion, interfering the vehicle navigation (Licensed under CC BY-NC).

1.2 Research Contributions

The contributions of this work can be summarized as follows. First, a novel model predictive control (MPC) method is proposed to limit the risk of unsafety through Conditional Value-at-Risk constraint by incorporating available data samples, as well as taking advantage of the obstacles geometry. Second, we improve the method to make sure that the safety constraint holds for any perturbation of the empirical distribution within the Wasserstein ambiguity set. Thus, the resulting control decision is guaranteed to satisfy the risk constraints for avoiding randomly moving obstacles in the presence of allowable distribution errors. Moreover, the proposed method provides a finite-sample probabilistic guarantee of limiting *out-of-sample risk*, meaning that the risk constraints are satisfied with probability no less than a certain threshold even when evaluated with new sample data chosen independently of the training data. Second, for computational tractability, we reformulate both the SAA-based MPC and the Wasserstein Distributionally Robust MPC (DR-MPC) problems into a finite-dimensional nonconvex optimization problem. The proposed reformulation procedures are developed using geometry of the obstacles, as well as modern DRO techniques based on the Kan-

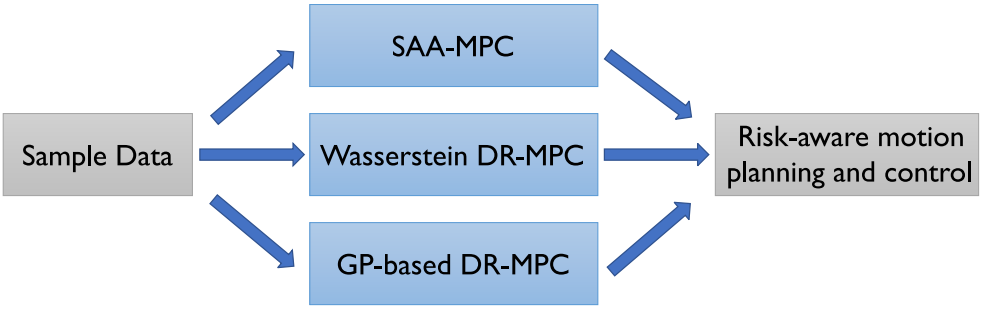


Figure 1.2: The three methods proposed in this work use the available sample data for risk-aware motion planning and control.

torovich duality principle [1]. Third, a Gaussian Process is used to predict obstacles' motion in combination with DR-MPC in the previous stages.

The performance and utility of the proposed methods are demonstrated through simulation studies. For the SAA-based risk-aware MPC, we used linearize quadrotor model and investigated its performance under different parameters. For the Wasserstein DR-MPC we performed simulations on a nonlinear car-like vehicle model, as well as the same linearized quadrotor model as in the first simulations. Finally, for Learning-based Wasserstein DR-MPC we used a nonlinear on-road vehicle model to demonstrate the utility of the method.

The results of numerical experiments confirm that, even when the sample size is small, the proposed DR-MPC method can successfully avoid randomly moving obstacles with a guarantee of limiting out-of-sample risk, while its SAA counterpart fails to do so.

1.3 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, a two-stage motion planning and control method based on CVaR-constrained optimization is introduced. In the first stage a reference trajectory is planned using RRT*. In the second stage a novel

MPC method is introduced, where the risk of unsafety is measured using CVaR and limited to some risk tolerance level. In Chapter 3 the Wasserstein DR-MPC problem is formulated using CVaR constraint for collision avoidance and a set of reformulation procedures is proposed to resolve the infinite-dimensionality issue inherent in the DR-MPC problem. Besides, the probabilistic guarantee of limiting out-of-sample risk is discussed using the measure concentration inequality for Wasserstein ambiguity sets. In this chapter the advantage of the method over SAA, as well as its out-of-sample performance are demonstrated and analyzed on two mobile robots: (i) nonlinear car-like vehicle model and (ii) linearized quadrotor model. In Chapter 4, we apply Gaussian Process Regression (GPR) for obstacle motion prediction and combine the DR-MPC problem with GP. Here we also show the performance of the proposed method through numerical experiments on an on-road vehicle model.

Chapter 2

RISK-AWARE MOTION PLANNING AND CONTROL USING CVAR-CONSTRAINED OPTIMIZATION

2.1 Introduction

The safe operation of robots and autonomous systems in uncertain and dynamic environments, particularly those where humans are involved, has been regarded as a critical challenge. Moving objects and agents in robots' paths pose a major safety issue in practical environments. Unfortunately it is difficult to accurately predict an object's movement in many situations due to uncertainty in or incomplete knowledge about the object motion. The focus of this work is to develop an algorithmic tool for safe motion planning and decision-making in uncertain conditions by integrating prior knowledge about preferred motions of obstacles in a *risk-aware* manner.

Safe motion planning and control approaches under uncertainty have been extensively studied (see, for example, [2] and the references therein) and can be categorized as deterministic or stochastic. Deterministic methods often assume a bounded set of uncertainties and seek decision-making strategies that are robust with respect to this set of uncertainties. To obtain such robust solutions, algorithms using model predictive control [3], reachability [4], and safety funnels [5] have been developed, among

others. However, robust methods often lead to unnecessarily conservative solutions, particularly when the set of uncertainties is overestimated. This conservativeness may be alleviated by using adaptive online planning [6]. Stochastic methods can be used to systematically adjust the safety and conservativeness by incorporating probabilistic information about environments into decision making. Chance constraint-based methods have been one of the most popular stochastic tools in motion planning and control as chance constraints have the intuitive and practical role of limiting the probability of unsafe events. In particular, chance constraints have been widely used in conjunction with model predictive control [7, 8], optimal control [9], and sampling-based planning [10, 11]. Due to the nonconvexity of chance-constraints, however, these methods often use an approximation technique or a limited class of probability distributions and system dynamics to obtain a computationally tractable solution.[1]

Departing from chance constraint-based methods, we propose a safe motion planning and control approach using *conditional value-at-risk* (CVaR) constraints. The CVaR of a random loss is equal to the *conditional expectation* of the loss within the $(1-\alpha)$ worst-case quantile of the loss distribution [16, 17]. On the other hand, the value-at-risk (VaR) of a loss represents this quantile of the loss distribution, and thus it is closely related to chance constraints. The CVaR and VaR of a random variable X, the distribution of which has no probability atoms, have the following relationship: $\mathrm{CVaR}_\alpha(X) = \mathbb{E}[X \mid X \geq \mathrm{VaR}_\alpha(X)]$. CVaR constraints have several advantages over VaR or chance constraints, including the following: First, CVaR is a *coherent* risk measure, unlike VaR. According to Artzner *et al.* [18], a risk measure is said to be *coherent* if it satisfies the four axioms of translation invariance, subadditivity, positive homogeneity, and monotonicity. Majumdar and Pavone [19] claim that these four axioms should be satisfied for rational assessments of risk in robotics applications.

[1] There have been a few attempts to resolve the nonconvexity issue in using chance constraints for arbitrary probability distributions. In [12, 13], nonconvexity is handled by looking at higher order moments, while [14] considers moment-based ambiguity sets of distributions with given mean and covariance. [15] uses a sampling-based method for approximating arbitrary distribution by finite number of particles.

Second, CVaR constraints can distinguish tail events where losses exceed VaR, while chance constraints cannot [17]. Third, CVaR constraints are convex unlike most of the chance constraints [16].

To implement CVaR in a robot's risk assessment for safe motion planning, we propose a safety risk measure by extending our previous work [20] to the case of model predictive control (MPC) with randomly moving obstacles.[2] Specifically, the proposed safety risk measure represents the conditional expectation of the deviation within the $(1-\alpha)$ worst-case quantile of an associated safety loss distribution. With this safety risk measure, we develop a two-stage method for safe motion planning and control. In the first stage, a reference trajectory is generated by a fast sampling algorithm, such as RRT*, given the initial configuration of obstacles. However, as obstacles start to move randomly, this reference trajectory may no longer be safe to follow. Thus, in the second stage, a receding horizon controller is used to limit the risk of unsafety. This MPC problem is a CVaR-constrained stochastic program. Despite the convexity of CVaR constraints, this problem is nontrivial to solve because (i) each CVaR constraint involves bilevel optimization problems (one for CVaR and another for set distance), and (ii) the safe region is nonconvex in many practical cases. To overcome the first challenge, we reformulate the CVaR constraints without sacrificing optimality as more tractable expectation constraints. We then employ the sample average approximation (SAA) of the expectation constraints to further remove the minimization problem for computing a set distance. We show that the optimal value and solution obtained by the proposed SAA converge their original counterpart. The second issue, caused by nonconvex safe regions, can be addressed by recasting the MPC problem as a *linearly constrained* mixed integer convex program.

The remainder of this chapter is organized as follows: In Section 2.2, the problem

[2] As opposed to the dynamic programming approach in [20], we propose an MPC-based method and a different set of new reformulation procedures to obtain a *linearly constrained* mixed integer convex program, even in the presence of randomly moving obstacles.

setup is presented including stochastic obstacle movements. In Section 2.3, we describe the proposed motion planning and control method using CVaR-constrained optimization. In Section 2.4, we demonstrate the performance and utility of our method through simulations using a 12D quadrotor model in a 3D environment.

2.2 System and Obstacle Models

Consider a robotic vehicle, with dynamics that can be modeled as a discrete-time linear system of the form[3]

$$\xi(t+1) = A\xi(t) + Bu(t),$$
$$y(t) = C\xi(t) + Du(t),$$

where $\xi(t) \in \mathbb{R}^{n_\xi}$ and $y(t) \in \mathbb{R}^{n_y}$ are the system state and system output, respectively, and $u(t) \in \mathbb{R}^{n_u}$ is the control input at stage t. We assume that the output vector corresponds to the robotic vehicle's position in the n_y-dimensional configuration space. It is typical that a robotic system is subject to constraints on the system state and the control input:

$$\xi(t) \in \Xi, \quad u(t) \in \mathcal{U} \quad \forall t \geq 0. \tag{2.2.1}$$

We assume that $\Xi \subseteq \mathbb{R}^{n_\xi}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are convex sets.

The robot must navigate the space while avoiding L *randomly moving* rigid body obstacles. The region occupied by obstacle ℓ at stage t is represented by subset $\mathcal{O}_\ell(t)$ of $\mathbb{R}^{n_y}$. If an obstacle is not a convex polytope, we over-approximate it as a polytope and form its convex hull as shown in Fig. 2.1 [21].[4] After such approximation, we model the initial location of each obstacle as a convex polytope, i.e.,

$$\mathcal{O}_\ell(0) = \{\mathbf{x} \mid c_{j,\ell}^\top \mathbf{x} \leq d_{j,\ell},\ j = 1, \ldots, m_\ell\}, \tag{2.2.2}$$

where m_ℓ is the number of half-spaces defining the obstacle.

[3]The reformulation results in Section 2.3 are also valid for nonlinear systems. However, we focus on linear systems because the computational costs of nonlinear MPC are often prohibitive.

[4]This over-approximation is reasonable to plan safe paths.

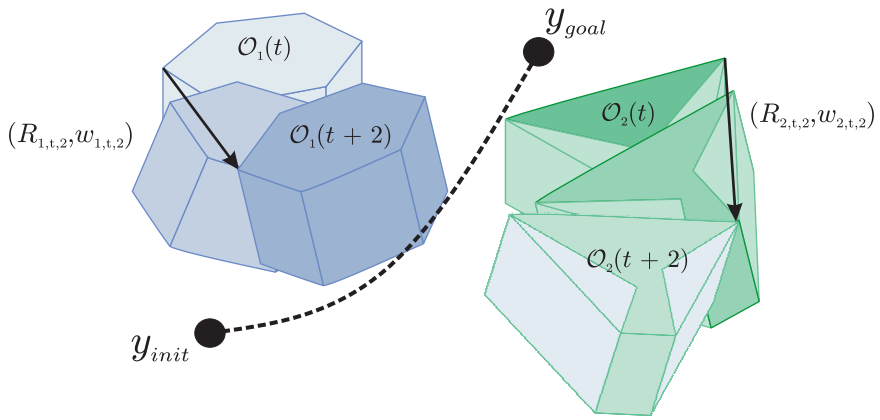


Figure 2.1: Robot configuration space with moving obstacles: $\mathcal{O}_1(t)$ is a convex obstacle, while $\mathcal{O}_2(t)$ is a nonconvex obstacle, approximated as its convex hull.

To formalize obstacle avoidance problems, we define the *safe region* as the complement to the region occupied by the obstacles. Specifically, let

$$\mathcal{Y}_\ell(t) := \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t) \quad \forall t \geq 0,$$

where $\mathcal{O}_\ell^o(t)$ denotes the interior of $\mathcal{O}_\ell$. Note that $\mathcal{Y}_\ell(t)$ is a closed set. For safety, the output is subject to the following constraint:

$$y(t) \in \mathcal{Y}_\ell(t) \quad \forall t \geq 0 \ \ \forall \ell.$$

The safe region also changes over time because the obstacle is moving randomly. The obstacle's movement between two stages is assumed to be represented by a linear transform, which includes the composition of rotations and translations, i.e.,

$$\mathcal{O}_\ell(t+k) = R_{\ell,t,k}\mathcal{O}_\ell(t) + w_{\ell,t,k},$$

where the nonsingular matrix $R_{\ell,t,k}$ is the product of random rotation matrices and $w_{\ell,t,k}$ is a random translation vector. Here, adding a vector w to a set $\mathcal{A}$ is defined by $\mathcal{A} + w := \{a + w \mid a \in \mathcal{A}\}$, multiplying a matrix R to a set $\mathcal{A}$ is defined by $R\mathcal{A} := \{R \cdot (a - c_\mathcal{A}) + c_\mathcal{A} \mid a \in \mathcal{A}\}$, where $c_\mathcal{A}$ is the centroid of $\mathcal{A}$. Accordingly,

$\mathcal{Y}_\ell(t)$ is shifted to $\mathcal{Y}_\ell(t+k) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t+k)$. It is straightforward to check

$$\mathcal{Y}_\ell(t+k) = R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}.$$

We assume that the region $\mathcal{O}_\ell(0)$ occupied by obstacle ℓ is known in advance.

2.3 CVaR-constrained Motion Planning and Control

The proposed risk-aware motion planning and control method consists of two stages:

1. fast reference trajectory planning, and
2. risk-constrained MPC.

In the first stage, a reference trajectory, which avoids $\mathcal{O}_\ell(0)\ \forall \ell$, is generated by using path-planning tools, such as RRT*. However, even if the robotic vehicle follows this reference trajectory, its safety is not guaranteed, due to the interference of randomly moving obstacles. To systematically limit the risk of collision, a risk-aware model predictive controller is employed in the second stage. The risk awareness is incorporated into the receding horizon optimization or MPC problem by using CVaR constraints. We develop a computationally tractable solution approach to the MPC problem through three reformulation procedures.

2.3.1 Reference Trajectory Planning

The first step is computing a collision-free path given the initial configuration of obstacles. For this work, we employ RRT* [22]. This particular tool efficiently searches nonconvex, high-dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally in a way that quickly reduces the expected distance between a randomly-chosen point and the tree. It provides an asymptotically optimal solution using tree rewiring and near neighbor search to improve the path quality. The tree starts from an initial state y_{init} and expands to find a path towards the goal state

y_{goal}, by randomly sampling the configuration space of obstacles in their initial positions and steering towards the random sample. However, the path generated by RRT* might not be possible to trace, given the dynamics of a robotic vehicle. In order to generate a traceable trajectory that takes into account robot dynamics, we perform kinodynamic motion planning based on RRT* [23]. The major difference from the baseline RRT* algorithm is that the vehicle dynamics is used for local steering to return a trajectory connecting two states while minimizing the distance between them. Compared to RRT [24], the main advantage of RRT* is that it provides an asymptotically optimal solution. Furthermore, in RRT* two major features are introduced: tree rewiring and near neighbor search, thus improving the path quality. Note, however, that any fast algorithms that generate a safe reference trajectory can be used in the first stage as a more sophisticated decision-making is performed in the second stage to limit the safety risk that a robot faces during its operation.

2.3.2 Safety Risk

As previously mentioned, the reference trajectory may no longer be safe once the obstacles start to move. To account for the randomness of obstacles' movement and develop a risk-aware decision-making tool, we mathematically define a notion of *safety risk* by combining set distance and CVaR.

Regarding the obstacle $\mathcal{O}_\ell^o(t)$, we measure the *loss of safety* at stage t as the distance between the robot's position $y(t)$ and the safe region $\mathcal{Y}_\ell(t) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t)$:

$$\operatorname{dist}(y(t), \mathcal{Y}_\ell(t)) := \min_{a \in \mathcal{Y}_\ell(t)} \|y(t) - a\|. \tag{2.3.1}$$

Obviously, it is desirable to drive the robot so that $\operatorname{dist}(y(t), \mathcal{Y}_\ell(t)) = 0$ for safety. However, due to the randomness of an obstacle's motion, imposing a hard constraint may lead to a very conservative decision. Instead of making such a deterministic guarantee, we systematically take into account the risk of unsafety to make the corresponding risk-aware decision depending upon the degree of risk that the robot can take.

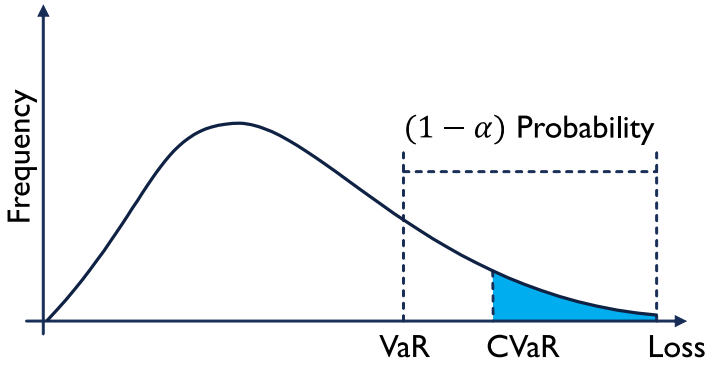


Figure 2.2: Conditional Value-at-Risk of a random loss.

Specifically, we evaluate the risk of system unsafety using the CVaR defined by

$$\mathrm{CVaR}_\alpha(X) := \min_{z\in\mathbb{R}} \mathbb{E}\left[z + \frac{(X-z)^+}{1-\alpha}\right], \tag{2.3.2}$$

where $(\boldsymbol{x})^+ = \boldsymbol{x}$ if $\boldsymbol{x} \geq 0$ and $(\boldsymbol{x})^+ = 0$ otherwise [17].

We quantify the *safety risk* of the robot at stage t as $\mathrm{CVaR}_\alpha[\mathrm{dist}(y(t), \mathcal{Y}_\ell(t))]$ concerned with random obstacle ℓ. This safety risk measures the conditional expectation of the distance between the robot position $y(t)$ and the safe region $\mathcal{Y}_\ell(t) = \mathbb{R}^{n_y} \setminus \mathcal{O}_\ell^o(t)$ within the $(1-\alpha)$ worst-case quantile of the safety loss distribution as illustrated in Fig. 2.2. In the following subsection, we formulate an MPC problem with constraints on the safety risk regarding all the obstacles:

$$\mathrm{CVaR}_\alpha[\mathrm{dist}(y(t), \mathcal{Y}_\ell(t))] \leq \delta_\ell \quad \forall \ell, \tag{2.3.3}$$

where $\delta_\ell \geq 0$ is a user-specified risk tolerance parameter for obstacle ℓ. In practice, δ_ℓ can be chosen as the maximum allowable expected deviation of a robot's position from the safe region $\mathcal{Y}_\ell$. On the other hand, another parameter α must be chosen by assessing the risk aversion of the user or the robot. For example, our Monte Carlo simulation results in Section 2.4 indicate that $\alpha = 0.95$ induces extremely risk-averse decisions that cause all sample trajectories to be safe, among 10,000 samples. Another way to determine δ_ℓ and α is to use the *mean-CVaR efficient frontier*, which represents

the possible tradeoff between minimizing the expected cost of motion planning and minimizing the safety risk (e.g., [25]).

2.3.3 Risk-Constrained Model Predictive Control

The main part of our risk-aware method is the following CVaR-constrained MPC problem (for stage t):[5]

$$\min_{\mathbf{u},\xi,\mathbf{y}} \quad J(\xi(t),\mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) \tag{2.3.4a}$$

$$\text{s.t.} \quad \xi_{k+1} = A\xi_k + Bu_k \tag{2.3.4b}$$

$$y_k = C\xi_k + Du_k \tag{2.3.4c}$$

$$\xi_0 = \xi(t) \tag{2.3.4d}$$

$$\xi_k \in \Xi,\ u_k \in \mathcal{U} \tag{2.3.4e}$$

$$\mathrm{CVaR}_\alpha[\mathrm{dist}(y_k, \mathcal{Y}_\ell(t+k))] \le \delta_\ell \ \ \forall \ell, \tag{2.3.4f}$$

where $\mathbf{u} := (u_0, \ldots, u_{K-1})$, $\xi := (\xi_0, \ldots, \xi_K)$, $\mathbf{y} := (y_0, \ldots, y_{K-1})$ and all the constraints must hold for $k = 0, \ldots, K-1$ except for $\xi_k \in \Xi$ in (2.3.4e) and (2.3.4c) which must hold for $k = 0, \ldots, K$. After computing an optimal $\mathbf{u}^\star$, only the first component $u_0^\star$ is chosen to be the control input at stage t, i.e., $u(t) := u_0^\star$. For the next stage, the MPC problem is defined in a receding horizon fashion and solved to obtain $u(t+1)$ given $\xi(t+1)$. Here, the constraints (2.3.4b) and (2.3.4c) account for the system state and output predicted in the MPC horizon by initializing ξ_0 as the current state $\xi(t)$. The state and input constraints (2.2.1) are specified in (2.3.4e). The stage-wise and terminal cost functions $r : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and $q : \mathbb{R}^{n_\xi} \to \mathbb{R}$ are chosen to evaluate the control performance. For example, given a reference trajectory, $\xi^{ref}(t)$,

[5] Our problem formulation and solution method is different from the one investigated by Singh *et al.* [26] as they consider uncertainties in model parameters (matrices A and B), whereas we consider uncertainties in obstacles' movement.

obtained by RRT*, the stage-wise cost function may be selected as follows:

$$r(\xi_k, u_k) := \left\| \xi_k - \xi^{ref}(t+k) \right\|_Q^2 + \|u_k\|_R^2 \tag{2.3.5}$$

$$q(\xi_K) := \left\| \xi_K - \xi^{ref}(t+K) \right\|_P^2 \tag{2.3.6}$$

where $Q \succeq 0$, $P \succeq 0$, $R \succ 0$ are the penalty weighing matrices, which penalize the running trajectory deviations and large input values, respectively.

Even when the cost function r is convex, it is nontrivial to solve the optimization problem (2.3.4) due to the CVaR constraint (2.3.4f). This constraint involves bilevel minimization, where the inner problem involves computing the distance $\mathrm{dist}(y(t), \mathcal{Y}_\ell(t))$ using (2.3.1) and the outer problem involves evaluating CVaR via (2.3.2). To develop a computationally tractable method, we first reformulate the MPC problem as the following stochastic program without loss of optimality:

Theorem 1. *Suppose that the stage-wise cost function r is continuous and the obstacles are convex polytopes as in* (2.2.2)*. Then, the set of optimal* $\mathbf{u}$*'s of the CVaR-constrained MPC problem* (2.3.4) *is equal to that of the following problem:*

$$\min_{\mathbf{u},\xi,\mathbf{y},\mathbf{z},\mathbf{h}} \; J(\xi(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) \tag{2.3.7a}$$

$$\text{s.t. } \mathbb{E}\left[z_{\ell,k} + \frac{(h_{\ell,k} - z_{\ell,k})^+}{1-\alpha} \right] \leq \delta_\ell \tag{2.3.7b}$$

$$y_k + R_{\ell,t,k} \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k} \in \mathcal{Y}_\ell(t+k) \tag{2.3.7c}$$

$$h_{\ell,k} \geq 0 \tag{2.3.7d}$$

$$z_{\ell,k} \in \mathbb{R} \tag{2.3.7e}$$

$$\text{(2.3.4b)–(2.3.4e)}, \tag{2.3.7f}$$

where the constraint (2.3.7c) *must hold for at least one* $j \in \{1, \ldots, m_\ell\}$*. All the remaining constraints must hold for all* $\ell = 1, \ldots, L$*, and* $k = 0, \ldots, K-1$ *except for* (2.3.4c) *and* $\xi_k \in \Xi$ *in* (2.3.7f) *which must hold for* $k = 0, \ldots, K$*.*

Proof. Let $J^\star$ and J' be the optimal value of the original and reformulated MPC problems, respectively. We first show that $J^\star = J'$. Let $(\mathbf{u}^\star, \xi^\star, \mathbf{y}^\star)$ be an optimal solution of the original MPC problem (2.3.4). By the definition of CVaR, $\mathrm{CVaR}_\alpha[\mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k))] = \min_{z_{\ell,k}\in\mathbb{R}} \mathbb{E}[z_{\ell,k} + \frac{(\mathrm{dist}(y_k, \mathcal{Y}_\ell(t+k)) - z_{\ell,k})^+}{1-\alpha}]$ for each ℓ and k. Let $z_{\ell,k}^\star \in \mathbb{R}$ be an optimal solution of the minimization problem on the right-hand side. Due to the constraint (2.3.4f), we have $\mathbb{E}[z_{\ell,k}^\star + \frac{(\mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k)) - z_{\ell,k}^\star)^+}{1-\alpha}] \leq \delta_\ell$. By setting $h_{\ell,k}^\star = \mathrm{dist}(y_k^\star, \mathcal{Y}_\ell(t+k))$, we see that $(y_k^\star, z_{\ell,k}^\star)$ satisfies the constraints (2.3.7b) and (2.3.7e) in the reformulated problem, since $h_{\ell,k}^\star$ satisfies the condition in (2.3.7c) and (2.3.7d). All the other constraints in the reformulated problem also appear in the original one. Thus, $(\mathbf{u}^\star, \xi^\star, \mathbf{y}^\star, \mathbf{z}^\star, \mathbf{h}^\star)$ satisfies all the constraints (2.3.7b)–(2.3.7f). This implies that $J' \leq J(\xi(t), \mathbf{u}^\star) = J^\star$.

We now show that $J' \geq J^\star$. Let $(\mathbf{u}', \xi', \mathbf{y}', \mathbf{z}', \mathbf{h}')$ be an optimal solution of the reformulated stochastic program. We first observe that

$$
\begin{aligned}
&\mathrm{CVaR}_\alpha[\mathrm{dist}(y_k', \mathcal{Y}_\ell(t+k))] \\
&= \min_{z\in\mathbb{R}} \mathbb{E}\left[z_{\ell,k} + \frac{(\mathrm{dist}(y_k', \mathcal{Y}_\ell(t+k)) - z_{\ell,k})^+}{1-\alpha} \right] \\
&\leq \mathbb{E}\left[z_{\ell,k}' + \frac{(h_{\ell,k}' - z_{\ell,k}')^+}{1-\alpha} \right] \leq \delta_\ell,
\end{aligned}
$$

where the first inequality is valid due to the constraint (2.3.7c) and (2.3.7d) and last inequality is valid due to the constraint (2.3.7b). Therefore, y_k' satisfies the CVaR constraint (2.3.4f) in the original problem. All the other constraints in (2.3.4) clearly hold with $(\mathbf{u}', \xi', \mathbf{y}')$. Thus, we have $J^\star \leq J(\xi(t), \mathbf{u}') = J'$.

We conclude that $J^\star = J(\xi(t), \mathbf{u}^\star) = J(\xi(t), \mathbf{u}') = J'$. Therefore, any optimal $\mathbf{u}^\star$ of the original MPC problem is optimal to the reformulated one, and conversely any optimal $\mathbf{u}'$ of the reformulated problem is optimal to the original one, as desired. □

To numerically solve the reformulated MPC problem (2.3.7), we need to compute the expected value in the constraint (2.3.7b). The constraint (2.3.7c) is equivalent to $y_k + R_{\ell,t,k}\frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k} \in R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}$ because $\mathcal{Y}_\ell(t+k) = R_{\ell,t,k}\mathcal{Y}_\ell(t) + w_{\ell,t,k}$.

One can rewrite the expectation as an integral with respect to a probability measure, then discretize a probability density to compute the integral. However, this approach involves a multi-dimensional integral, which is computationally demanding. In stochastic programming, a typical way to alleviate this issue is to employ *sample average approximation* (SAA). This approach approximates an expected constraint function or objective function using a sample average estimate, where the sample data are generated according to the underlying distribution. Specifically, given the sample $\{(\hat{R}^{(1)}_{\ell,t,k}, \hat{w}^{(1)}_{\ell,t,k}), \ldots, (\hat{R}^{(N_k)}_{\ell,t,k}, \hat{w}^{(N_k)}_{\ell,t,k})\}$ of $(R_{\ell,t,k}, w_{\ell,t,k})$, we approximate the constraint (2.3.7b) as

$$\frac{1}{N_k}\sum_{i=1}^{N_k}\left[z_{\ell,k} + \frac{(h^{(i)}_{\ell,k} - z_{\ell,k})^+}{1-\alpha}\right] \leq \delta_\ell. \tag{2.3.8}$$

Based on the SAA, we propose the following SAA-MPC problem:

$$\min_{\mathbf{u},\xi,\mathbf{y},\mathbf{z},\mathbf{h},\eta} J(\xi(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) \tag{2.3.9a}$$

$$\text{s.t. } \frac{1}{N_k}\sum_{i=1}^{N_k}\left[z_{\ell,k} + \frac{\eta^{(i)}_{\ell,k}}{1-\alpha}\right] \leq \delta_\ell \tag{2.3.9b}$$

$$y_k + \hat{R}^{(i)}_{\ell,t,k}\frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|}h^{(i)}_{\ell,k} \in \hat{R}^{(i)}_{\ell,t,k}\mathcal{Y}_\ell(t) + \hat{w}^{(i)}_{\ell,t,k}\ \forall i \tag{2.3.9c}$$

$$\eta^{(i)}_{\ell,k} \geq h^{(i)}_{\ell,k} - z_{\ell,k},\ \eta^{(i)}_{\ell,k} \geq 0,\ h^{(i)}_{\ell,k} \geq 0\ \forall i \tag{2.3.9d}$$

$$z_{\ell,k} \in \mathbb{R} \tag{2.3.9e}$$

$$\text{(2.3.4b)–(2.3.4e)}, \tag{2.3.9f}$$

where the constraint (2.3.9c) must hold for at least one $j \in \{1, \ldots, m_\ell\}$. All the constraints must hold for all $\ell = 1, \ldots, L$ and $k = 0, \ldots, K-1$ except for (2.3.4c) and $\xi_k \in \Xi$ in (2.3.9f) which must hold for $k = 0, \ldots, K$. Here an auxiliary real variable η is introduced to tackle the nonlinearity of (2.3.8).

To establish convergence properties of the proposed approximation method, we assume the following:

Assumption 1. *Suppose that*

1. *the stage-wise cost function r is continuous;*

2. *the set $\mathcal{U}$ is compact; and*

3. *for any optimal $\mathbf{u}^\star$ of the original problem, there exists a sequence $\mathbf{u}_N \in U_N$ such that $\mathbf{u}_N \to \mathbf{u}^\star$ with probability 1 as $N \to \infty$, where U_N is the set of feasible $\mathbf{u}$'s in* (2.3.9).

We then have the following convergence results regarding the optimal value and optimal solutions of the approximate problem (2.3.9):

Theorem 2. *Let $J^\star$ and $\mathbb{U}^\star$ be the optimal value and the set of optimal $\mathbf{u}$'s of the original problem* (2.3.4)*. Similarly, let J_N and $\mathbb{U}_N$ be the optimal value and the set of optimal $\mathbf{u}$'s of the approximate problem* (2.3.9)*, where $N := \min_{k\in\{0,\dots,K-1\}} N_k$. Then, under Assumption 1, we have*

$$J_N \to J^\star \quad \textit{and} \quad D(\mathbb{U}_N, \mathbb{U}^\star) \to 0 \quad \textit{as } N \to \infty,$$

where $D(A, B) := \sup_{\boldsymbol{x}\in A} \operatorname{dist}(\boldsymbol{x}, B)$ denotes the deviation of the set A from the set B.

Proof. We call the sample average approximation of (2.3.7) as SAA-MPC. Applying SAA to (2.3.7) replaces the constraint (2.3.7b) with (2.3.8). Let J_N^{SAA} and $\mathbb{U}_N^{\text{SAA}}$ denote the optimal value and the set of optimal $\mathbf{u}$ of the SAA-MPC problem. It is trivial that $J_N^{\text{SAA}} = J_N$ and $\mathbb{U}_N^{\text{SAA}} = \mathbb{U}_N$, thus (2.3.7) is equivalent to (2.3.9).

It now suffices to show that

$$J_N^{\text{SAA}} \to J^\star \quad \text{and} \quad D(\mathbb{U}_N^{\text{SAA}}, \mathbb{U}^\star) \to 0 \quad \text{as } N \to \infty.$$

Due to the continuity of the function f and Assumption 1, all the conditions of Theorems 5.3 and 5.5 in Shapiro et al. [27] are satisfied. Therefore, the result follows. □

Note that the proposed SAA method provides a provable *asymptotic guarantee* of satisfying the original CVaR constraints. Although our numerical simulation results in

Section 2.4 indicate that the SAA method always satisfies the risk constraints *for all time steps* even with a relatively small sample size ($N_k = 20$), the proposed method has no provable *finite-sample* guarantee of the risk constraints. If such a finite-sample guarantee is required, then the following advanced techniques can be used in conjunction with the proposed method. First, a robust version of SAA, which combines distributionally robust optimization and hypothesis testing of goodness-of-fit, can be used to obtain finite-sample guarantees in addition to retaining SAA's tractability and asymptotic properties [28]. Second, out-of-sample performance guarantees can be achieved in a probabilistic manner with a finite sample size by using a data-driven distributionally robust optimization framework [1, 29]. It is worth mentioning that the proposed reformulation procedures can also be used to enhance the computational tractability of the aforementioned techniques applied to motion planning. Specifically, an important feature of the approximate problem (2.3.9) is that it does not involve multiple-level optimization, unlike the original MPC problem (2.3.4). We can further reformulate (2.3.9) as a *linearly constrained* mixed integer convex program, when the cost function is convex, as proposed in the following subsection. A similar benefit may be obtained by using the proposed reformulation methods in conjunction with the advanced techniques for finite-sample guarantees.

2.3.4 Linearly Constrained Mixed Integer Convex Program

Due to the nonconvexity of the safe region $\mathcal{Y}_\ell(t)$, it is nontrivial to find an optimal solution to the problem (2.3.9). We now use the polyhedral characterization (2.2.2) of obstacles to reformulate the nonconvex constraints (2.3.9c) and recast the MPC problem as a *linearly constrained* mixed integer convex program (MICP). The proposed approach allows us to solve the MPC problem while retaining the full nonconvex constraints by using commercial MICP solvers. By continuous relaxation, we can also obtain *a posteriori* bounds on the gap between the optimal and approximate solution.

Due to the polyhedral characterization (2.2.2) of $\mathcal{O}_\ell(t)$, the constraint (2.3.9c) can

be written as the following disjunctive representation:

$$\bigvee_{j=1}^{m_\ell} c_{\ell,t,j}^\top \Big[(\hat{R}_{\ell,t,k}^{(i)})^{-1}(y_k - \hat{w}_{\ell,t,k}^{(i)}) + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)} \Big] \geq d_{\ell,t,j},$$

where m_ℓ is the number of half-spaces defining the obstacle, and $\bigvee$ denotes the logical disjunction operation. In order to tackle the 'OR' operation, the *Big-M reformulation* using binary variables is introduced as follows [30]:

$$\begin{aligned}
& c_{\ell,t,j}^\top \Big[(\hat{R}_{\ell,t,k}^{(i)})^{-1}(y_k - \hat{w}_{\ell,t,k}^{(i)}) + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)} \Big] \\
& \geq d_{\ell,t,j} - M_{j,\ell}\zeta_{j,\ell} \quad \forall j = 1, \ldots, m_\ell, && \text{(2.3.10a)} \\
& \zeta_{j,\ell} \in \{0, 1\} \quad \forall j = 1, \ldots, m_\ell, && \text{(2.3.10b)} \\
& \sum_{j=1}^{m_\ell} \zeta_{j,\ell} \leq m_\ell - 1, && \text{(2.3.10c)}
\end{aligned}$$

where $\ell = 1, \ldots, L$ and $k = 0, \ldots, K-1$. Here, $M_{j,\ell}$ is a constant greater than any possible value that the left-hand side of the inequality (2.3.10a) can have. By this inequality, when $\zeta_{j,\ell} = 0$, we have $c_{\ell,t,j}^\top [(\hat{R}_{\ell,t,k}^{(i)})^{-1}(y_k - \hat{w}_{\ell,t,k}^{(i)}) + \frac{c_{\ell,t,j}}{\|c_{\ell,t,j}\|} h_{\ell,k}^{(i)}] \geq d_{\ell,t,j}$. The inequality (2.3.10c) ensures that at least one of the binary variables is zero, and therefore (2.3.9c) holds at least for one $j \in 1, ..., m_\ell$.

By using this Big-M reformulation, the optimization problem (2.3.9) can be reformulated as follows:

$$\begin{aligned}
\min_{\mathbf{u},\xi,\mathbf{y},\mathbf{z},\eta,\zeta} \; & J(\xi(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) && \text{(2.3.11a)} \\
\text{s.t. } & \frac{1}{N_k} \sum_{i=1}^{N_k} \left[z_{\ell,k} + \frac{\eta_{\ell,k}^{(i)}}{1-\alpha} \right] \leq \delta_\ell \; \forall i && \text{(2.3.11b)} \\
& \eta_{\ell,k}^{(i)} \geq h_{\ell,k}^{(i)} - z_{\ell,k}, \; \eta_{\ell,k}^{(i)} \geq 0, \; h_{\ell,k}^{(i)} \geq 0 \; \forall i && \text{(2.3.11c)} \\
& z_{\ell,k} \in \mathbb{R}, && \text{(2.3.11d)} \\
& \text{(2.3.4b)–(2.3.4e)}, \; \text{(2.3.10a)–(2.3.10c)} && \text{(2.3.11e)}
\end{aligned}$$

where all the constraints must hold for all $\ell = 1, \ldots, L$ and $k = 0, \ldots, K-1$ except

for (2.3.4c) and $\xi_k \in \Xi$ in (2.3.11e) which must hold for $k = 0, \ldots, K$. This problem is a *linearly constrained* MICP when the cost function is convex.

Proposition 1. *Suppose that* $r : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}$ *and* $q : \mathbb{R}^{n_\xi} \to \mathbb{R}$ *are convex. Then, the reformulated MPC problem* (2.3.11) *is a linearly constrained mixed integer convex program.*

The linearly constrained MICP problem can be solved by using several methods, such as branch-and-bound [31], outer approximation [32] and polyhedral approximation [33]. State-of-the-art solvers like Bonmin [34], SCIP [35], and Artelys Knitro [36] support other techniques, such as cutting planes, that accelerate the solution-search process.

2.4 Numerical Experiments

In this section, we present simulation results that demonstrate the performance of the proposed approach. Consider a quadrotor that aims to travel from a starting point y_{init} to a goal point y_{goal} in a 3D space. Dynamic obstacles interfere with the quadrotor's possible paths. The position of a 6 DOF quadrotor can be expressed in the space of (x^r, y^r,z^r,ϕ^r, θ^r, ψ^r). The first three variables—x^r, y^r, and z^r—represent the distances of the quadrotor's center of mass along the X, Y and Z axes, respectively, from an Earth-fixed frame, whereas ϕ^r, θ^r, and ψ^r are the three Euler angles that represent the orientation of the quadrotor. Note that ϕ^r, θ^r, and ψ^r are the roll, pitch, and yaw angles about the X, Y, and Z axes, respectively. The dynamics of the quadrotor can then be modeled as

$$\ddot{x}^r = -g\theta, \quad \ddot{y}^r = g\phi^r, \quad \ddot{z}^r = -\frac{l_Q}{m_Q}u_1^r,$$
$$\ddot{\phi}^r = \frac{1}{I_{xx}}u_2^r, \quad \ddot{\theta}^r = \frac{l_Q}{I_{yy}}u_3^r, \quad \ddot{\psi}^r = \frac{l_Q}{I_{zz}}u_4^r,$$

where m_Q is the quadrotor's mass, g is the gravitational acceleration, and I_{xx}, I_{yy}, and I_{zz} are the area moments of inertia about the principle axes in the body frame. Accordingly, the state space model of the quadrotor has 12 states, including its position

and orientation in 3D space, as well as the corresponding velocities and rates. The output is chosen as the (x^r, y^r, z^r) position of the quadrotor. The rotors can be operated within a specific range of velocity. Thus, the input feasible set $\mathcal{U} := \{\boldsymbol{u} \in \mathbb{R}^4 \mid u_{\min} \leq \boldsymbol{u} \leq u_{\max}\}$ has been selected according to the chosen motor specifications. The set $\Xi := \{\boldsymbol{\xi} \in \mathbb{R}^1 2 \mid -\pi \leq \phi^r \leq \pi\ ,\ -\frac{\pi}{2} \leq \theta^r \leq \frac{\pi}{2},\ -\pi \leq \psi^r \leq \pi\}$ has been selected to limit the angles to avoid kinematic singularity. The following parameters were employed throughout simulation [37]: $m = 0.65\ kg,\ l = 0.23\ m,\ I_{xx} = 0.0075\ kg \cdot m^2,\ I_{yy} = 0.0075\ kg \cdot m^2,\ I_{zz} = 0.013\ kg \cdot m^2,\ g = 9.81\ ms^2$.

As the first step, RRT* is used to generate a safe reference trajectory given the initial configuration of the obstacles. The quadrotor starts tracking the reference trajectory by using the receding horizon controller obtained by solving the MICP (2.3.11) using Gurobi 8.1.0. The MPC horizon and the number of time steps are selected as $K = 15$ and $T = 50$, respectively, and the weights in the stage-wise cost function (2.3.5) are chosen as $Q = P = I$ and $R = 0.01I$.

2.4.1 Effect of Confidence Level

In the first scenario, two randomly translating obstacles are interfering with the quadcopter's initially feasible optimal path. To demonstrate the the effect of confidence level, we consider three different cases with $\alpha = 0.1, 0.5, 0.95$ and $\delta = 0.04$. The resulting trajectories are shown in Fig. 2.3. The simulations were performed with $N_k = 20$ samples, where the random movement of each obstacle in each stage $t + k$ is uniformly distributed over $[-0.4, 0.4]^3$ and summed up from 1 to k to form $\hat{w}^{(i)}_{\ell,t,k}$.

In early stages (before $t = 9$), for all α's, the quadrotor deviates from its reference trajectory, even if it is not close to any obstacles, as shown in Fig. 2.3 (a). This is because the robot should always satisfy all the CVaR constraints within the prediction horizon. Until $t = 19$, the quadrotor is close to the obstacles, as shown in Fig. 2.3 (b). With the $\alpha = 0.95$, the robot makes sure that it is a safe margin away from the obstacles, so as not to violate the CVaR constraints within the prediction horizon. Such

Table 2.1: Computation time and operation cost for Scenario 1 with $N_k = 20$ and $\delta_\ell = 0.04$

α	0.1	0.5	0.95
Cost	5228.0	1253.2	1323.3
Time (sec)	697.44	533.34	175.65

a deviation also occurs when $\alpha = 0.5, 0.1$, but with less magnitude than in the previous case. Fig. 2.3 (c) shows the quadrotor's position after it passes the obstacles at $t = 27$. The quadrotor starts to follow the reference trajectory without deviation. The complete trajectories are shown in Fig. 2.3 (d).

To empirically demonstrate the safety guarantees in the proposed SAA method, we test if the CVaR value estimated by SAA and Monte Carlo simulations satisfies the original risk constraint. We first calculate SAA-CVaR by using (2.3.2) with the trajectories generated by SAA-MPC with $N_k = 20$ samples. In addition, given the risk-aware controller, we performed Monte Carlo simulations using 10,000 new samples generated at each time step to compute MC-CVaR. The results for the first obstacle for different confidence levels at $t = [10, 25]$ are summarized in Fig. 2.4. In other stages, both of these values are 0, as the vehicle is far from the obstacle. These results demonstrate the capability of this method in adjusting the safety and conservativeness. It can be noticed that as the confidence level α increases, the robot's risk aversion increases and this encourages conservative decisions that induce high control costs $\sum_{t=0}^{T} J(\xi(t), u(t))$ as shown in Table 2.1. For all confidence levels, the SAA-CVaR and MC-CVaR are always strictly less than the risk tolerance level δ. This confirms that the safety risk constraint (2.3.4f) is satisfied even when the proposed SAA method uses small number of samples ($N_k = 20$). The computational time also depends on the confidence level.

In order to compare the proposed method with existing chance-constrained approach, we also implemented MPC problem with sampling-based chance-constraints

Table 2.2: Computation time and operation cost for Scenario 2 with $\alpha = 0.99$ and $\delta = 0.02$

N_k	50	90	100
Cost	1589.57	1726.47	1737.16
Time (sec)	799.72	2167.33	2580.51

using the results from [38] and [15]. Specifically, the following chance constraint is used: $\Pr[\text{dist}(y_k, \mathcal{Y}_{\ell,t,k}) = 0] \geq 0.95$. For comparison, we use $\delta = 0$ and $\alpha = 0.95$ in the CVaR constraint.[6] Fig. 2.5 shows the trajectories obtained by the chance-constrained approach and our method. The result obtained by CVaR-constrained optimization is safer than the one obtained by chance-constrained approach. This is explained by the fact that CVaR takes into account the tail events, while chance-constraint performs decision making only for the $(1-\alpha)$-worst case quantile. In fact, $\text{CVaR}_\alpha(X) \geq \text{VaR}_\alpha(X)$ by definition, and thus CVaR induces a safer decision-making than VaR or chance-constraints.

2.4.2 Effect of Sample Size

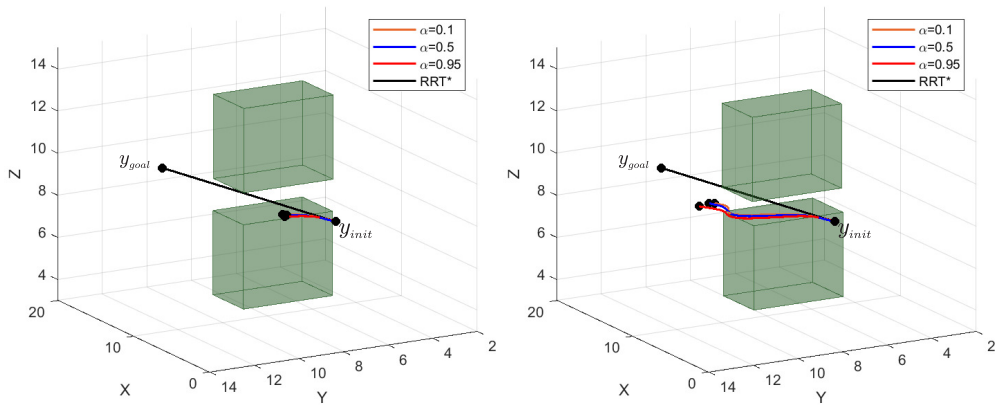
In the second scenario, the rotational motion of the first obstacle is uniformly distributed over $[-0.1, 0.1]^3$. In addition to the random rotation, a deterministic translation is considered so that the obstacle obstructs the nominal trajectory as time goes. On the other hand, both of the obstacles present random translational motion, sampled from a Gaussian distribution with mean $\mu = 0$ and covariance $\sigma = 0.01$ in each stage. Fig. 2.6 illustrates the simulation results for different numbers of samples to observe their effect on decision making. In all cases, the risk tolerance and confidence levels were chosen as $\delta = 0.02$ and $\alpha = 0.95$, respectively. It can be seen from Table 2.2 that the computation time increases with the number of samples, as expected. In addi-

[6]The risk tolerance level $\delta = 0$ induces the CVaR constraint to become a hard (deterministic) constraint.

tion, we examine the effect of N_k on the total cost and CVaR. As shown in Table 2.2, the cost value converges as the number of sample data increases, which can also be seen in Fig. 2.6, where the trajectories generated for $N_k = 100$ and $N_k = 110$ are the same. This convergence is consistent with Theorem 2. Regarding SAA-CVaR and MC-CVaR, both of them are equal to 0 for all N_k's because that big α and small δ are used. The second scenario requires more random samples than the first scenario because of two factors: the obstacles configuration and the existence of additional rotational uncertainties.

2.5 Conclusions

A risk-aware motion planning and control approach has been presented for robots operating in uncertain and dynamic environments. Our strategy consists of two stages: (i) generating a safe reference trajectory by using RRT*, and (ii) utilizing CVaR to assess safety risks and design a CVaR constrained receding horizon controller to track the reference trajectory. A computationally tractable solution to the MPC problem has been developed using the following three procedures. First, we reformulated the CVaR constraints without loss of optimality. Second, we proposed a convergent SAA method to completely remove multi-level optimization. Third, the nonconvexity of safe regions was addressed by recasting the MPC problem as a linearly constrained mixed integer convex program. Simulations using a quadrotor in a 3D environment demonstrate this method's capability to systematically adjust the safety and conservativeness in motion planning and control, as well as the effect of sample size on risk-aware decision-making.



(a) $t = 9$ (b) $t = 19$

(c) $t = 27$ (d) $t = 50$

Figure 2.3: Generated quadrotor trajectories at difference stages with confidence levels $\alpha = 0.1, 0.5, 0.95$.

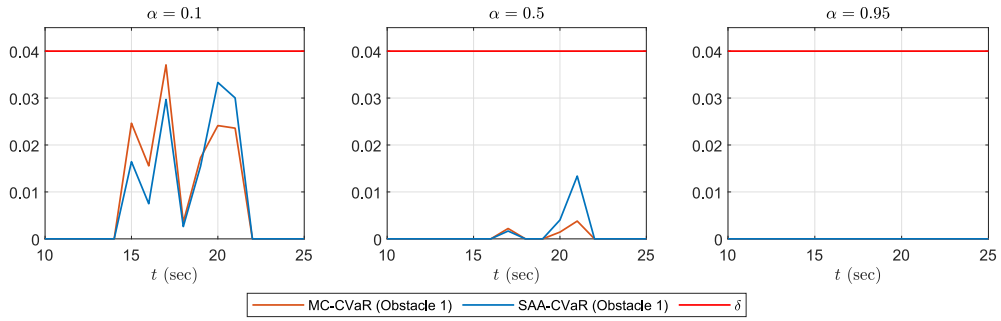


Figure 2.4: Comparison of SAA-CVaR and MC-CVaR to the threshold δ for $\alpha = 0.1, 0.5, 0.95$.

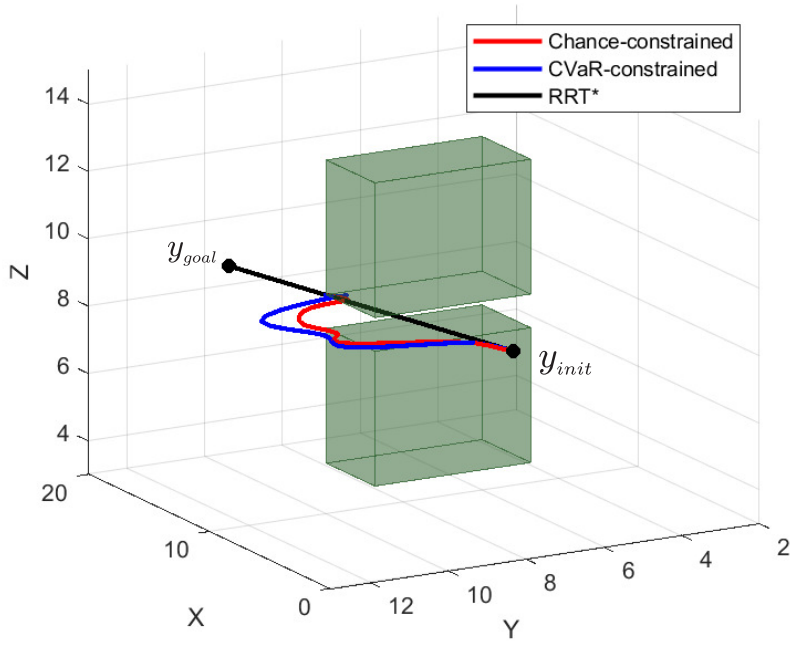
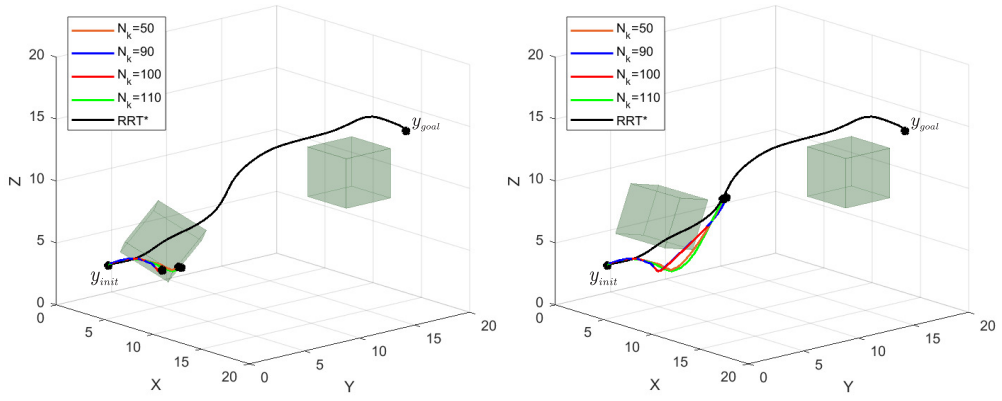


Figure 2.5: Comparison between the chance-constrained and CVaR-constrained motion control.



(a) $t = 13$ (b) $t = 21$

(c) $t = 40$ (d) $t = 50$

Figure 2.6: Generated quadrotor trajectories for different number of samples $N_k = 50, 90, 100, 110$.

Chapter 3

WASSERSTEIN DISTRIBUTIONALLY ROBUST MPC

3.1 Introduction

Safety is one of the most fundamental challenges in the operation of mobile robots and autonomous vehicles in practical environments, which are uncertain and dynamic. In particular, the unexpected movement of objects and agents often jeopardizes the collision-free navigation of mobile robots. Unfortunately, predicting an object's motion is a challenging task in many circumstances due to the lack of knowledge about the object's possibly uncertain dynamics. To gather information about an obstacle's uncertain movement, it is typical to use (historical) sample data of its motion. The main goal of this work is to develop an optimization-based method for risk-aware motion planning and control by incorporating data about moving obstacles into the robot's decision-making in a *distributionally robust* manner..

Several risk-sensitive decision-making methods have been proposed for robots to avoid obstacles in uncertain environments. Chance-constrained methods are among the most popular approaches, as they can be used to directly limit the probability of collision. Because of their intuitive and practical role, chance constraints have been extensively used in sampling-based planning [10, 11, 14] and model predictive control (MPC) [7, 8]. However, it is computationally challenging to handle a chance constraint

due to its nonconvexity. This often limits the admissible class of probability distributions and system dynamics and/or requires an undesirable approximation. To resolve the issue of nonconvexity, a few theoretical and algorithmic tools have been developed using a particle-based approximation [15] and semidefinite programming formulation [39], among others. Another approach is to use a convex risk measure, which is computationally tractable. In particular, *conditional value-at-risk* (CVaR) has recently drawn a great deal of interest in motion planning and control [19,26,40,41]. The CVaR of a random loss represents the conditional expectation of the loss within the $(1-\alpha)$ worst-case quantile of the loss distribution, where $\alpha \in (0,1)$ [17]. As claimed in [19], CVaR is suitable for rational risk assessments in robotic applications because of its *coherence* in the sense of Artzner *et al.* [18]. In addition to its computational tractability, CVaR is capable of distinguishing the worst-case tail events, and thus it is effective to take into account rare but unsafe events. To enjoy these advantages, we adopt CVaR to measure the risk of unsafety.

The performance of such risk-aware motion control tools critically depends on the quality of information about the probability distribution of underlying uncertainties, such as an obstacle's random motion. If a poorly estimated distribution is used, it may cause unwanted behaviors of the robot, leading to a collision. One of the most straightforward ways to estimate the probability distribution is to collect the sample data of an obstacle's movement and construct an empirical distribution. The use of an empirical distribution is equivalent to a sample average approximation (SAA) of the stochastic programs [27]. Although SAA is quite effective with asymptotic optimality, it does not have a finite-sample guarantee of satisfying risk constraints. In our previous work using SAA, it was empirically observed that risk constraints are likely to be violated when the sample size is very small [41].

To account for this issue of limited distributional information, we seek an efficient risk-aware motion control method that is robust against distribution errors. Our method is based on *distributionally robust optimization* (DRO), which is employed to solve

a stochastic program in the face of the worst-case distribution drawn from a given set, called the *ambiguity set* [42–44]. In this work, we use the Wasserstein ambiguity set, a statistical ball that contains all the probability distributions whose *Wasserstein distance* from an empirical distribution is no greater than a certain radius [1, 45, 46]. The Wasserstein ambiguity set has several salient features, such as providing a non-asymptotic performance guarantee and addressing the closeness between two points in the support, unlike other statistical distance-based ambiguity sets (e.g., using phi-divergence) [46–48]. The proposed motion control method is robust against obstacle movement distribution errors characterized by the Wasserstein ambiguity set.

The contributions of this work can be summarized as follows. First, a novel model predictive control (MPC) method is proposed to limit the risk of unsafety through CVaR constraints that must hold for any perturbation of the empirical distribution within the Wasserstein ambiguity set. Thus, the resulting control decision is guaranteed to satisfy the risk constraints for avoiding randomly moving obstacles in the presence of allowable distribution errors. Moreover, the proposed method provides a finite-sample probabilistic guarantee of limiting *out-of-sample risk*, meaning that the risk constraints are satisfied with probability no less than a certain threshold even when evaluated with new sample data chosen independently of the training data. Second, for computational tractability, we reformulate the distributionally robust MPC (DR-MPC) problem, which is infinite-dimensional, into a finite-dimensional nonconvex optimization problem. The proposed reformulation procedure is developed using modern DRO techniques based on the Kantorovich duality principle [1]. Third, a spatial branch-and-bound (sBB) algorithm is designed with McCormick relaxation to address the issue of nonconvexity. The proposed algorithm finds a globally optimal control action in the case of affine system dynamics and output equations. The performance and utility of the proposed method are demonstrated through two simulation studies, one with a nonlinear car-like vehicle model and another with a linearized quadrotor model. The results of numerical experiments confirm that, even when the sample size is small, the

proposed DR-MPC method can successfully avoid randomly moving obstacles with a guarantee of limiting out-of-sample risk, while its SAA counterpart fails to do so.

The rest of this chapter is organized as follows. In Section 3.2, the problem setup and the obstacles' model are introduced. In Section 3.3, Wasserstein DR-MPC problem is formulated using CVaR constraints for collision avoidance. Also, a set of reformulation procedures is proposed to resolve the infinite-dimensionality issue inherent in the DR-MPC problem. In Section 3.4, the probabilistic guarantee of limiting out-of-sample risk is discussed using the measure concentration inequality for Wasserstein ambiguity sets. Finally, the simulation results are presented and analyzed in Section 3.5.

3.2 System and Obstacle Models

In this chapter, we consider a mobile robot, which can be modeled by the following discrete-time dynamical system:

$$\begin{aligned} \xi(t+1) &= f(\xi(t), u(t)), \\ y(t) &= h(\xi(t), u(t)), \end{aligned}$$

where $\xi(t) \in \mathbb{R}^{n_\xi}$, $u(t) \in \mathbb{R}^{n_u}$ and $y(t) \in \mathbb{R}^{n_y}$ are the system state, the control input, and the system output, respectively. In general, $f : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_\xi}$ and $h : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ are nonlinear functions, representing the system dynamics and the output mapping, respectively. We regard the output as the robot's current position in the n_y-dimensional configuration space. Typical robotic systems operate under some state and control constraints:

$$\xi(t) \in \Xi, \quad u(t) \in \mathcal{U}.$$

We assume that $\Xi \subseteq \mathbb{R}^{n_\xi}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are convex sets.

To formulate a collision avoidance problem, we consider *L randomly moving* rigid body obstacles that the robotic vehicle has to avoid while navigating the configuration

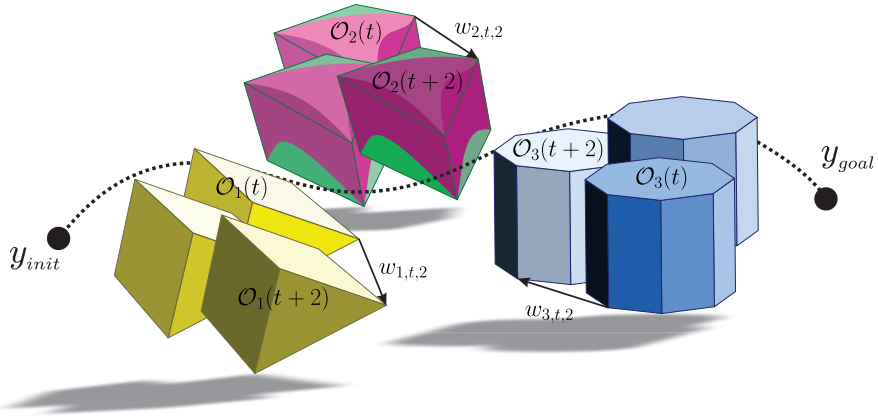


Figure 3.1: Robot configuration space with randomly moving obstacles.

space. Let the region occupied by the obstacle ℓ at stage t be denoted by $\mathcal{O}_\ell(t) \subset \mathbb{R}^{n_y}$. If $\mathcal{O}_\ell(t)$ is not a convex polytope, we over-approximate it as a polytope and choose its convex hull as illustrated in the second obstacle in Fig. 3.1 in a similar way to 2.2.

In this chapter we consider obstacles with random translational motion between two stages:

$$\mathcal{O}_\ell(t+k) = \mathcal{O}_\ell(t) + w_{\ell,t,k}.$$

An example of obstacles' movements is illustrated in Fig. 3.1. Here, the sum of a set $\mathcal{A}$ and a vector w is defined by adding w to all elements of $\mathcal{A}$, i.e. $\mathcal{A} + w := \{a + w \mid a \in \mathcal{A}\}$.

It is straightforward to see that the safe region has the same translational motion

$$\begin{aligned}
\mathcal{Y}_\ell(t+k) &= \mathbb{R}^{n_y} \setminus \mathcal{O}^o_\ell(t+k) \\
&= \{x \in \mathbb{R}^{n_y} \mid x \notin \mathcal{O}^o_\ell(t+k)\} \\
&= \{x + w_{\ell,t,k} \in \mathbb{R}^{n_y} \mid x \notin O^o_\ell(t)\} \\
&= \mathcal{Y}_\ell(t) + w_{\ell,t,k}.
\end{aligned}$$

where $w_{\ell,t,k}$ is a random translation vector in $\mathbb{R}^{n_y}$.

3.3 Wasserstein Distributionally Robust MPC

Computing safety risk in (2.3.3) requires information about the probability distribution of $w_{\ell,t,k}$. However, the exact probability distribution is unknown in practice, and obtaining a reliable distribution is a challenging task. In most cases, we only have a limited amount of sample data generated from the underlying distribution. Probably the simplest way to incorporate the available data into the motion control problem is to employ an empirical distribution as in SAA of stochastic programs [27]. Specifically, given sample data $\{\hat{w}_{\ell,t,k}^{(1)}, \ldots, \hat{w}_{\ell,t,k}^{(N_k)}\}$ of $w_{\ell,t,k}$, the empirical distribution is defined as

$$\nu_{\ell,t,k} := \frac{1}{N_k} \sum_{i=1}^{N_k} \boldsymbol{\delta}_{\hat{w}_{\ell,t,k}^{(i)}}, \tag{3.3.1}$$

where $\boldsymbol{\delta}_w$ is the Dirac delta measure concentrated at w. However, this empirical distribution is not capable of reliably estimating the safety risk, particularly when the sample size N_k is small. This fundamental limitation results in unsafe decision-making without respecting the original risk constraint. Thus, the approach of using empirical distributions may lead to damaging collisions as the safety risk is poorly assessed.

To resolve the issue of unreliable distribution information, we take a DRO approach. Instead of using the risk constraint (2.3.3), we limit the safety risk evaluated under the worst-case distribution of $w_{\ell,t,k}$ lying in a given set $\mathbb{D}_{\ell,t,k}$, called an *ambiguity set*. More precisely, we impose the following *distributionally robust risk constraint*:

$$\sup_{\mu_{\ell,t,k} \in \mathbb{D}_{\ell,t,k}} \mathrm{CVaR}_{\alpha}^{\mu_{\ell,t,k}}[\mathrm{dist}(y_k, \mathcal{Y}_\ell(t+k))] \leq \delta_\ell \quad \forall \ell.$$

By limiting the worst-case risk value that the robot can bear, the resulting control action is robust against distribution errors characterized by the ambiguity set. In this work, the ambiguity set is chosen as the following statistical ball centered at the empirical distribution (3.3.1) with radius $\theta > 0$:

$$\mathbb{D}_{\ell,t,k} := \{\mu \in \mathcal{P}(\mathbb{W}) \mid W(\mu, \nu_{\ell,t,k}) \leq \theta\}, \tag{3.3.2}$$

where $\mathcal{P}(\mathbb{W})$ denotes the set of Borel probability measures on the support $\mathbb{W} \subseteq \mathbb{R}^{n_y}$. Here, the Wasserstein distance (of order 1) $W(\mu, \nu)$ between μ and ν represents the

minimum cost of redistributing mass from one measure to another using a small non-uniform perturbation, and is defined by

$$W(\mu,\nu) := \min_{\kappa\in\mathcal{P}(\mathbb{W}^2)} \Big\{ \int_{\mathbb{W}^2} \|w - w'\| \, \mathrm{d}\kappa(w,w') \mid \Pi^1\kappa = \mu, \Pi^2\kappa = \nu \Big\},$$

where $\Pi^i\kappa$ denotes the ith marginal of the transportation plan κ for $i = 1, 2$, and $\|\cdot\|$ is an arbitrary norm on $\mathbb{R}^{n_y}$.

From (3.3.2) it is straightforward to see that by adjusting the radius θ, one can control the degree of conservatism of the underlying optimization problem. If the radius drops to zero, then the ambiguity set shrinks to a singleton that contains only the nominal distribution, in which case the distributionally robust problem reduces to an ambiguity-free stochastic program.

It is worth mentioning that other types of ambiguity sets can be chosen in the proposed DR-MPC formulation. A popular choice in the literature of DRO is moment-based ambiguity sets [42–44]. However, such ambiguity sets are often overly conservative and require a large sample size to reliably estimate moment information. Statistical distance-based ambiguity sets have also received considerable interest, by using phi-divergence [47] and Wasserstein distance [1, 45, 46, 49], among others. However, unlike other statistical distance-based ones, the Wasserstein ambiguity set contains a richer set of relevant distributions, and the corresponding Wasserstein DRO provides a superior finite-sample performance guarantee [1]. These desirable features play an important role in the proposed motion control tool.

Model Predictive Control (MPC) is one of the advanced techniques for mobile robot control. One of the advantages of using MPC is that it gives an opportunity to add constraints and make decision within the allowable set. Thus, it is reasonable to consider an MPC with given robot vehicle dynamics and a set of constraints and formulate formulate the risk-aware motion control problem as the following *Wasserstein*

distributionally robust MPC (DR-MPC) problem:[1]

$$\inf_{\mathbf{u},\xi,\mathbf{y}} \; J(\xi(t),\mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k,u_k) + q(\xi_K) \tag{3.3.3a}$$

$$\text{s.t. } \xi_{k+1} = f(\xi_k,u_k) \tag{3.3.3b}$$

$$y_k = h(\xi_k,u_k) \tag{3.3.3c}$$

$$\xi_0 = \xi(t) \tag{3.3.3d}$$

$$\xi_k \in \Xi \tag{3.3.3e}$$

$$u_k \in \mathcal{U} \tag{3.3.3f}$$

$$\sup_{\mu_{\ell,t,k}\in\mathbb{D}_{\ell,t,k}} \mathrm{CVaR}_\alpha^{\mu_{\ell,t,k}}[\mathrm{dist}(y_k,\mathcal{Y}_\ell(t+k))] \le \delta_\ell, \tag{3.3.3g}$$

where $\mathbf{u} := (u_0,\ldots,u_{K-1})$, $\xi := (\xi_0,\ldots,\xi_K)$, $\mathbf{y} := (y_0,\ldots,y_K)$. The constraints (3.3.3b) and (3.3.3f) should be satisfied for $k = 0,\ldots,K-1$, the constraints (3.3.3c) and (3.3.3e) should hold for $k = 0,\ldots,K$, and the constraint (3.3.3g) is imposed for $k = 1,\ldots,K$ and $\ell = 1,\ldots,L$. Here, the stage-wise cost function $r : \mathbb{R}^{n_\xi}\times\mathbb{R}^{n_u} \to \mathbb{R}$ and the terminal cost function $q : \mathbb{R}^{n_\xi} \to \mathbb{R}$ are chosen to penalize the deviation from the reference trajectory $\xi^{ref}(t)$ generated in Section 2.3.1 and to minimize the control effort. Specifically, we set

$$J(\xi(t),\mathbf{u}) := \|\xi_K - \xi^{ref}(t+K)\|_P^2 + \sum_{k=0}^{K-1} \|\xi_k - \xi^{ref}(t+k)\|_Q^2 + \|u_k\|_R^2,$$

where $Q \succeq 0$, $R \succ 0$ are the weight matrices for state and input, respectively, and $P \succeq 0$ is chosen in a way to ensure stability. The constraints (3.3.3b) and (3.3.3c) account for the system state and output predicted in the MPC horizon when ξ_0 is initialized as the current state $\xi(t)$, and (3.3.3e) and (3.3.3f) are the constraints on system state and control input, respectively. The distributionally robust risk constraint

[1] Our problem formulation and solution method is different from the one studied by Coulson *et al.* [50] as they consider uncertainties in systems, whereas we consider uncertainties in obstacles' motions. Dynamic programming approaches to distributionally robust optimal control problems have also been studied in [51–54].

is specified in (3.3.3g), which is the most important part in this problem for safe motion control with limited distribution information.

The Wasserstein DR-MPC problem is defined and solved in a receding horizon manner. Once an optimal solution $\mathbf{u}^\star$ is obtained given the current state $\xi(t)$, the first component $u_0^\star$ of $\mathbf{u}^\star$ is selected as the control input at stage t, i.e., $u(t) := u_0^\star$. Unfortunately, it is challenging to solve the Wasserstein DR-MPC problem due to the distributionally robust risk constraint (3.3.3g). This risk itself involves an optimization problem, which is infinite-dimensional. To alleviate the computational difficulty, we reformulate the Wasserstein DR-MPC problem in a tractable form and propose efficient algorithms for solving the reformulated problem in the following sections.

To develop a computationally tractable approach to solving the Wasserstein DR-MPC problem, we propose a set of reformulation procedures. For ease of exposition, we suppress the subscripts in the DR-risk constraint (3.3.3g) and consider

$$\sup_{\mu\in\mathbb{D}} \mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y, \mathcal{Y} + w)] \le \delta. \tag{3.3.4}$$

3.3.1 Distance to the Safe Region

The first step is to derive a simple expression for the loss of safety, $\mathrm{dist}(y, \mathcal{Y} + w)$. Recall that the region occupied by an obstacle is represented as a convex polytope (via over-approximation if needed), i.e.,

$$\mathcal{O} = \{y \mid c_j^\top y \le d_j, j = 1, \ldots, m\}$$

for some $c_j \in \mathbb{R}^{n_y}$ and $d_j \in \mathbb{R}$. Since $\mathcal{Y} = \mathbb{R}^{n_y} \setminus \mathcal{O}^o$, the corresponding safe region can be expressed as the union of half spaces, i.e.,

$$\mathcal{Y} := \bigcup_{j=1}^{m} \{y \mid c_j^\top y \ge d_j\}. \tag{3.3.5}$$

From (3.3.5) we see that the safe region is a union of halfspaces, resulting in the next lemma.

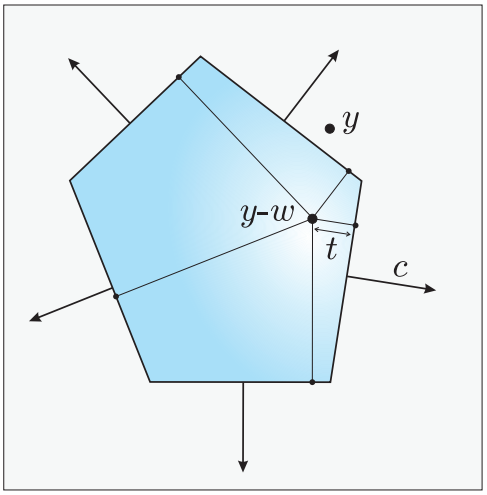


Figure 3.2: Illustration of the distance to the union of halfspaces.

Lemma 1. *Suppose that the safe region is given by* (3.3.5)*. Then, the loss of safety* (2.3.1) *can be expressed as*

$$\operatorname{dist}(y, \mathcal{Y} + w) = \left[\min_{j=1,\dots,m} \frac{d_j - c_j^\top (y - w)}{\|c_j\|_2} \right]^+.$$

Proof. First, we let

$$\mathcal{Y}_j := \{y \mid c_j^\top y \geq d_j\}.$$

Then, using the property that $\mathcal{Y}_j + w = \{y \mid c_j^\top (y - w) \geq d_j\}$, the distance between y and each halfspace can be represented by

$$\operatorname{dist}(y, \mathcal{Y}_j + w) = \inf_t \{\|t\|_2 \mid c_j^\top (y - t - w) \geq d_j\}. \tag{3.3.6}$$

This equality is illustrated in Fig. 3.2. To derive the dual of the optimization problem in (3.3.6), we first find the Lagrangian as

$$L(t, \lambda) = \|t\|_2 + \lambda \big[d_j - c_j^\top (y - t - w)\big].$$

The corresponding dual function is obtained by

$$\begin{aligned} g(\lambda) &= \min_t \big\{ \|t\|_2 + \lambda \big[d_j - c_j^\top (y - t - w)\big] \big\} \\ &= \min_t \big\{ \|t\|_2 + \lambda c_j^\top t \big\} + \lambda \big[d_j - c_j^\top (y - w)\big]. \end{aligned}$$

Note that

$$\min_t\{\|t\|_2 + \lambda c_j^\top t\} = \begin{cases} 0 & \text{if } \lambda\|c_j\|_2 \le 1 \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, the dual problem of (3.3.6) can be derived as

$$\begin{cases} \max_\lambda \ \lambda\big[d_j - c_j^\top(y-w)\big] \\ \text{s.t.} \quad \lambda\|c_j\|_2 \le 1 \\ \qquad \lambda \ge 0 \end{cases} = \begin{cases} \max_\lambda \ \lambda\big[d_j - c_j^\top(y-w)\big] \\ \text{s.t.} \quad \lambda \le \frac{1}{\|c_j\|_2} \\ \qquad \lambda \ge 0 \end{cases}$$
$$= \left[\frac{d_j - c_j^\top(y-w)}{\|c_j\|_2}\right]^+. \tag{3.3.7}$$

The primal problem satisfies the refined Slater's conditions, as the inequality constraint is linear and the primal problem is feasible [55, Section 5.2.3]. Therefore, we conclude that strong duality holds.

Now that we have the distance from a single halfspace, the distance from the safe region can be written as

$$\begin{aligned} \operatorname{dist}(y, \mathcal{Y} + w) &= \min_{j=1,\dots,m}\{\operatorname{dist}(y, \mathcal{Y}_j + w)\} \\ &= \min_{j=1,\dots,m}\left\{\frac{\big[d_j - c_j^\top(y-w)\big]^+}{\|c_j\|_2}\right\}, \end{aligned}$$

where the second equality follows directly from (3.3.7). This concludes the proof because minimum and $(\cdot)^+$ are interchangeable. □

3.3.2 Reformulation of Distributionally Robust Risk Constraint

The next step is to reformulate the distributionally robust risk constraint (3.3.4) in a conservative manner. This reformulation will then be suitable for our purpose of limiting safety risk.

Lemma 2. *Suppose that the safe region is given by* (3.3.5)*. Then, the distributionally robust safety risk is upper-bounded as follows:*

$$\sup_{\mu\in\mathbb{D}} \mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y,\mathcal{Y}+w)] \le \inf_{z\in\mathbb{R}} z+\frac{1}{1-\alpha}\sup_{\mu\in\mathbb{D}}\mathbb{E}^\mu\Big[\max\big\{\min_j p_j(y,w)-z,-z,0\big\}\Big],$$

where $p_j(y,w)=\frac{d_j-c_j^\top(y-w)}{\|c_j\|_2}$.

Proof. By the definition of CVaR and Lemma 1, we have

$$\begin{aligned}\mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y,\mathcal{Y}+w)] &= \inf_{z\in\mathbb{R}}\mathbb{E}^\mu\left[z+\frac{\big(\mathrm{dist}(y,\mathcal{Y}+w)-z\big)^+}{1-\alpha}\right]\\ &= \inf_{z\in\mathbb{R}}\mathbb{E}^\mu\left[z+\frac{1}{1-\alpha}\Big(\big[\min_j p_j(y,w)\big]^+-z\Big)^+\right].\end{aligned}$$

By the minimax inequality, we obtain that

$$\begin{aligned}&\sup_{\mu\in\mathbb{D}}\mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y,\mathcal{Y}+w)]\\ &\qquad\le \inf_{z\in\mathbb{R}}\sup_{\mu\in\mathbb{D}}\mathbb{E}^\mu\left[z+\frac{1}{1-\alpha}\Big(\big[\min_j p_j(y,w)\big]^+-z\Big)^+\right]\\ &\qquad= \inf_{z\in\mathbb{R}}\sup_{\mu\in\mathbb{D}}\mathbb{E}^\mu\left[z+\frac{1}{1-\alpha}\max\big\{\min_j p_j(y,w)-z,-z,0\big\}\right],\end{aligned}$$

and therefore the result follows. □

The upper-bound of the worst-case CVaR in Lemma 2 is still difficult to evaluate because its inner maximization problem involves optimization over a set of distributions. To resolve this issue, we use Wasserstein DRO based on Kantorovich duality to transform it into a finite-dimensional optimization problem as follows.

Proposition 2. *Suppose that the uncertainty set is a compact convex polytope, i.e.* $\mathbb{W}:=\{w\in\mathbb{R}^{n_y}\mid Hw\le h\}$*, where* $H\in\mathbb{R}^{q\times n_y}$ *and* $h\in\mathbb{R}^q$*. Then, the following*

equality holds:

$$
\begin{aligned}
&\sup_{\mu\in\mathbb{D}} \quad \mathbb{E}^{\mu}\Big[\max\Big\{\min_{j=1,\dots,m} p_j(y,w)-z,-z,0\Big\}\Big] = \\
&\inf_{\lambda,s,\rho,\gamma,\eta,\zeta} \lambda\theta+\sum_{i=1}^{N} s_i \\
&\quad s.t.\ \langle\rho_i, G(y-\hat{w}^{(i)})+g\rangle+\langle\gamma_i, h-H\hat{w}^{(i)}\rangle \le s_i+z \\
&\qquad \langle\eta_i, h-H\hat{w}^{(i)}\rangle \le s_i+z \\
&\qquad \langle\zeta_i, h-H\hat{w}^{(i)}\rangle \le s_i \\
&\qquad \|H^{\top}\gamma_i - G^{\top}\rho_i\|_* \le \lambda \\
&\qquad \|H^{\top}\eta_i\|_* \le \lambda \\
&\qquad \|H^{\top}\zeta_i\|_* \le \lambda \\
&\qquad \langle\rho_i, e_m\rangle = 1 \\
&\qquad \gamma_i\ge 0,\ \rho_i\ge 0,\ \eta_i\ge 0,\ \zeta_i\ge 0,
\end{aligned}
$$

where all the constraints hold for $i=1,\dots,N$, and the dual norm $\|\cdot\|_*$ is defined by $\|z\|_* := \sup_{\|\xi\|\le 1}\langle z,\xi\rangle$. Here, $G\in\mathbb{R}^{m\times n_y}$ is a matrix with rows $-\frac{c_j^{\top}}{\|c_j\|_2}$, $j=1,\dots,m$, $g\in\mathbb{R}^m$ is a column vector with entries $\frac{d_j}{\|c_j\|_2}$, $j=1,\dots,m$, and $e_m\in\mathbb{R}^m$ is a vector of all ones.

Proof. By the Kantorovich duality principle, we can rewrite the upper-bound of the worst-case CVaR in Lemma 2 in the following dual form:

$$
\begin{aligned}
&\sup_{\mu\in\mathbb{D}} \mathbb{E}^{\mu}\Big[\max\Big\{\min_{j=1,\dots,m} p_j(y,w)-z,-z,0\Big\}\Big] = \\
&\inf_{\lambda\ge 0}\Big[\lambda\theta+\frac{1}{N}\sum_{i=1}^{N}\sup_{w\in\mathbb{W}}\Big[\max\Big\{\min_{j=1,\dots,m} p_j(y,w)-z,-z,0\Big\}-\lambda\|w-\hat{w}^{(i)}\|\Big]\Big].
\end{aligned}
$$

It is proved in [46, Theorem 1] that strong duality holds. Introducing new auxiliary variable s and following the procedure in [1], the dual problem above can be expressed

as

$$
\begin{cases}
\inf_{\lambda,s} \ \lambda\theta + \frac{1}{N}\sum_{i=1}^{N} s_i \\
\text{s.t.} \quad \sup_{w\in\mathbb{W}} \Big[\max\big\{ \min_j p_j(y,w) - z, -z, 0 \big\} - \lambda\|w - \hat{w}^{(i)}\| \Big] \leq s_i \\
\qquad \lambda \geq 0
\end{cases}
$$

$$
= \begin{cases}
\inf_{\lambda,s} \ \lambda\theta + \frac{1}{N}\sum_{i=1}^{N} s_i \\
\text{s.t.} \quad \sup_{w\in\mathbb{W}} [- \max_{\|\xi_{i,1}\|_*\leq\lambda} \langle \xi_{i,1}, w - \hat{w}^{(i)} \rangle + \min_j p_j(y,w)] - z \leq s_i \\
\qquad \sup_{w\in\mathbb{W}} [- \max_{\|\xi_{i,2}\|_*\leq\lambda} \langle \xi_{i,2}, w - \hat{w}^{(i)} \rangle] - z \leq s_i \\
\qquad \sup_{w\in\mathbb{W}} [- \max_{\|\xi_{i,3}\|_*\leq\lambda} \langle \xi_{i,3}, w - \hat{w}^{(i)} \rangle] \leq s_i \\
\qquad \lambda \geq 0,
\end{cases}
$$

where the constraints hold for all i. In the second problem, we decompose the expression inside maximum and employ the definition of dual norm. Thereafter, since the set $\{\xi_{i,k} \mid \|\xi_{i,k}\|_* \leq \lambda\}$ is compact for any $\lambda \geq 0$, the minimax theorem can be used to rewrite the problem as

$$
\begin{cases}
\inf_{\lambda,s} \ \lambda\theta + \frac{1}{N}\sum_{i=1}^{N} s_i \\
\text{s.t.} \quad \min_{\|\xi_{i,1}\|_*\leq\lambda} \sup_{w\in\mathbb{W}} [- \langle \xi_{i,1}, w - \hat{w}^{(i)} \rangle + \min_j p_j(y,w)] - z \leq s_i \\
\qquad \min_{\|\xi_{i,2}\|_*\leq\lambda} \sup_{w\in\mathbb{W}} [- \langle \xi_{i,2}, w - \hat{w}^{(i)} \rangle] - z \leq s_i \\
\qquad \min_{\|\xi_{i,3}\|_*\leq\lambda} \sup_{w\in\mathbb{W}} [- \langle \xi_{i,3}, w - \hat{w}^{(i)} \rangle] \leq s_i \\
\qquad \lambda \geq 0
\end{cases}
$$

$$
= \begin{cases}
\inf_{\lambda,s,\xi} \ \lambda\theta + \frac{1}{N}\sum_{i=1}^{N} s_i \\
\text{s.t.} \quad \sup_{w\in\mathbb{W}} [\langle \xi_{i,1}, w \rangle + \min_j p_j(y,w)] - \langle \xi_{i,1}, \hat{w}^{(i)} \rangle - z \leq s_i \\
\qquad \sup_{w\in\mathbb{W}} \langle \xi_{i,2}, w \rangle - \langle \xi_{i,1}, \hat{w}^{(i)} \rangle - z \leq s_i \\
\qquad \sup_{w\in\mathbb{W}} \langle \xi_{i,3}, w \rangle - \langle \xi_{i,1}, \hat{w}^{(i)} \rangle \leq s_i \\
\qquad \|\xi_{i,k}\|_* \leq \lambda, \ k = 1,2,3,
\end{cases}
$$

where the constraints hold for all i. The first constraint can be written as sum of a conjugate function and the support function $\sigma_{\mathbb{W}}(\nu_i) := \sup_{w\in\mathbb{W}} \langle \nu_i, w \rangle$ since $-p_j(y,w)$ is proper, convex and lower semicontinuous. Likewise, the next two constraints can be

represented using $\sigma_{\mathbb{W}}(\xi_{i,2})$ and $\sigma_{\mathbb{W}}(\xi_{i,3})$ as follows:

$$\begin{cases} \inf_{\lambda,s,\xi,\nu} \ \lambda\theta + \sum_{i=1}^{N} s_i \\ \text{s.t.} \qquad \sup_w[\langle \xi_{i,1} - \nu_i, w\rangle + \min_j p_j(y,w)] + \sigma_{\mathbb{W}}(\nu_i) - \langle \xi_{i,1}, \hat{w}^{(i)}\rangle - z \leq s_i \\ \qquad\quad \sigma_{\mathbb{W}}(\xi_{i,2}) - \langle \xi_{i,2}, \hat{w}^{(i)}\rangle - z \leq s_i \\ \qquad\quad \sigma_{\mathbb{W}}(\xi_{i,3}) - \langle \xi_{i,3}, \hat{w}^{(i)}\rangle \leq s_i \\ \qquad\quad \|\xi_{i,k}\|_* \leq \lambda, \ k = 1,2,3, \end{cases} \tag{3.3.8}$$

where the constraints hold for all i.

On the other hand, we note that

$$\sup_w \left[\langle \xi_{i,1} - \nu_i, w\rangle + \min_{j=1,\ldots,m} p_j(y,w)\right] = \begin{cases} \sup_{w,\tau} & \langle \xi_{i,1} - \nu_i, w\rangle + \tau \\ \text{s.t.} & G(y-w) + g \geq \tau e \end{cases}$$

$$= \begin{cases} \inf_{\rho_i} & \langle \rho_i, g + Gy\rangle \\ \text{s.t.} & G^\top \rho_i = \nu_i - \xi_{i,1} \\ & \langle \rho_i, e_m\rangle = 1 \\ & \rho_i \geq 0, \end{cases}$$

where the last equality follows from strong duality of linear programming, which holds because the primal maximization problem is feasible. By the definition of support functions, we also have

$$\sigma_{\mathbb{W}}(\nu_i) = \begin{cases} \sup_w & \langle \nu_i, w\rangle \\ \text{s.t.} & Hw \leq h \end{cases} = \begin{cases} \inf_{\gamma_i} & \langle \gamma_i, h\rangle \\ \text{s.t.} & H^\top \gamma_i = \nu_i \\ & \gamma_i \geq 0, \end{cases}$$

where the last equality follows from strong duality of linear programming, which holds since the uncertainty set is nonempty. Similar expressions are derived for $\sigma_{\mathbb{W}}(\xi_{i,2})$ and $\sigma_{\mathbb{W}}(\xi_{i,3})$ with Lagrangian multipliers η_i and ζ_i, respectively. By substituting the results above into (3.3.8), we conclude that the proposed reformulation is exact. □

3.3.3 Reformulation of the Wasserstein DR-MPC Problem

We are now ready to reformulate the Wasserstein DR-MPC problem (3.3.3) as a finite-dimensional optimization problem by using Lemma 2 and Proposition 2. Putting all the pieces in Lemma 2 and Proposition 2 together into (3.3.3), we have

$$\inf_{\substack{\mathbf{u},\boldsymbol{\xi},\mathbf{y},\mathbf{z},\boldsymbol{\lambda},\\ \mathbf{s},\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\eta},\boldsymbol{\zeta}}} J(\xi(t),\mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k,u_k) + q(\xi_K) \tag{3.3.9a}$$

$$\text{s.t. } \xi_{k+1} = f(\xi_k,u_k) \tag{3.3.9b}$$

$$y_k = h(\xi_k,u_k) \tag{3.3.9c}$$

$$\xi_0 = \xi(t) \tag{3.3.9d}$$

$$z_{\ell,k} + \frac{1}{1-\alpha}\Big[\lambda_{\ell,k}\theta + \frac{1}{N_k}\sum_{i=1}^{N_k} s_{\ell,k,i}\Big] \le \delta_\ell \tag{3.3.9e}$$

$$\langle \rho_{\ell,k,i}, G_{\ell,t}(y_k - \hat{w}^{(i)}_{\ell,t,k}) + g_{\ell,t}\rangle + \langle \gamma_{\ell,k,i}, h_\ell - H_\ell \hat{w}^{(i)}_{\ell,t,k}\rangle \le s_{\ell,k,i} + z_{\ell,k} \tag{3.3.9f}$$

$$\langle \eta_{\ell,k,i}, h_\ell - H_\ell \hat{w}^{(i)}_{\ell,t,k}\rangle \le s_{\ell,k,i} + z_{\ell,k} \tag{3.3.9g}$$

$$\langle \zeta_{\ell,k,i}, h_\ell - H_\ell \hat{w}^{(i)}_{\ell,t,k}\rangle \le s_{\ell,k,i} \tag{3.3.9h}$$

$$\|H_\ell^\top \gamma_{\ell,k,i} - G_{\ell,t}^\top \rho_{\ell,k,i}\|_* \le \lambda_{\ell,k} \tag{3.3.9i}$$

$$\|H_\ell^\top \eta_{\ell,k,i}\|_* \le \lambda_{\ell,k} \tag{3.3.9j}$$

$$\|H_\ell^\top \zeta_{\ell,k,i}\|_* \le \lambda_{\ell,k} \tag{3.3.9k}$$

$$\langle \rho_{\ell,k,i}, e_m\rangle = 1 \tag{3.3.9l}$$

$$\gamma_{\ell,k,i}, \rho_{\ell,k,i}, \eta_{\ell,k,i}, \zeta_{\ell,k,i} \ge 0 \tag{3.3.9m}$$

$$\xi_k \in \Xi,\ u_k \in \mathcal{U},\ z_{\ell,k} \in \mathbb{R}, \tag{3.3.9n}$$

where all the constraints hold for $k = 1,\ldots,K$, $\ell = 1,\ldots,L$ and $i = 1,\ldots,N_k$, except for the first constraint and $u_k \in \mathcal{U}$, which should hold for $k = 0,\ldots,K-1$, and the second constraint and $\xi_k \in \Xi$, which should hold for $k = 0,\ldots,K$.

The overall motion control process is as follows. First, at stage t the initial state ξ_0 in MPC is set to be the current state $\xi(t)$. Also, the current safe region $\mathcal{Y}_\ell(t)$ is observed

to return $G_{\ell,t}$ and $g_{\ell,t}$. Second, the Wasserstein DR-MPC problem (3.3.9) is solved to find a solution $\mathbf{u}^\star$ satisfying the risk constraint even when the actual distribution deviates from the empirical distribution (3.3.1) within the Wasserstein ball (3.3.2). Then, the first component of the optimal control input sequence $u_0^\star$ is selected as the control input at stage t and applied to the robotic vehicle. These two steps are repeated for all time stages until the desired position in the configuration space is reached.

As a consistency check, we ascertain that in ambiguity-free case the above problem reduces to a risk-constrained MPC problem evaluated under the empirical distribution, equivalent to the SAA-MPC introduced in our previous paper [41]. By replacing $\theta = 0$, we get that

$$\inf_{\substack{\mathbf{u},\boldsymbol{\xi},\mathbf{y}\\ \mathbf{z},\mathbf{s},\boldsymbol{\rho}}} J(\xi(t),\mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k,u_k) + q(\xi_K) \tag{3.3.10a}$$

$$\text{s.t. } \xi_{k+1} = f(\xi_k,u_k) \tag{3.3.10b}$$

$$y_k = h(\xi_k,u_k) \tag{3.3.10c}$$

$$\xi_0 = \xi(t) \tag{3.3.10d}$$

$$z_{\ell,k} + \frac{1}{(1-\alpha)N_k}\sum_{i=1}^{N_k} s_{\ell,k,i} \le \delta_\ell \tag{3.3.10e}$$

$$\langle \rho_{\ell,k,i}, G_{\ell,t}(y_k - \hat{w}_{\ell,t,k}^{(i)}) + g_{\ell,t}\rangle \le s_{\ell,k,i} + z_{\ell,k} \tag{3.3.10f}$$

$$0 \le s_{\ell,k,i} + z_{\ell,k} \tag{3.3.10g}$$

$$0 \le s_{\ell,k,i} \tag{3.3.10h}$$

$$\langle \rho_{\ell,k,i}, e_m\rangle = 1 \tag{3.3.10i}$$

$$\rho_{\ell,k,i} \ge 0 \tag{3.3.10j}$$

$$\xi_k \in \Xi,\ u_k \in \mathcal{U},\ z_{\ell,k} \in \mathbb{R}, \tag{3.3.10k}$$

The proposed reformulation resolves the infinite-dimensionality issue in the original Wasserstein DR-MPC problem. Thus, the reformulated problem is easier to solve than the original one. However, it is still nonconvex due to the nonlinear system dynamics and output equations, as well as the bilinearity of the fifth constraint (3.3.9f);

all the other constraints and the objective function are convex. Thus, a locally optimal solution can be found by using efficient nonlinear programming (NLP) algorithms such as interior-point methods, sequential quadratic programming, etc [56]. However, in some specific cases, e.g., when the system dynamics and the output equations are affine, we can use relaxation techniques to find a globally optimal solution. One such relaxation method is the Spatial Branch-and-Bound, the detailed application of which on our problem can be found in [57].

3.4 Out-of-Sample Performance Guarantee

As mentioned previously, the distribution μ is never observable but must be inferred from data. However, if we calibrate a stochastic program to a given training dataset and evaluate its optimal decision on a different testing dataset, then the resulting out-of-sample performance is often disappointing, even when the two datasets are generated from the same underlying distribution μ. Such defect exists in case of SAA-based methods. However, the advantage of the Wasserstein DR-MPC method is that it assures a *probabilistic out-of-sample performance guarantee*, meaning that the safety risk constraint is satisfied with probability no less than a certain threshold, even when evaluated under a set of new samples chosen independently of the training data. This is a finite-sample (non-asymptotic) guarantee, which cannot be attained in many popular methods such as SAA.

Let $(\mathbf{u}^\star, \xi^\star, \mathbf{y}^\star)$ denote an optimal solution to the Wasserstein DR-MPC problem (3.3.3) at stage t, obtained by using the training dataset $\{\hat{w}_{\ell,t,k}^{(1)}, \ldots, \hat{w}_{\ell,t,k}^{(N_k)}\}$. Then, the *out-of-sample risk* at stage t is defined by

$$\mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t) + w_{\ell,t,1})], \tag{3.4.1}$$

which represents the risk of unsafety evaluated under the (unknown) true loss distribution μ. However, as μ is unknown in practice, it is impossible to exactly evaluate the out-of-sample risk. Instead, we seek a motion control solution that provides the

following probabilistic performance guarantee:

$$\mu^{N_1}\Big\{\mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y^\star(t+1),\mathcal{Y}(t)+w_{\ell,t,1})]\le\delta_\ell\Big\}\ge 1-\beta\quad\forall t, \tag{3.4.2}$$

where $\beta\in(0,1)$. This inequality represents that the risk of unsafety is no greater than the risk-tolerance parameter δ with $(1-\beta)$ confidence level. We refer to the probability on the left-hand side of (3.4.2) as the *reliability* of the motion control. The reliability increases with the Wasserstein ball radius θ. Thus, θ needs to be carefully determined to establish the probabilistic out-of-sample performance guarantee with desired β.

The required radius can be found from the following measure concentration inequality for Wasserstein ambiguity sets [58, Theorem 2]:[2]

$$\mu^{N_1}\big\{\hat{w}\mid W(\mu,\nu)\ge\theta\big\}\le c_1\big[b_1(N_1,\theta)\mathbf{1}_{\{\theta\le 1\}}+b_2(N_1,\theta)\mathbf{1}_{\{\theta>1\}}\big], \tag{3.4.3}$$

where

$$b_1(N,\theta):=\begin{cases}\exp(-c_2N\theta^2) & \text{if } n_y<2\\ \exp(-c_2N(\frac{\theta}{\log(2+1/\theta)})^2) & \text{if } n_y=2\\ \exp(-c_2N\theta^{n_y}) & \text{otherwise}\end{cases}$$

$$b_2(N,\theta):=\exp(-c_2N\theta^c)$$

for some constants $c_1,c_2>0$. Suppose that the radius is chosen as

$$\theta:=\begin{cases}\left[\frac{\log(c_1/\beta)}{c_2N_1}\right]^{1/c} & \text{if } N_1<\frac{1}{c_2}\log(c_1/\beta)\\ \left[\frac{\log(c_1/\beta)}{c_2N_1}\right]^{1/n_y} & \text{if } N_1\ge\frac{1}{c_2}\log(c_1/\beta),\ n_y<2\\ \left[\frac{\log(c_1/\beta)}{c_2N_1}\right]^{1/2} & \text{if } N_1\ge\frac{1}{c_2}\log(c_1/\beta),\ n_y>2\\ \bar{\theta} & \text{if } N_1\ge\frac{(\log 3)^2}{2}\log(c_1/\beta),\ n_y=2\end{cases}$$

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

$$\frac{\bar{\theta}}{\log(2+1/\bar{\theta})}=\left[\frac{\log(c_1/\beta)}{c_2N_1}\right]^{1/2}.$$

[2]The measure concentration inequality assumes that $\mathbb{E}^\mu[\exp(\|w\|^c)]\le B$ for $c>1$ and $B>0$, i.e. light-tailed distribution. In our problem formulation, this condition holds trivially for any compact uncertainty set $\mathbb{W}$.

Then, by the measure concentration inequality (3.4.3), we have

$$\mu^{N_1}\big\{\hat{w} \mid W(\mu,\nu) \leq \theta\big\} \geq 1-\beta.$$

It follows that for each t

$$\mu^{N_1}\Big\{\mathrm{CVaR}_\alpha^{\mu}[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t)+w_{\ell,t,1})] \\ \leq \sup_{\mu'\in\mathbb{D}} \mathrm{CVaR}_\alpha^{\mu'}[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t)+w_{\ell,t,1})]\Big\} \geq 1-\beta.$$

Since $\sup_{\mu'\in\mathbb{D}} \mathrm{CVaR}_\alpha^{\mu'}[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t)+w_{\ell,t,1})] \leq \delta_\ell$ by the definition of $y^\star$, we conclude that the probabilistic performance guarantee (3.4.2) holds with the choice of θ above. Similar results are also derived in [1, Theorem 3.5] and [29, Theorem 3]. The constants c_1 and c_2 can be explicitly found using the proof of [58, Theorem 2]. However, this choice often leads to an overly conservative radius θ. One can obtain a less conservative θ by using bootstrapping or cross-validation methods [1]. In the following section, we show how θ can be selected based on numerical experiments, depending on the choice of sample size.

3.5 Numerical Experiments

In this section, we present simulation results that demonstrate the performance and the utility of the Wasserstein Distributionally Robust motion control method.

In our simulations we consider randomly translating obstacles and analyze the behavior of two mobile robots, (i) a 5-dimensional nonlinear car-like vehicle model, and (ii) a 12-dimensional linearized quadrotor model when controlled by WDR-MPC problem (3.3.9) to follow the reference trajectory generated by RRT*. Here we also investigate the out-of-sample performance of our method, by computing the worst-case reliability and the worst-case and average risks for different Wasserstein radii and sample sizes.

In our experimental results we demonstrate the advantage of using DR-MPC over the SAA-based risk-aware MPC method. The parameters of the robot models in Ta-

Table 3.1: Robotic vehicle parameters

Car-like model		Quadrotor model	
m_V	$1700\ kg$	m_Q	$0.65\ kg$
C_f	$50\ kN/rad$	g	$9.81\ ms^2$
C_r	$50\ kN/rad$	l_Q	$0.23\ m$
I_z	$6000\ kg \cdot m^2$	I_{xx}	$0.0075\ kg \cdot m^2$
L_f	$1.2\ m$	I_{yy}	$0.0075\ kg \cdot m^2$
L_r	$1.3\ m$	I_{zz}	$0.013\ kg \cdot m^2$
v_x^r	$5\ m/s$		

ble 3.1 were used throughout the simulations. All the simulations were conducted on a PC with 3.70 GHz Intel Core i7-8700K processor and 32 GB RAM. The optimization problem was modeled in AMPL [59] and solved using interior-point method-based solver IPOPT [60]. For the quadrotor model, we also applied an algorithm called Spatial Branch-and-Bound (sBB) to find the globally optimal solution, which in general cannot be obtained by nonlinear program solvers.

3.5.1 Nonlinear Car-Like Vehicle Model

Consider a car-like vehicle navigating in a 2D environment with the following nonlinear model [61]:

$$
\begin{aligned}
\dot{X}^r &= v_x^r \cos\theta^r - v_y^r \sin\theta^r \\
\dot{Y}^r &= v_x^r \sin\theta^r + v_y^r \cos\theta^r \\
\dot{\theta}^r &= \omega^r \\
\dot{v}_y^r &= \frac{-2(C_f + C_r)}{m_V v_x^r} v_y^r - \Big(\frac{2l_f C_f - 2l_r C_r}{m_V v_x^r} + v_x^r\Big)\omega^r + \frac{2C_f}{m_V}\delta_f^r \\
\dot{\omega}^r &= \frac{-2(l_f C_f + l_r C_r)}{I_z v_x^r} v_y^r - \frac{2l_f^2 C_f - 2l_r^2 C_r}{I_z v_x^r}\omega^r + \frac{2l_f C_f}{I_z}\delta_f^r,
\end{aligned}
$$

Table 3.2: Computation time and operation cost for the nonlinear car-like vehicle motion control with $N_k = 10$, $\delta_\ell = 0.02$, and $\alpha = 0.95$

	SAA-MPC	DR-MPC (θ)			
		0.0005	0.001	0.0015	0.002
Cost	$+\infty$	642.31	788.52	885.69	942.97
Time (sec)	-	113.64	131.19	221.66	226.66

where the state variables $X^r, Y^r, \theta^r, v_y^r$, and ω^r correspond to the vehicle's center of gravity in the inertial frame, lateral velocity, orientation and yaw rate, respectively. In addition, v_x^r is the constant longitudinal velocity, m_V is the mass of the vehicle, I_z is the moment of inertia around the z axis, C_f and C_r are the cornering stiffness coefficients for the respective front and rear tires, and finally, L_f and L_r are the distances from the center of gravity to the front and rear wheels. The output variables are chosen as the X^r and Y^r coordinates of the vehicle.

The task is to design a controller that steers the vehicle to its goal position while avoiding the two randomly perturbing rectangular obstacles that are shown in Fig. 3.3. The random movement of each obstacle in each direction is sampled from a uniform distribution in $[-0.2, 0.2]$. Therefore, the support is $\mathbb{W}_\ell = \{w \in \mathbb{R}^2 \mid w \leq 0.2, w \geq -0.2\},\ \ell = 1, 2$. Also, we used ℓ_2-norm to measure distance in the uncertainty space. The MPC horizon is set to $K = 20$. The weight matrix Q is chosen as a 5×5 diagonal matrix with diagonal entries $(1, 1, 0, 0, 0)$. Let $P = 1.2Q$ and $R = 0.01$. The MPC problem is solved for $T = 80$ iterations using the discretized vehicle model with sample time $T_s = 0.05\ sec$. The interior-point method-based solver IPOPT was used to numerically solve the optimization problem (3.3.9) at each MPC iteration.

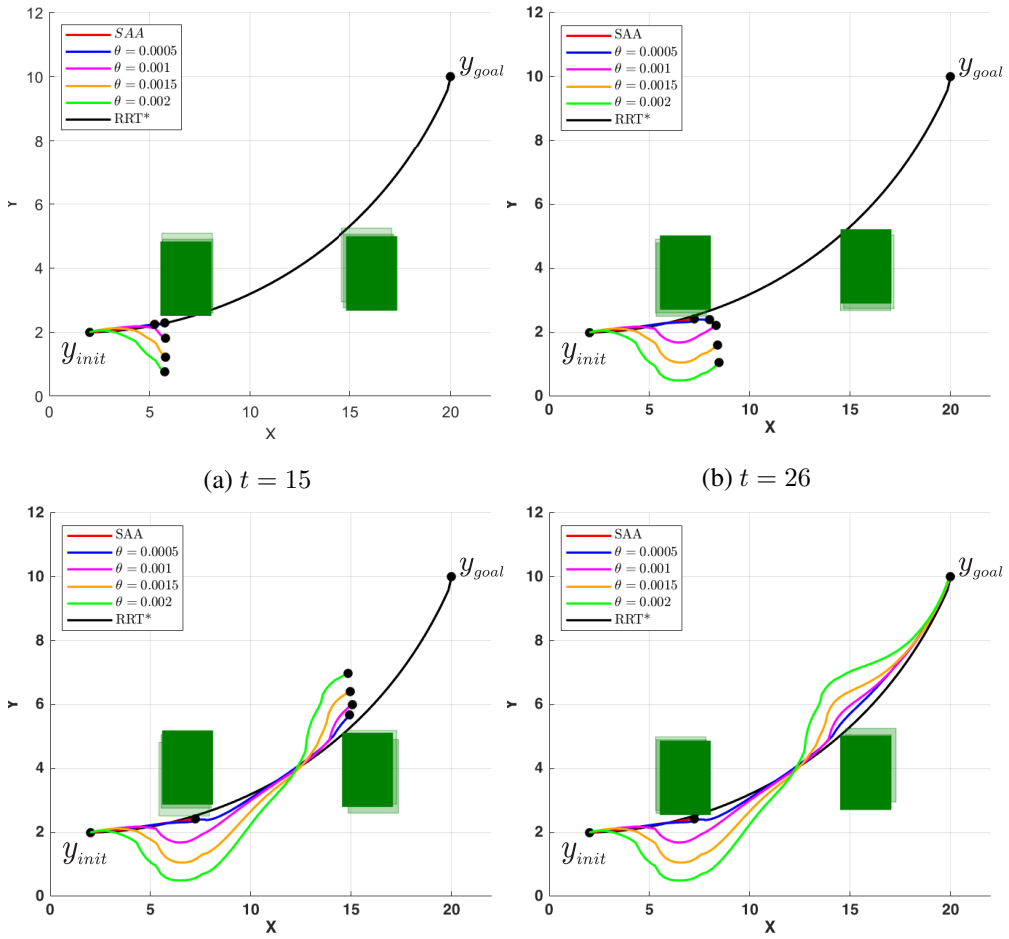
We first examine the effect of the Wasserstein ball radius θ and compare DR-MPC with SAA-MPC [41]. Fig. 3.3 shows the controlled trajectories for different θ's computed with $\delta_\ell = 0.02$, $\alpha = 0.95$ and $N_k = 10$ sample data. As shown in Fig. 3.3

Table 3.3: The worst-case reliability for the car-like vehicle motion control

$N_k \setminus \theta$	0.0005	0.00075	0.001	0.00125
10	0.26	0.26	0.49	1.00
20	0.31	0.66	0.66	1.00
50	0.55	0.65	0.74	1.00
100	0.60	0.65	1.00	1.00

(a), in the early stages the vehicle follows the reference trajectory in the case of SAA-MPC and DR-MPC with small θ, even though the robot gets closer to the first obstacle. However, in the case of DR-MPC with $\theta = 0.0015$, the robot proactively takes into account the obstacle's uncertainty for collision avoidance. The same behavior occurs for $\theta = 0.002$ with a bigger safety margin. Thus, the robot further deviates from the reference trajectory. At $t = 24$, the robot controlled by SAA-MPC violates the safety constraint and thus its operation is terminated. When DR-MPC is used, the robot passes the obstacle at $t = 26$ without any collision. The trajectory generated with $\theta = 0.0005$ barely avoids the obstacle because the control action is not sufficiently robust. However, when a bigger radius is used, the robot avoids the obstacle with a wide enough safety margin. At $t = 56$, the vehicle reaches the second obstacle. All four trajectories generated by DR-MPC are collision-free as desired. At $t = 80$, the vehicle reaches the goal position and the MPC iterations terminate.

Overall, we conclude that, with a small sample size of $N_k = 10$, the SAA approach gives an infeasible result due to a violation of safety constraints, while the DR approach successfully avoids obstacles. Table 3.2 shows the total computation time and the total cost $\sum_{t=0}^{T} r(\xi^\star(t), u^\star(t))$ for SAA-MPC and DR-MPC with different θ's. The total cost increases with θ because a larger θ induces a more cautious control action that causes further deviations from the reference path. Thus, there is a fundamental tradeoff between risk and cost.



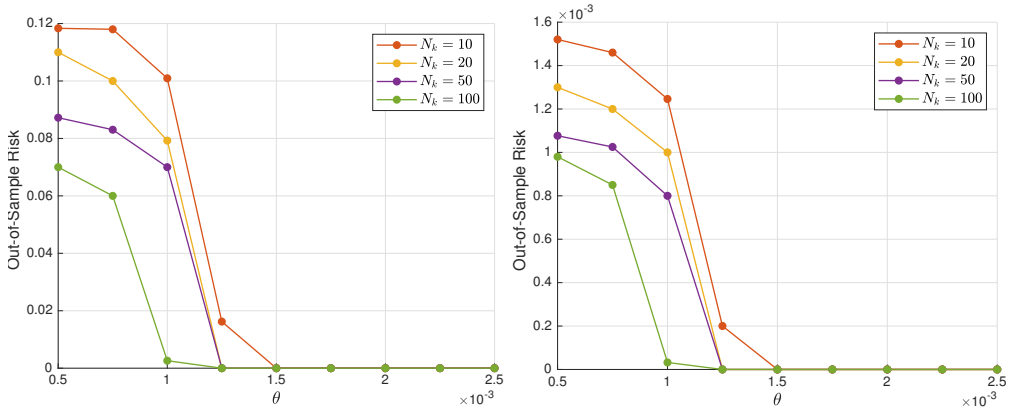
(a) $t = 15$

(b) $t = 26$

(c) $t = 56$

(d) $t = 80$

Figure 3.3: Trajectories of the nonlinear car-like vehicle model controlled by SAA-MPC and DR-MPC with multiple θ's.



(a) (b)

Figure 3.4: (a) Worst-case and (b) the average out-of-sample risk for the car-like vehicle.

We now investigate the out-of-sample safety risk by varying radius θ and sample size N_k. Specifically, for the ℓth obstacle we evaluate the worst-case out-of-sample risk

$$\max_{t=0,\ldots,T-1} \mathrm{CVaR}_\alpha^\mu\big[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t) + w_{\ell,t,1})\big],$$

and the average out-of-sample risk

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathrm{CVaR}_\alpha^\mu\big[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t) + w_{\ell,t,1})\big].$$

We estimated the CVaR using 20,000 independent samples generated from the true distribution μ. The worst-case and average out-of-sample risks for different sample sizes and radii are shown in Fig. 3.4a and 3.4b, respectively. The worst-case out-of-sample risk is approximately 70 times larger than its average counterpart. Both out-of-sample risks monotonically decrease with radius θ and sample size N_k. Recall that the risk tolerance is chosen as $\delta_\ell = 0.02$. In the case of $N_k = 10$, the risk constraints for all stages are satisfied if $\theta \geq 0.0015$. In all the other cases, the constraints are met for $\theta \geq 0.00125$.

For the probabilistic out-of-sample performance guarantee (3.4.2), we compute the worst-case reliability

$$\min_{t=0,\ldots,T-1} \mu^{N_1}\Big\{\mathrm{CVaR}_\alpha^\mu[\mathrm{dist}(y^\star(t+1), \mathcal{Y}(t) + w_{\ell,t,1})] \leq \delta_\ell\Big\}$$

with 200 independent simulations with 1,000 samples in each. Table 3.3 shows the estimated reliability depending on radius θ and sample size N_k. The reliability increases with θ and N_k as expected. When $N_k = 10$, the probability of meeting all the risk constraints for all stages is as low as 0.26 (with a very small radius, $\theta = 0.0005$). However, there is a sharp transition between $\theta = 0.001$ and $\theta = 0.00125$, and the reliability reaches its maximal value 1 when $\theta = 0.00125$. In the case of larger sample sizes, e.g., $N_k = 100$, the reliability is relatively high even with a very small radius and reaches 1 when $\theta = 0.001$.

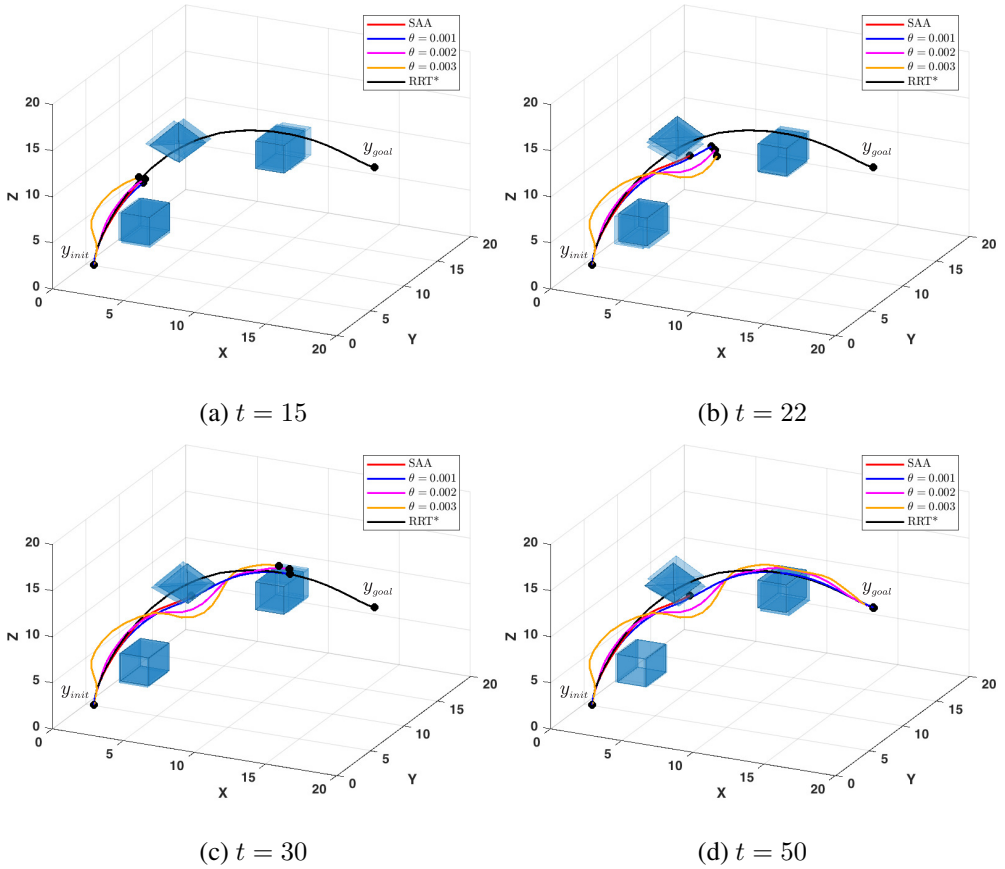
3.5.2 Linearized Quadrotor Model

Consider a quadrotor navigating in a 3D environment with the following linear dynamics:

$$\ddot{x}^r = -g\theta, \quad \ddot{y}^r = g\phi^r, \quad \ddot{z}^r = -\frac{l_Q}{m_Q}u_1^r,$$
$$\ddot{\phi}^r = \frac{1}{I_{xx}}u_2^r, \quad \ddot{\theta}^r = \frac{l_Q}{I_{yy}}u_3^r, \quad \ddot{\psi}^r = \frac{l_Q}{I_{zz}}u_4^r,$$

where m_Q is the quadrotor's mass, g is the gravitational acceleration, and I_{xx}, I_{yy} and I_{zz} are the area moments of inertia about the principle axes in the body frame, and l_Q represents the distance between the rotor and the center of mass of the quadrotor. The state of the quadrotor can be represented by its position and orientation with the corresponding velocities and rates in a 3D space — $(x^r, \dot{x}^r, y^r, \dot{y}^r, z^r, \dot{z}^r, \phi^r, \dot{\phi}^r, \theta^r, \dot{\theta}^r, \psi^r, \dot{\psi}^r) \in \mathbb{R}^{12}$. The outputs are taken as the X, Y and Z coordinates of the quadrotor's center of mass.

The quadrotor is controlled to reach the desired goal position while avoiding three randomly perturbing obstacles. The random motions of the obstacles in each direction are drawn from the normal distributions $\mathcal{N}(0.2, 0.1)$, $\mathcal{N}(-0.8, 0.3)$ and $\mathcal{N}(0.3, 0.2)$,



(a) $t = 15$ (b) $t = 22$

(c) $t = 30$ (d) $t = 50$

Figure 3.5: Trajectories of the quadrotor model controlled by SAA-MPC and DR-MPC with multiple θ's.

respectively. Therefore, the support for all the obstacles is $\mathbb{W}_\ell = \mathbb{R}^3,\ \ell = 1, 2, 3$. Also, we used ℓ_2-norm to measure distance in the uncertainty space. The MPC horizon is set to $K = 10$. The weight matrix Q is selected as a 12×12 diagonal matrix with diagonal entries $(1, 0, 1, 0, 1, 0, 0, \ldots, 0)$. We let $P = Q$ and $R = 0.02I$. The MPC problem is solved for $T = 50$ iterations by discretizing the quadrotor model with sample time $T_s = 0.1\ sec.$

The Wasserstein DR-MPC problem for the quadrotor model was solved using the sBB method with McCormick relaxation as proposed in [57], as all the constraints are

Table 3.4: Computation time and operation cost for the quadrotor motion control with $N_k = 10$, $\delta_\ell = 0.02$, and $\alpha = 0.95$

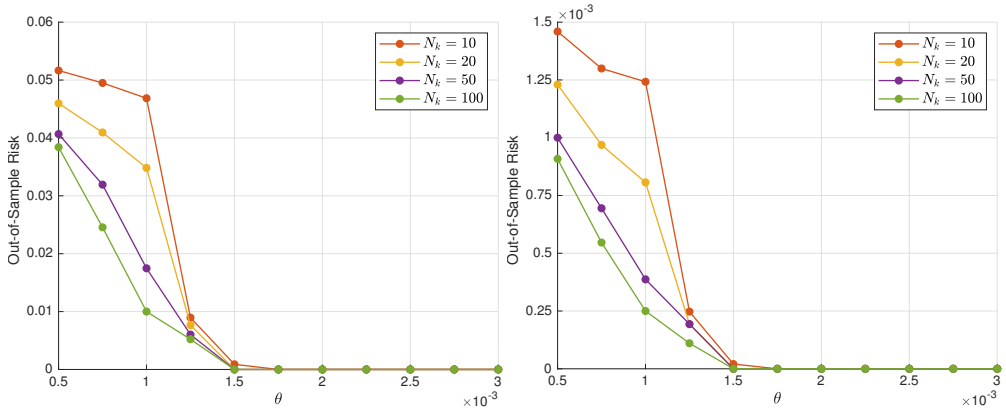
	Method	SAA	$\theta = 0.001$	$\theta = 0.002$	$\theta = 0.003$
Cost	**NLP**	$+\infty$	14.02	29.93	86.34
	sBB	$+\infty$	13.49	28.86	80.31
Time (sec)	**NLP**	$-$	47.69	77.08	136.02
	sBB	$-$	892.46	6093.33	9959.93

convex except the constraint (3.3.9f). The relaxed problem in the algorithm was solved using the solver Gurobi, while the original one was solved using the solver IPOPT.

The bound on the control input was chosen based on the range of angular velocity of the rotors. Thus, the control input is restricted to the set $\mathcal{U} := \{\boldsymbol{u} \in \mathbb{R}^4 \mid u_{\min} \leq \boldsymbol{u} \leq u_{\max}\}$ selected according to the motor specifications. In the simulation, we used $u_{\min} = (0, -22.52, -22.52, -1.08)$ and $u_{\max} = (90, 22.52, 22.52, 1.08)$. The state feasibility set $\Xi := \{\boldsymbol{x} \in \mathbb{R}^{12} \mid -\pi \leq \phi^r \leq \pi\ ,\ -\frac{\pi}{2} \leq \theta^r \leq \frac{\pi}{2},\ -\pi \leq \psi^r \leq \pi\}$ has been selected to limit the angles to avoid kinematic singularity.

The trajectories generated using $N_k = 10$ samples with $\delta_\ell = 0.02$ and $\alpha = 0.95$ are shown in Fig. 3.5. We observe that for $t < 15$ no collision occurs with the first obstacle. The trajectory with $\theta = 0.003$ is the safest as its deviation from the reference trajectory is the largest. At $t = 22$, the robot controlled by DR-MPC has passed the second moving obstacle while avoiding it. However, in the case of SAA-MPC the safety constraint at $t = 20$ is not satisfied, thereby resulting in a collision. At $t = 30$, the quadrotor controlled by DR-MPC is near the third obstacle. Similar to the previous stages, trajectories with bigger θ's continue to deviate further from the risky reference trajectory with a larger operation cost as shown in Table 3.4. At $t = 50$, the robot completes the task and reaches the desired goal position.

Table 3.4 shows the computation time and the total cost $\sum_{t=0}^{T} r(\xi^\star(t), u^\star(t))$ for



(a) (b)

Figure 3.6: (a) Worst-case and (b) the average out-of-sample risk for the quadrotor.

SAA-MPC and DR-MPC with different θ's. The Wasserstein DR-MPC problem is computed by two different methods: the sBB method with McCormick relaxation and the interior-point method implemented in IPOPT. Compared to SAA-MPC, Wasserstein DR-MPC shows a better performance in terms of the total cost and safety risk, while the computation time for SAA-MPC is lower than that for DR-MPC. From Table 3.4, we observe that the cost obtained by sBB is less than that obtained by the interior-point method. This is consistent with the fact that sBB finds a globally optimal solution while the interior-point method converges to a local optimum. However, the interior-point method is faster than sBB as expected.

The selection of θ meeting the desired out-of-sample performance guarantee can be achieved by the same method as in the previous scenario. Figures 3.6a and 3.6b show the worst-case and average out-of-sample risks estimated using 20,000 independent samples from the true distribution. As expected, the out-of-sample risk decreases with the sample size and the ambiguity set size. Table 3.5 shows the reliability $\min_{t=0,\ldots,T-1} \mu^{N_1}\{\mathrm{CVaR}[\mathrm{dist}(y_1, \mathcal{Y}(t) + w_{\ell,t,1})] \leq \delta_\ell\}$. The reliability does not significantly improve until $\theta = 0.001$. Instead, it remains almost constant for all sample sizes when $\theta \leq 0.001$, and then rapidly increases. A probabilistic guarantee of 0.92

Table 3.5: The worst-case reliability for the quadrotor motion control

$N_k \setminus \theta$	0.0005	0.00075	0.001	0.00125	0.0015
10	0.54	0.54	0.64	0.72	1.00
20	0.64	0.64	0.65	0.92	1.00
50	0.65	0.65	0.69	1.00	1.00
100	0.69	0.69	0.69	1.00	1.00

can be achieved on out-of-sample with only 20 sample data and $\theta = 0.00125$. Thus, we can conclude that 0.00125 is a reasonable choice for θ when only 20 sample data are available, with which we achieve an acceptable out-of-sample performance guarantee.

3.6 Conclusions

In this work, we developed a risk-aware distributionally robust motion control method for avoiding collisions with randomly moving obstacles. By limiting the safety risk in the presence of distribution errors within a Wasserstein ball, the proposed approach resolves the issue related to the inexact empirical distribution obtained from a small amount of available data and provides a probabilistic out-of-sample performance guarantee. The computational tractability of the resulting DR-MPC problem was achieved via a set of reformulations. Finally, the performance of Wasserstein DR-MPC was demonstrated through numerical experiments on a nonlinear car-like vehicle model and a linearized quadrotor model. According to the simulation studies, even with a very small sample size ($N_k = 10$), Wasserstein DR-MPC successfully avoids randomly moving obstacles and limits the out-of-sample safety risk (in a probabilistic manner), unlike the popular SAA method.

Chapter 4

LEARNING-BASED DISTRIBUTIONALLY ROBUST MPC

4.1 Introduction

The adoption of learning-based decision-making tools for the intelligent operation of mobile robots and autonomous systems is rapidly growing because of advances in machine learning, sensing, and computing technologies. By learning its uncertain and dynamic environment, a robot can use additional information to improve the control performance. However, the accuracy of inference is often poor, as it is subject to the quality of the observations, statistical models, and learning methods. Employing inaccurately learned information in the robot's decision making may cause catastrophic system behaviors, in particular, leading to collision. The focus of this work is to develop an optimization-based method for safe motion control that is robust against errors in learned information about obstacles moving with unknown dynamics.

Learning-based control methods for mobile robots and autonomous systems can be categorized into two classes. The first class learns unknown system models, while the second class learns unknown environments. Control methods that learn unknown system dynamics typically use model predictive control (MPC) [62–66] and model-

based reinforcement learning (RL) [67–69]. 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 put more emphasis on "learning the environment" rather than "controlling the robot". In particular, for learning the behavior (or intention) of obstacles or other vehicles, several methods have been proposed that use inverse RL [70–72], imitation learning [73, 74], and Gaussian mixture models [75, 76], among others.

Our method is classified as the second since it learns the movement of obstacles. However, departing from the previous approaches, we emphasize the importance of "control" in correcting potential errors in "learning". The key idea is to determine the motion control action that is robust against errors in learned information about the obstacles' motion. Specifically, our method uses Gaussian process (GP) regression [77] to estimate the probability distribution of the obstacles' locations for future stages based on the current and past observations. To actively take into account the possibility that the learned distribution information may be inaccurate, we propose a novel MPC method that optimizes the motion control action subject to constraints on the *risk of unsafety* evaluated under the worst-case distribution in a so-called *ambiguity set*. Thus, the resulting control action will satisfy the risk constraints for safety even when the true distribution deviates from the learned one within the ambiguity set.

Unfortunately, the distributionally robust MPC (DR-MPC) problem is challenging to solve since the worst-case risk constraint involves an infinite-dimensional optimization problem over the ambiguity set of probability distributions. To resolve this issue, we propose a reformulation approach using (i) modern distributionally robust optimization techniques based on Kantorovich duality [1], (ii) the extremal representation of conditional value-at-risk, and (iii) a geometric expression of the distance to the union of half-spaces. The reformulated DR-MPC problem is finite-dimensional and can be efficiently solved by using existing nonlinear programming algorithms. Through simulations using a nonlinear car-like vehicle model for collision-avoidance

racing, we empirically show that, unlike the standard non-robust version, our method preserves safety even with moderate errors in the results of GP regression.

The remainder of the chapter is organized as follows. In Section 4.2, we present a GP regression approach to learning the motion of obstacles. In Section 4.3, we introduce the learning-based Wasserstein DR-MPC method with a tractable reformulation technique. The simulation results for collision-avoidance racing are presented in Section 4.4.

4.2 Learning the Movement of Obstacles Using Gaussian Processes

4.2.1 Obstacle Model

Unlike the previous chapter, here we consider a rigid body obstacle with some discrete-time dynamics. The obstacle state $\mathrm{x}_o(t) \in \mathbb{R}^{n_\mathrm{x}}$ is defined as the position and orientation of an arbitrary point on the obstacle. Thus, the obstacle state evolves with

$$\mathrm{x}_o(t+1) = \mathrm{x}_o(t) + T_o \mathrm{v}_o(\mathrm{x}_o(t)), \tag{4.2.1}$$

where $\mathrm{v}_o(\mathrm{x}_o(t)) \in \mathbb{R}^{n_\mathrm{x}}$ is the vector of the obstacle's velocity, and T_o is the sample time. For ease of exposition, we describe the case of a single obstacle, but our method is valid in multi-obstacle case as well.

Having the obstacle's state vector, as well as its geometric parameters, the region occupied by the obstacle at stage t can be modeled as a convex polytope defined by m number of half-spaces:

$$\mathcal{O}(t) := \{\mathbf{x} \in \mathbb{R}^{n_y} \mid c_t \mathbf{x} \leq d_t\}. \tag{4.2.2}$$

Here, $c_t \in \mathbb{R}^{m \times n_y}$ and $d_t \in \mathbb{R}^m$ are found from the geometry of the obstacle and the current state by $c_t = \mathrm{c}(\mathrm{x}_o(t))$ and $d_t = \mathrm{d}(\mathrm{x}_o(t))$.

For example, for a car-like obstacle in a 2D environment, the state can be chosen as the coordinate and angle of an arbitrary point on the obstacle. However, by symmetry,

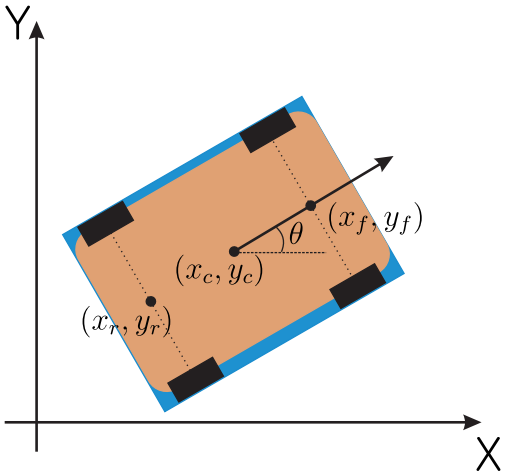


Figure 4.1: Car-like obstacle in 2D environment. By symmetry, the simplest motion pattern will be obtained for the following three candidates of state: $[x_r, y_r, \theta]^\top$, $[x_f, y_f, \theta]^\top$ and $[x_c, y_c, \theta]^\top$, where (x_r, y_r), (x_f, y_f) and (x_c, y_c) are the coordinates of the center of the rear axle, front axle, and center of mass, respectively, with θ as the heading angle. The region occupied by the vehicle is over-approximated by the blue rectangle.

the simplest motion pattern will be obtained for the three candidate states that are shown in Fig. 4.1. The region occupied by the obstacle is over-approximated as a rectangle, the parameters of which can be found using the geometry of the vehicle and any of the three states. To find the c_t and d_t, we need to know the exact expression of v_o. However, in practice it is impossible for a robot to have full knowledge of its environment, in particular, the behavior of the obstacle. For predicting the obstacle's motion, we use the GP regression approach introduced in the following subsection.

4.2.2 Gaussian Process Regression

GP regression is a nonparametric Bayesian approach to regression and infers a probability distribution over all possible values of a function given some training data [77]. A GP is a collection of random variables, any finite number of which have a joint Gaus-

sian distribution. In this work, GP regression is used for predicting the noisy velocity function $\mathrm{v}_o(\mathrm{x}_o(t))$ from previous observations of the obstacle's behavior.

We choose the training input data as $\hat{\mathrm{x}} = \{\mathrm{x}_o(t-1), \mathrm{x}_o(t-2), \ldots, \mathrm{x}(t-M)\}$, consisting of the obstacle's state for M previous stages. The corresponding measured velocities $\hat{\mathrm{v}}$ are selected as the training output data. In reality, we do not have access to function values; instead, the following noisy observations are available: for the ith observation

$$\hat{\mathrm{v}}^{(i)} = \mathrm{v}_o(\hat{\mathrm{x}}^{(i)}) + \varepsilon, \quad i = 1, \ldots, M,$$

where $\hat{\mathrm{x}}^{(i)} := \mathrm{x}_o(t-i)$, and ε is an i.i.d. zero-mean Gaussian noise with covariance $\Sigma^{\varepsilon} = \mathrm{diag}([\sigma^2_{\varepsilon,1}\ \sigma^2_{\varepsilon,2}, \ldots, \sigma^2_{\varepsilon,n_{\mathrm{x}}}])$.

Since the velocities in different dimensions are assumed to be independent, each of them can be learned individually. The dataset for the jth dimension is thus constructed as

$$\mathcal{D}_j = \big\{\big(\hat{\mathrm{x}}^{(i)}, \hat{\mathrm{v}}_j^{(i)}\big),\ i = 1, \ldots, M\big\}.$$

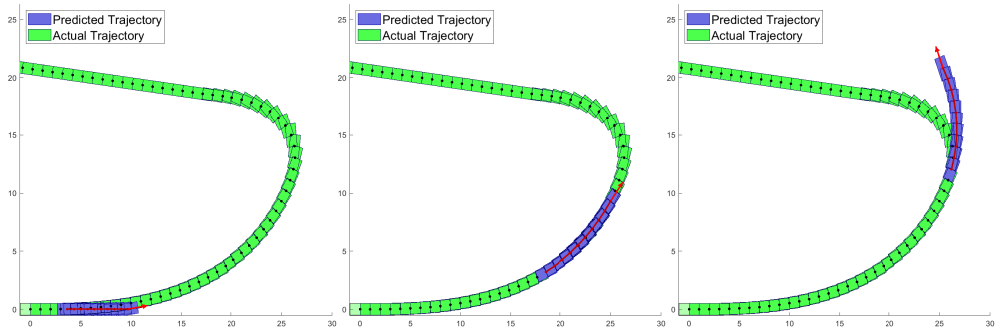
For each dimension of output $\mathrm{v}_o(\cdot)$, we specify a GP prior with mean function $m_j(x)$ and kernel function $k_j(x, x')$. In this thesis, we use an RBF kernel that is defined by

$$k_j(x, x') = \sigma^2_{f,j} \exp\Big[-\frac{1}{2}(x - x')^{\top} L_j^{-1} (x - x')\Big],$$

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

It follows that the joint distribution of the training output data $\hat{\mathrm{v}}_j$ and the output $\mathbf{v}_j$ at an arbitrary test point $\mathbf{x}$ is given by

$$\begin{bmatrix} \hat{\mathrm{v}}_j \\ \mathbf{v}_j \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_j(\hat{\mathrm{x}}) \\ m_j(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma^2_{\varepsilon,j} I & K_j(\hat{\mathrm{x}}, \mathbf{x}) \\ K_j(\mathbf{x}, \hat{\mathrm{x}}) & k_j(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right),$$



(a) $t = 5$ (b) $t = 16$ (c) $t = 30$

Figure 4.2: Means of predicted trajectories for 10 steps at different time stages. In (a) the GP dataset is small and contains only 4 elements, so the prediction is not accurate. In (b) and (c) there is enough data to learn the motion pattern, thus the predicted trajectory is close to the actual one.

where $K_j^{(l)}(\hat{\mathrm{x}}, \mathbf{x}) = k_j(\hat{\mathrm{x}}^{(l)}, \mathbf{x})$, and $K_j(\mathbf{x}, \hat{\mathrm{x}}) = K_j(\hat{\mathrm{x}}, \mathbf{x})^\top$. As a result, the posterior distribution of the output in the jth dimension at an arbitrary test point $\mathbf{x}$ conditioned on the observed data is Gaussian, with the following mean and covariance:

$$\boldsymbol{\mu}_{\mathrm{v}}^j(\mathbf{x}) := m_j(\mathbf{x}) + K_j(\mathbf{x}, \hat{\mathrm{x}})(K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma_{\varepsilon,j}^2 I)^{-1}(\hat{\mathrm{v}}_j - m_j(\hat{\mathrm{x}})), \tag{4.2.3}$$

$$\boldsymbol{\Sigma}_{\mathrm{v}}^j(\mathbf{x}) = k_j(\mathbf{x}, \mathbf{x}) - K_j(\mathbf{x}, \hat{\mathrm{x}})(K_j(\hat{\mathrm{x}}, \hat{\mathrm{x}}) + \sigma_{\varepsilon,j}^2 I)^{-1} K^j(\hat{\mathrm{x}}, \mathbf{x}). \tag{4.2.4}$$

The resulting GP approximation of v_o is then given by

$$\mathbf{v}(\mathbf{x}) \sim \mathcal{GP}(\boldsymbol{\mu}_{\mathrm{v}}(\mathbf{x}), \boldsymbol{\Sigma}_{\mathrm{v}}(\mathbf{x})),$$

where $\boldsymbol{\mu}_{\mathrm{v}}(\mathbf{x}) = [\boldsymbol{\mu}_{\mathrm{v}}^1(\mathbf{x}), \ldots, \boldsymbol{\mu}_{\mathrm{v}}^{n_{\mathrm{x}}}(\mathbf{x})]^\top$, and $\boldsymbol{\Sigma}_{\mathrm{v}}(\mathbf{x}) = \mathrm{diag}([\boldsymbol{\Sigma}_{\mathrm{v}}^1(\mathbf{x}), \ldots, \boldsymbol{\Sigma}_{\mathrm{v}}^{n_{\mathrm{x}}}(\mathbf{x})])$.

4.2.3 Prediction of the Obstacle's Motion

Assuming that $\mathrm{x}_o(0) \sim \mathcal{N}(\mu_{\mathrm{x}}(0), \mathbf{0})$, it is straightforward to check that $\mathrm{x}_o(t)$ is normally distributed at each stage t with mean $\mu_{\mathrm{x}}(t)$ and covariance $\Sigma_{\mathrm{x}}(t)$ to be specified. Having the posterior of the velocity vector, the state of the obstacle at the next stage

can be predicted by considering the following joint distribution of the state and velocity vectors [66]:

$$\begin{bmatrix} \mathrm{x}_o(t) \\ \mathbf{v}(t) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathrm{x}}(t) \\ \mu_{\mathrm{v}}(t) \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathrm{x}}(t) & \Sigma_{\mathrm{xv}}(t) \\ \Sigma_{\mathrm{vx}}(t) & \Sigma_{\mathrm{v}}(t) \end{bmatrix} \right).$$

Following procedures in [78] and [79] and applying the first-order Taylor approximation to (4.2.3) and (4.2.4) with Gaussian input $\mathrm{x}_o(t) \sim \mathcal{N}(\mu_{\mathrm{x}}(t), \Sigma_{\mathrm{x}}(t))$ yields the following approximate mean and covariance functions:

$$\begin{aligned} \tilde{\mu}_{\mathrm{v}}(t) &= \boldsymbol{\mu}_{\mathrm{v}}(\mu_{\mathrm{x}}(t)) \\ \tilde{\Sigma}_{\mathrm{v}}(t) &= \boldsymbol{\Sigma}_{\mathrm{v}}(\mu_{\mathrm{x}}(t)) + \nabla \boldsymbol{\mu}_{\mathrm{v}}(\mu_{\mathrm{x}}(t)) \Sigma_{\mathrm{x}}(t) \nabla \boldsymbol{\mu}_{\mathrm{v}}(\mu_{\mathrm{x}}(t))^{\top} \\ \tilde{\Sigma}_{\mathrm{xv}}(t) &= \Sigma_{\mathrm{x}}(t) \nabla \boldsymbol{\mu}_{\mathrm{v}}(\mu_{\mathrm{x}}(t))^{\top}. \end{aligned} \tag{4.2.5}$$

Now, it follows from (4.2.1) that the obstacle's state at the next stage is also normally distributed with the following mean and covariance:

$$\begin{aligned} \mu_{\mathrm{x}}(t+1) &= \mu_{\mathrm{x}}(t) + T_o \mu_{\mathrm{v}}(t) \\ \Sigma_{\mathrm{x}}(t+1) &= \Sigma_{\mathrm{x}}(t) + T_o^2 \Sigma_{\mathrm{v}} + T_o(\Sigma_{\mathrm{xv}} + \Sigma_{\mathrm{vx}}). \end{aligned} \tag{4.2.6}$$

Using (4.2.5) and (4.2.6), the approximate mean and variance of $\mathrm{x}_o(t)$ can be updated.

Having the inferred or predicted obstacle state $\mathrm{x}_o(t)$, it is straightforward to obtain d_t and c_t in (4.2.2) as $d_t = \mathrm{d}(\mathrm{x}_o(t))$ and $c_t = \mathrm{c}(\mathrm{x}_o(t))$. An example of predicting the motion of an obstacle is shown in Fig. 4.2, where a car-like vehicle is chosen as the obstacle with unknown dynamics. GP regression is used to predict the trajectory of the vehicle for the next 10 stages. As shown in Fig. 4.2a, the predicted mean in an early stage ($t = 5$) deviates from the actual trajectory, as there were no observations available. As more data are collected, the robot better learns the motion pattern of the car-like obstacle. As a result, in Figures 4.2b the difference between the predicted mean and the actual one is small.

However, in practice, the motion predicted by GP regression can be quite different from the actual movement of an obstacle, for example, when it abruptly changes the

heading angle, as in the case of Fig. 4.2c. To guarantee safety even when learning fails, we propose a distributionally robust motion control tool in the following section.

4.3 Gaussian Process based Wasserstein DR-MPC

In this section we combine the Wasserstein Distributionally Robust MPC formulation with the GP prediction derived in Section 4.2.3 and extend it to incorporate the rotation motion.

As previously, we define the loss of safety as the deviation of the robot's position from the safe region $\mathcal{Y}(t)$ and apply Lemma 1 from Section 3.3.1 to get

$$\operatorname{dist}(y(t), \mathcal{Y}(t)) := \min_{a \in \mathcal{Y}(t)} \|y(t) - a\|_2 = \min_{j=1,\dots,m} \left\{ \frac{\left(d_{t,j} - c_{t,j} y(t)\right)^+}{\|c_{t,j}\|_2} \right\}. \quad (4.3.1)$$

where $d_{t,j}$ is the jth element of d_t, $c_{t,j}$ is the jth row of c_t.

We now let

$$G_{t,j} := -\frac{c_{t,j}}{\|c_{t,j}\|_2}, \quad g_{t,j} := \frac{d_{t,j}}{\|c_{t,j}\|_2}. \quad (4.3.2)$$

The safe region in (4.3.1) depends on d_t and c_t, which define the region occupied by the obstacle at stage t. Unfortunately, the distribution of these two parameters is unknown and challenging to directly identify in practice. However, having sample data $\{\tilde{\mathrm{x}}_o^{(1)}(t), \tilde{\mathrm{x}}_o^{(2)}(t), \dots, \tilde{\mathrm{x}}_o^{(N)}(t)\}$ generated according to the learned distribution of $\mathrm{x}_o(t)$, it is possible to obtain samples of g_t and G_t using

$$\tilde{d}_t^{(i)} := \mathrm{d}(\tilde{\mathrm{x}}_o^{(i)}(t)), \quad \tilde{c}_t^{(i)} := \mathrm{c}(\tilde{\mathrm{x}}_o^{(i)}(t)). \quad (4.3.3)$$

Using the samples (4.3.3) of $\tilde{d}_t^{(i)}$ and $\tilde{c}_t^{(i)}$, we can then generate samples $\{\tilde{w}\}_{i=1}^N = \{(\tilde{G}_{t,j}^{(i)}, \tilde{g}_{t,j}^{(i)})\}_{i=1}^N$ of $(G_{t,j}, g_{t,j})$ according to the definition above. Let Q_t be the joint empirical distribution of $(G_t, g_t) \in \mathbb{W} \subseteq \mathbb{R}^{m(n_y+1)}$ constructed using the sample data, i.e., $\mathrm{Q}_t := \sum_{i=1}^N \boldsymbol{\delta}_{(\tilde{G}_t^{(i)}, \tilde{g}_t^{(i)})}$, where $\boldsymbol{\delta}_{\boldsymbol{x}}$ denotes the Dirac delta measure concentrated at $\boldsymbol{x}$.

We can then use the sample data to design an empirical distribution and impose the same distributionally robust risk constraint as in previous chapter to limit the risk of unsafety:

$$\sup_{\mathrm{P}_t \in \mathbb{D}_t} \mathrm{CVaR}_{\alpha}^{\mathrm{P}_t}[\mathrm{dist}(y(t), \mathcal{Y}(t))] \leq \delta. \tag{4.3.4}$$

Here, the left-hand side of the inequality represents the worst-case CVaR when the joint distribution P_t of (G_t, g_t) lies in a given *ambiguity set* $\mathbb{D}_t$. Thus, any motion control action that satisfies (4.3.4) can meet the original risk constraint under any distribution error characterized by $\mathbb{D}_t$.

Next, the following proposition can be used to reformulate the distributionally robust risk constraint (4.3.4) in a conservative manner.

Proposition 3. *Suppose that the uncertainty set is a compact convex polytope, i.e.* $\mathbb{W} := \{w \in \mathbb{R}^{m(n_y+1)} \mid Hw \leq h\}$*, where* $H \in \mathbb{R}^{q \times m(n_y+1)}$ *and* $h \in \mathbb{R}^q$*. Then, the following equality holds:*

$$\begin{aligned}
&\sup_{\mathrm{P}_t \in \mathbb{D}_t} \mathrm{CVaR}_{\alpha}^{\mathrm{P}_t}[\mathrm{dist}(y(t), \mathcal{Y}(t)] \\
&\quad \leq \inf_{z,\lambda,s,\rho} z + \frac{1}{1-\alpha}\left[\lambda\theta + \sum_{i=1}^{N} s_i\right] \\
&\qquad s.t\ \langle \rho_i, Q\tilde{w}^{(i)} \rangle + \langle \gamma_i, h - H\tilde{w}^{(i)} \rangle \leq s_i + z \\
&\qquad\quad \langle \eta_i, h - H\tilde{w}^{(i)} \rangle \leq s_i + z \\
&\qquad\quad \langle \zeta_i, h - H\tilde{w}^{(i)} \rangle \leq s_i \\
&\qquad\quad \|H^\top \gamma_i - Q^\top \rho_i\|_* \leq \lambda \\
&\qquad\quad \|H^\top \eta_i\|_* \leq \lambda \\
&\qquad\quad \|H^\top \zeta_i\|_* \leq \lambda \\
&\qquad\quad \langle \rho_i, e_m \rangle = 1 \\
&\qquad\quad \gamma_i \geq 0,\ \rho_i \geq 0,\ \eta_i \geq 0,\ \zeta_i \geq 0,
\end{aligned} \tag{4.3.5}$$

where all the constraints hold for $i = 1, \ldots, N$*, and* $e \in \mathbb{R}^m$ *is a vector of all ones.* $\rho_{i,j}$ *represents the* j*th element of* ρ_i *and* y_l *is the* l*th element of* y*,* $Q \in \mathbb{R}^{m \times m(n_y+1)}$ *is*

a block diagonal matrix with vector $[y^\top, 1]$ *in the diagonal.*

Proof. Its proof follows directly from Lemma 2, and Proposition 2 in Section 3.3.2 by setting $p_j(y, w) = G_j y + g_j = Q_j w$, where Q_j is the jth row of matrix Q, $w \in \mathbb{R}^{m(n_y+1)}$ is a random vector consisting of all elements of G and g. □

Note that this optimization problem on the right-hand side is finite-dimensional, unlike the original one on the left-hand side. Specifically, according to Proposition 3, the GP-based DR-MPC problem can be reformulated as follows:

$$\inf_{\substack{\mathbf{u},\boldsymbol{\xi},\mathbf{y},\mathbf{z},\boldsymbol{\lambda},\\ \mathbf{s},\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{\eta},\boldsymbol{\zeta}}} J(\xi(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) \tag{4.3.6a}$$

$$\text{s.t. } \xi_{k+1} = f(\xi_k, u_k) \tag{4.3.6b}$$

$$y_k = h(\xi_k, u_k) \tag{4.3.6c}$$

$$\xi_0 = \xi(t) \tag{4.3.6d}$$

$$z_k + \frac{1}{1-\alpha}\left[\lambda_k \theta + \frac{1}{N_k}\sum_{i=1}^{N_k} s_{k,i}\right] \leq \delta \tag{4.3.6e}$$

$$\langle \rho_{k,i}, Q_k \tilde{w}_k^{(i)} \rangle + \langle \gamma_i, h - H\tilde{w}_k^{(i)} \rangle \leq s_{k,i} + z_k \tag{4.3.6f}$$

$$\langle \eta_{k,i}, h - H\tilde{w}_k^{(i)} \rangle \leq s_{k,i} + z_k \tag{4.3.6g}$$

$$\langle \zeta_{k,i}, h - H\tilde{w}_k^{(i)} \rangle \leq s_{k,i} \tag{4.3.6h}$$

$$\|H^\top \gamma_{k,i} - Q_k^\top \rho_{k,i}\|_* \leq \lambda_k \tag{4.3.6i}$$

$$\|H^\top \eta_{k,i}\|_* \leq \lambda_k \tag{4.3.6j}$$

$$\|H^\top \zeta_{k,i}\|_* \leq \lambda_k \tag{4.3.6k}$$

$$\langle \rho_{k,i}, e_m \rangle = 1 \tag{4.3.6l}$$

$$\gamma_i \geq 0,\ \rho_i \geq 0,\ \eta_i \geq 0,\ \zeta_i \geq 0, \tag{4.3.6m}$$

$$z_k \in \mathbb{R}, \tag{4.3.6n}$$

$$\xi_k \in \Xi,\ u_k \in \mathcal{U}, \tag{4.3.6o}$$

where (4.3.6b) and $u_k \in \mathcal{U}$ in (4.3.6o) should hold for $k = 0, \ldots, K-1$, (4.3.6c) and $\xi_k \in \Xi$ in (4.3.6o) should hold for $k = 0, \ldots, K$, and all the other constraints should be satisfied for $k = 1, \ldots, K$ and $i = 1, \ldots, N$.

Now, let's suppose that $\mathbb{W} = \mathbb{R}^{m(n_y+1)}$ and use ℓ_2-norm to measure distances in the uncertainty space. Then, the right-hand side problem (4.3.6) simplifies to

$$\inf_{\substack{\mathbf{u},\boldsymbol{\xi},\mathbf{y},\mathbf{z},\\ \boldsymbol{\lambda},\mathbf{s},\boldsymbol{\rho}}} J(\xi(t), \mathbf{u}) := \sum_{k=0}^{K-1} r(\xi_k, u_k) + q(\xi_K) \tag{4.3.7a}$$

$$\text{s.t. } \xi_{k+1} = f(\xi_k, u_k) \tag{4.3.7b}$$

$$y_k = h(\xi_k, u_k) \tag{4.3.7c}$$

$$\xi_0 = \xi(t) \tag{4.3.7d}$$

$$z_k + \frac{1}{1-\alpha}\left[\lambda_k \theta + \frac{1}{N_k}\sum_{i=1}^{N_k} s_{k,i}\right] \leq \delta \tag{4.3.7e}$$

$$\langle \rho_{k,i}, \tilde{G}_k^{(i)} y_k + \tilde{g}_k^{(i)} \rangle \leq s_{k,i} + z_k \tag{4.3.7f}$$

$$s_{k,i} + z_k \geq 0 \tag{4.3.7g}$$

$$s_{k,i} \geq 0 \tag{4.3.7h}$$

$$\sum_{j=1}^{m} \rho_{k,i,j}^2 \Big(\sum_{l=1}^{n_y} y_{k,l}^2 + 1 \Big) \leq \lambda_k^2 \tag{4.3.7i}$$

$$\lambda_k \geq 0 \tag{4.3.7j}$$

$$\langle \rho_{k,i}, e \rangle = 1 \tag{4.3.7k}$$

$$\rho_{k,i} \geq 0 \tag{4.3.7l}$$

$$z_k \in \mathbb{R}, \tag{4.3.7m}$$

$$\xi_k \in \Xi,\ u_k \in \mathcal{U}, \tag{4.3.7n}$$

As desired, the reformulated problem is finite-dimensional. However, it is a non-convex optimization problem due to the constraints (4.3.7f) and (4.3.7i) even when the

Algorithm 1: Learning-based DR-MPC at stage t

Input: $\xi(t), \mathrm{x}_o(t), \hat{\mathrm{x}}^{(i)}, \hat{\mathrm{v}}^{(i)}, i = 1, \ldots, M$;

$\mathcal{D}_j := \big\{(\hat{\mathrm{x}}^{(i)}, \hat{\mathrm{v}}_j^{(i)}),\ i = 1, \ldots, M\big\},\ j = 1, \ldots, n_{\mathrm{x}}$;

Initialize $\mu_{\mathrm{x}}(0) := \mathrm{x}_o(t)$, $\Sigma_{\mathrm{x}}(0) := \mathbf{0}$;

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

  Compute $\tilde{\mu}_{\mathrm{v}}(k)$, $\tilde{\Sigma}_{\mathrm{v}}(k)$ and $\tilde{\Sigma}_{\mathrm{xv}}$ from (4.2.5);

  Update $\mu_{\mathrm{x}}(k+1)$ and $\Sigma_{\mathrm{x}}(k+1)$ from (4.2.6);

  Generate a sample $\{\tilde{\mathrm{x}}_o^{(1)}(k+1), \ldots, \tilde{\mathrm{x}}_o^{(N)}(k+1)\}$ from $\mathcal{N}(\mu_{\mathrm{x}}(k+1), \Sigma_{\mathrm{x}}(k+1))$;

  Compute $\tilde{G}_{k+1}^{(i)}$ and $\tilde{g}_{k+1}^{(i)}$, $i = 1, \ldots, N$ using (4.3.3) and (4.3.2);

Solve (4.3.6) to obtain $\mathbf{u}^*$;

return $u(t) = u_0^*$;

system dynamics and the output equation are affine and the cost function is convex. A locally optimal solution to this problem can be efficiently computed by using existing nonlinear programming algorithms such as interior-point methods (e.g., [56]).

The overall learning-based DR-MPC at stage t is shown in Algorithm 1. At each stage, the current states of the robot and the obstacle as well as M past observations $\{(\hat{\mathrm{x}}^{(i)}, \hat{\mathrm{v}}^{(i)})\}_{i=1}^{M}$ of the obstacle's position and velocity are taken as the input data. Then, the obstacle's movement for future stages is learned by GP regression, and is used in the DR-MPC problem (4.3.6). The first element of locally optimal solution $\mathbf{u}^*$ is taken as the motion control action for the robot at the current stage. Note that at stage $t = 0$, the dataset $\mathcal{D}$ consists of all zeros. As time goes on, new observations are added to the dataset for GP regression. During the update, old observations are removed so that only M latest data are stored.

4.4 Numerical Experiments

In this section, we present simulation results to demonstrate the performance of our motion control method. In our experiments we consider an on-road vehicle following the centerline of the track, while avoiding another on-road vehicle, which future trajectory is learned by GP. Then the GP-based WDR-MPC is applied to use the learned trajectories in limiting the risk of unsafety.

As in the previous experiments, all the simulations were conducted on a PC with 3.70 GHz Intel Core i7-8700K processor and 32 GB RAM. The optimization problem was modeled in AMPL and solved using interior-point method-based solver IPOPT.

The vehicle we control navigates in a 2D environment with the following dynamics [80]:

$$\begin{aligned}
\dot{x}^r(t) &= v^r(t)\cos(\theta^r(t)+\beta^r(t)) \\
\dot{y}^r(t) &= T_s v^r(t)\sin(\theta^r(t)+\beta^r(t)) \\
\dot{\theta}^r(t) &= v^r(t)\frac{\sin(\beta^r(t))}{l_r} \\
\dot{\beta}^r(t) &= \tan^{-1}\Big(\frac{l_r}{l_r+l_f}\tan\delta^r(t)\Big),
\end{aligned} \tag{4.4.1}$$

where $x^r(t)$ and $y^r(t)$ are the coordinates of the vehicle's center of gravity, $\theta^r(t)$ is the heading angle, $\beta^r(t)$ is the current velocity angle. The control inputs are velocity $v^r(t)$ and steering angle $\delta^r(t)$. The coefficients l_f and l_r represent the distances from the center of gravity to the front and rear wheels, respectively. Throughout our simulations we let $l_r = l_f = 2$. We also impose the following state and control constraints:

$$v^r(k) \in [0, 30], \quad u^r(k) \in [-\pi/6, \pi/6] \quad \forall k.$$

The vehicle is controlled to follow the centerline of the track, while avoiding two dynamic obstacles. The centerline is thus taken as the reference trajectory y^{ref} in the cost function. The two obstacles are rectangular car-like vehicles with size 2×1. It is straightforward to check that for both obstacles g_k and G_k are easily found from

the state that consists of the vehicle's center of mass and its heading angle. In our experiments, we set $Q = P = I$ and $R = 0.01I$. The sampling time T_s and T_o are set to be 0.01, and the MPC horizon is chosen as $K = 5$. The risk tolerance level and the confidence level were selected as $\delta = 0.01$ and $\alpha = 0.95$, respectively. As proposed in the previous chapter, we use support of $\mathbb{W} = \mathbb{R}^{m(n_y+1)}$ and ℓ_2-norm to measure the distance in the uncertainty space.

To evaluate the performance of learning-based DR-MPC, we compare it to its non-robust counterpart obtained by sample average approximation (SAA).

Fig. 4.3 shows the resulting trajectories for different sizes of the Wasserstein ambiguity set compared to the SAA version (SAA-MPC) with $N = 50$ samples. At each stage, the dataset for GP regression is updated to keep only the latest $M = 20$ observations.

In the early stages, the robotic vehicle follows the centerline while predicting the future motion of the obstacles. As shown in Fig. 4.3a, when reaching one obstacle that abruptly changes its heading angle at $t = 13$, the vehicle tries to avoid it. In the case of SAA-MPC, the vehicle collides with the obstacle because the distributional information learned by GP regression is inaccurate. As a result, the risk constraint is violated and the MPC problem becomes infeasible. Meanwhile, the vehicle controlled by our method successfully bypasses the obstacle. The safety margin increases with the radius θ of the Wasserstein ambiguity set.

Fig. 4.3b shows the situation at $t = 38$, where the vehicle controlled by our method continues to follow the reference trajectory for all θ's. Meanwhile, the GP is not well enough to be able to predict the motion of the obstacles around the corners, although it shows good performance when there is no sudden change in the obstacle's movement. As shown in Fig. 4.3c, at $t = 67$ the second obstacle interferes with the path of the vehicle. Similar to the previous obstacle, the vehicle controlled by DR-MPC avoids the obstacle for all θ's. In the case of the smallest radius of $\theta = 4 \times 10^{-5}$, the vehicle chooses to take aggressive action while satisfying the risk constraint. As the Wasser-

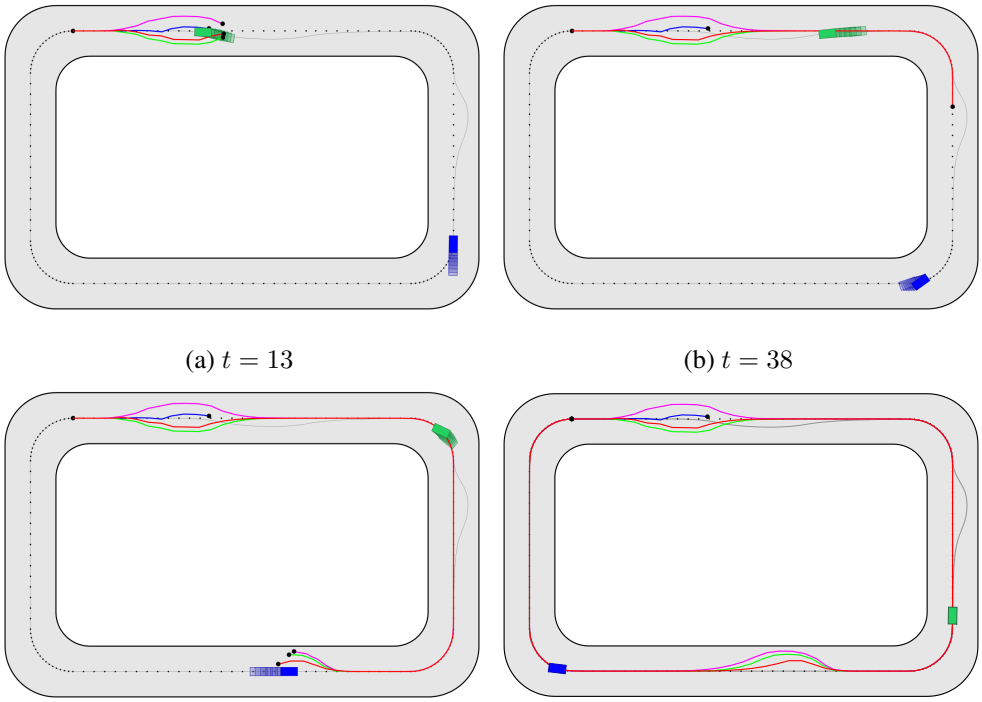


Figure 4.3: Trajectories of the vehicle controlled by SAA-MPC and DR-MPC with $\theta = 4 \times 10^{-5}, 5 \times 10^{-5}$, and 5.5×10^{-5}. The current vehicle position is marked with a black dot. The green and blue rectangles represent the two obstacles, while the transparent ones are the K steps-ahead prediction of the obstacles, obtained via GP regression. The reference centerline for the vehicle is displayed with points, while the thin grey curve is the actual trajectory of the obstacles.

Table 4.1: Accumulated cost, lap time, and average computation time for the nonlinear car-like vehicle motion control with $N = 50$, $\delta = 0.01$, and $\alpha = 0.95$

	SAA	DR-MPC (θ)		
		4×10^{-5}	5×10^{-5}	5.5×10^{-5}
Accumulated Cost	$+\infty$	491.79	594.68	703.59
Lap Time (sec)	-	105.26	109.45	110.31
Avg. Run Time (sec)	-	0.6572	0.6767	0.6942

stein ambiguity set increases, i.e., θ increases, the robot makes a more conservative (i.e., safer) decision, inducing a bigger safety margin. Fig. 4.3d displays the trajectories for all cases after the vehicle completes one lap. Note that only the non-robust SAA version failed to complete the lap due to collision, while our method succeeded to do so for all θ's.

In summary, we conclude that the proposed distributionally robust method successfully preserves safety even with moderate errors in the learning results. In the case of very small ambiguity sets (e.g., $\theta = 4 \times 10^{-5}$), the resulting control action may be too aggressive to guarantee safety when the learning errors are significant. Whereas, for $\theta = 5.5 \times 10^{-5}$, the vehicle deviates too much from the reference trajectory, inducing a large cost. Based on our experiments, $\theta = 5 \times 10^{-5}$ may be selected for a good tradeoff between safety and cost.

Table 4.1 shows the accumulated cost and the amount of time for completing one lap on the track, and the average computation time required for solving a single DR-MPC problem (4.3.6). As expected, both of the total cost and the lap time increase with θ since the vehicle controlled by DR-MPC with larger θ is more conservative and deviates further from the reference trajectory. Computation time is small in all cases although a nonconvex optimization problem is solved in each iteration. This result shows the potential of using our distributionally robust method in real-time applications.

4.5 Conclusions

We have proposed a distributionally robust decision-making tool for safe motion control of robotic vehicles in an environment with dynamic obstacles. Our DR-MPC method limits the risk of unsafety even with moderate errors in the obstacle's motion predicted by GP regression. For computational tractability, we have also developed a reformulation approach exploiting modern distributionally robust optimization techniques. The experimental results demonstrate the safety-preserving capability of our method under moderate learning errors and the potential for real-time application.

Chapter 5

CONCLUSIONS AND FUTURE WORK

In this thesis we developed a risk-aware motion planning and control approach for learning-based autonomous systems. First, we discussed a Model Predictive Control (MPC) method based on the Conditional Value-at-Risk (CVaR) for limiting the risk of unsafety of the robot by incorporating samples of the uncertain environment via Sample Average Approximation (SAA). The problem was then reformulated into a linearly constrained mixed integer convex program. Second, a novel Wasserstein Distributionally Robust MPC (DR-MPC) method was developed for ensuring safety when the empirical distribution obtained from a small amount of data deviates from the actual one within a Wasserstein ball. Then a computationally tractable form was developed by employing Kantorovich duality and the geometry. of the obstacles. The resulting approach provides a theoretically justified probabilistic out-of-sample performance guarantee, missing in the SAA-based method. Third, the Wasserstein DR-MPC method was combined with Gaussian Processes (GP) for learning the future motion of the obstacles.

The experimental results show that even for small sample size, Wasserstein DR-MPC successfully avoids randomly moving obstacles, unlike the SAA-based method. Also, we showed that for carefully chosen Wasserstein radius the out-of-sample risk is limited even just using a few samples. The simulations for GP-based DR-MPC demon-

strate the capability of our method to limit the risk of unsafety when the learning outcome is uncertain.

In the future, an explicit MPC method can be employed to reduce the real-time computations. In addition, the proposed method can be extended to enhance the capability of fast adaptive reactions, especially when considering sudden motion changes, and to address partial observability. Such extension is possible by an Inverse Reinforcement Learning-based approach, which can learn the reward function maximized for obtaining the observed trajectories subject to the unknown system dynamics, which in turn, can be learned via Neural Networks.

Bibliography

[1] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, pp. 115–166, 2018.

[2] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[3] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[4] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: a modular framework for fast and guaranteed safe motion planning," in *IEEE 56th Annual Conference on Decision and Control*, 2017.

[5] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[6] D. Fridovich-Keil, S. L. Herbert, J. F. Fisac, S. Deglurkar, and C. J. Tomlin, "Planning, fast and slow: A framework for adaptive real-time safe trajectory planning," in *IEEE International Conference on Robotics and Automation*, 2018.

[7] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[8] 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, 2012.

[9] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Autonomous Robots*, vol. 39, pp. 555–571, 2015.

[10] B. Luders, M. Kothari, and J. How, "Chance constrained RRT for probabilistic robustness to environmental uncertainty," in *AIAA Guidance, Navigation, and Control Conference*, 2010.

[11] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011.

[12] A. Jasour, N. S. Aybat, and C. M. Lagoa, "Semidefinite programming for chance constrained optimization over semialgebraic sets," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1411–1440, 2015.

[13] A. M. Jasour and B. C. Williams, "Risk contours map for risk bounded motion planning under perception uncertainties," in *Robotics: Science and Systems*, 2019.

[14] T. Summers, "Distributionally robust sampling-based motion planning under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[15] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[16] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[17] ——, "Conditional value-at-risk for general loss distribution," *Journal of Banking & Finance*, vol. 26, pp. 1443–1471, 2002.

[18] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical Finance*, vol. 9, no. 3, pp. 203–228, 1999.

[19] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *International Symposium on Robotics Research*, 2017.

[20] S. Samuelson and I. Yang, "Safety-aware optimal control of stochastic systems using conditional value-at-risk," in *American Control Conference*, 2018.

[21] T. Schouwenaars, A. Stubbs, J. Paduano, and E. Feron, "Multivehicle path planning for nonline-of-sight communication," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 269–290, 2006.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] ——, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *IEEE Conference on Decision and Control*, 2010.

[24] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[25] C. W. Miller and I. Yang, "Optimal control of conditional value-at-risk in continuous time," *SIAM Journal on Control and Optimization*, vol. 55, no. 2, pp. 856–884, 2017.

[26] S. Singh, Y.-L. Chow, A. Majumdar, and M. Pavone, "A framework for time-consistent, risk-sensitive model predictive control: Theory and algorithms," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2905–2912, 2019.

[27] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, 2nd ed. SIAM, 2014.

[28] D. Bertsimas, V. Gupta, and N. Kallus, "Robust sample average approximation," *Mathematical Programming*, vol. 171, pp. 217–282, 2018.

[29] I. Yang, "Wasserstein distributionally robust stochastic control: A data-driven approach," *arXiv preprint arXiv:1812.09808.*, 2018.

[30] A. Vecchietti, S. Lee, and I. E. Grossmann, "Modeling of discrete/continuous optimization problems: Characterization and formulation of disjunctions and their relaxations," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 433–448, 2003.

[31] P. Bonami, J. Lee, S. Leyffer, and A. Wächter, "On branching rules for convex mixed-integer nonlinear optimization," *Journal of Experimental Algorithmics*, vol. 18, pp. 2–6, 2013.

[32] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, 1986.

[33] M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma, "Polyhedral approximation in mixed-integer convex optimization," *Mathematical Programming*, vol. 172, no. 1-2, pp. 139–168, 2018.

[34] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, vol. 5, no. 2, pp. 186–204, 2008.

[35] T. Achterberg, "SCIP: solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.

[36] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*. Springer, 2006, pp. 35–59.

[37] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1-4, pp. 1–12, 2013.

[38] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[39] A. Jasour, N. S. Aybat, and C. M. Lagoa, "Semidefinite programming for chance constrained optimization over semialgebraic sets," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1411–1440, 2015.

[40] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-sensitive and robust decision-making: a CVaR optimization approach," in *Advances in Neural Information Processing Systems*, 2015.

[41] 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.

[42] G. C. Calafiore and L. El Ghaoui, "On distributionally robust chance-constrained linear programs," *Journal of Optimization Theory and Applications*, vol. 130, no. 1, pp. 1–22, 2006.

[43] 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.

[44] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[45] C. Zhao and Y. Guan, "Data-driven risk-averse stochastic optimization with Wasserstein metric," *Operations Research Letters*, vol. 46, no. 2, 2018.

[46] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with Wasserstein distance," *arXiv:1604.02199*, 2016.

[47] 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.

[48] G. Bayraksan and D. K. Love, "Data-driven stochastic programming using phi-divergences," *Tutorials in Operations Research*, pp. 1–19, 2015.

[49] A. R. Hota, A. Cherukuri, and J. Lygeros, "Data-driven chance constrained optimization under Wasserstein ambiguity sets," in *American Control Conference*, 2019.

[50] J. Coulson, J. Lygeros, and F. Dörfler, "Regularized and distributionally robust data-enabled predictive control," *arXiv preprint arXiv:1903.06804*, 2019.

[51] H. Xu and S. Mannor, "Distributionally robust Markov decision processes," *Mathematics of Operations Research*, vol. 37, no. 2, pp. 288–300, 2012.

[52] I. Yang, "A convex optimization approach to distributionally robust Markov decision processes with Wasserstein distance," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 164–169, 2017.

[53] ——, "A dynamic game approach to distributionally robust safety specifications for stochastic systems," *Automatica*, vol. 94, pp. 94–101, 2018.

[54] 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.

[55] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[56] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.

[57] A. Hakobyan and I. Yang, "Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk," *arXiv preprint arXiv:2001.04727*, 2020.

[58] 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.

[59] R. Fourer, D. M. Gay, and B. W. Kernighan, "A modeling language for mathematical programming," *Management Science*, vol. 36, no. 5, pp. 519–554, 1990.

[60] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[61] R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[62] 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.

[63] S. Di Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic MPC with learning for driver-predictive vehicle control and its application to HEV energy management," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2013.

[64] 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.

[65] 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.

[66] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, 2019.

[67] 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.

[68] 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.

[69] 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.

[70] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *IEEE International Conference on Robotics and Automation*, 2015.

[71] 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.

[72] 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 Robotics Research*, vol. 36, no. 10, pp. 1073–1087, 2017.

[73] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *IEEE Intelligent Vehicles Symposium*, 2017.

[74] 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.

[75] 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.

[76] 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.

[77] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[78] A. Girard, C. Rasmussen, and R. Murray-Smith, “Gaussian process priors with uncertainty inputs: multiple-step-ahead prediction,” *Technical Report TR-2002–119, Dept. of Computer Science*, 2002.

[79] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, “Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting,” in *Advances in Neural Information Processing Systems*, 2003.

[80] P. Polack, F. Altché, B. d’Andréa Novel, and A. de La Fortelle, “The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?” in *IEEE Intelligent Vehicles Symposium*, 2017.

초 록

본 연구에서 자율 시스템이 알려지지 않은 확률 분포로 랜덤하게 움직이는 장애물을 피하기 위한 위험 인식을 고려하는 모션 제어 기법을 개발한다. 따라서 본 논문에서 안전성과 보수성을 체계적으로 조절하는 새로운 Model Predictive Control (MPC) 방법을 제안한다. 본 방벙의 핵심 요소는 MPC 문제의 안전성 리스크를 제한하는 Conditional Value-at-Risk (CVaR)라는 리스크 척도이다. 안전성 리스크를 계산하기 위해 제한된 양의 표본 데이터를 이용하여 얻어진 경험적 분포를 사용하는 Sample Average Approximation (SAA)을 적용한다.

또한, 경험적 분포로부터 실제 분포가 Ambiguity Set라는 집합 내에서 벗어나도 리스크를 제한하는 방법을 제안한다. Ambiguity Set를 Wasserstein 거리로 측정된 반지름을 가진 통계적 공으로 선택함으로써 훈련 데이터와 독립적으로 생성된 새로운 샘플 데이터를 사용하여 평가한 out-of-sample risk에 대한 확률적 보장을 달성한다.

본 논문에서 SAA기반 MPC (SAA-MPC)와 Wasserstein Distributionally Robust MPC (DR-MPC)를 여러 과정을 통하여 다루기 쉬운 프로그램으로 재편성한다.

또한, 환경의 과거 관측으로부터 장애물의 미래 움직임을 예측하기 위해 Distributionally Robust MPC 방법을 Gaussian Process (GP)와 결합한다. 본 연구에서 개발되는 기법들의 성능을 비선형 자동차 모델과 선형화된 쿼드로터 모델을 이용한 시뮬레이션 연구를 통하여 분석한다.

주요어: 분포 강인 최적화, 최적제어, 모션 컨트롤, 충돌회피, 로봇 안전

학번: 2018-26905