

Wasserstein distributionally robust risk-constrained iterative MPC for motion planning: computationally efficient approximations

Alireza Zolanvari

Ashish Cherukuri

Abstract—This paper considers a risk-constrained motion planning problem and aims to find the solution combining the concepts of iterative model predictive control (MPC) and data-driven distributionally robust (DR) risk-constrained optimization. In the iterative MPC, at each iteration, safe states visited and stored in the previous iterations are imposed as terminal constraints. Furthermore, samples collected during the iteration are used in the subsequent iterations to tune the ambiguity set of the DR constraints employed in the MPC. In this method, the MPC problem becomes computationally burdensome when the iteration number goes high. To overcome this challenge, the emphasis of this paper is to reduce the real-time computational effort using two approximations. First one involves clustering of data at the beginning of each iteration and modifying the ambiguity set for the MPC scheme so that safety guarantees still holds. The second approximation considers determining DR-safe regions at the start of iteration and constraining the state in the MPC scheme to such safe sets. We analyze the computational tractability of these approximations and present a simulation example that considers path planning in the presence of randomly moving obstacle.

I. INTRODUCTION

Real-world autonomous mobile robots usually navigate in unknown or partially known environments. Thus, safety is one of the most significant priorities in motion planning problems. There are different optimal control techniques for providing safety guarantees in such situations, among which robust and probabilistic approaches are popular. Alternatively, risk-aware control design has gathered attention recently due to its ability to tune the conservativeness and the safety level of the designed controller between robust and probabilistic approaches. Moreover, the optimization problems leading to risk-aware decisions are often convex and tractable for a large class of risk measures. One challenge persists for all the above-listed methods, that of not knowing the distribution of the uncertainty fully and only having access to a small number of samples of it. In such scenarios, distributionally robust (DR) decisions provide an elegant way of tuning safety and cost-efficiency. Motivated by this, this paper considers a risk-constrained optimal control problem for a motion planning problem and improves on the iterative DR MPC scheme formulated in [1]. The method presented in [1] uses distributional robustness to find safe trajectories, even when the samples of the uncertainty are few in number. However, the method suffers from heavy computational burden when considering the realistic scenario of continuous

distributions. To this end, we provide “low-complexity” approximations of the DR risk constraint considered in the MPC routine without compromising on safety. This process brings the iterative DR MPC method closer to being real-time implementable.

Literature review: Distributional robustness in risk-constrained model predictive control is explored extensively in [2]–[5]. While most of them focus on out-of-sample performance guarantees, the proposed schemes become computationally burdensome as the number of available samples grows. To overcome this challenge, two different strategies have been suggested in the distributionally robust (DR) optimization literature. On the one hand, [6]–[8] reduce the number of samples by data compression and provide performance guarantees for optimizers of the DR problems formulated by the compressed data. The idea of complexity reduction using data compression is known as scenario reduction and was introduced in [9]. On the other hand, constraint tightening approaches, proposed in [10]–[12], tackle the computational challenge by inner-approximating the distributionally robust region and limiting the feasible states to the DR-safe region. None of the above-mentioned studies explore these techniques for iterative approaches. Iterative approaches are well-suited to many real-world problems as they can overcome the challenge of insufficient data points through progressive exploration of the environment and repeated execution of the task. With this motivation, we explore both scenario reduction and constraint-tightening types of approximations of the iterative algorithm from [1]. Our emphasis is on reducing the computational effort and making our method real-time implementable.

Setup and Contributions: We start in Section III with the explanation of the risk-constrained optimal control problem that is tailored for the motion planning problem. Here the risk constraint encodes collision avoidance in presence of a random obstacle. In Section IV, we present the iterative DR MPC method from [1] as a solution strategy and explain the general framework when considering Wasserstein ambiguity sets. We present in Section V the finite-dimensional reformulation of the DR risk constraints when considering Wasserstein ambiguity sets. We discuss the computational challenge of imposing these reformulated constraints, which lead to our main contributions that entail providing two computationally efficient approximations of the DR risk constraints. The first one considers Wasserstein ambiguity sets with a larger radius and the center as the distribution generated using clustered data. The second one generates an inner estimate of the set of points that satisfy DR risk constraints. The inner estimate

The authors are with the Engineering and Technology Institute Groningen, University of Groningen. Email: {a.zolanvari, a.k.cherukuri}@rug.nl. This work was partly supported with a scholarship from the Data Science and Systems Complexity (DSSC) Center, University of Groningen.

has a simple form of being the union of half-spaces. We show that both our approximations are safe in the sense that they satisfy the originally imposed risk constraint up to a pre-specified probability. We identify conditions under which the iterative DR MPC with the defined approximations are recursively feasible and asymptotically convergent. Lastly, we demonstrate the advantage of our method for a risk-averse path planning task.

II. PRELIMINARIES

Here we collect notation and mathematical background.

1) *Notation:* Let \mathbb{R} , $\mathbb{R}_{\geq 0}$, and $\mathbb{N}_{\geq 1}$ denote the set of real, nonnegative real, and natural numbers excluding zero, respectively. Let $\|\cdot\|$ denote the 2-norm. For $N \in \mathbb{N}_{\geq 1}$, we denote $[N] := \{1, \dots, N\}$ and $[N]_0 := \{0, 1, \dots, N\}$. Given $x \in \mathbb{R}$, we let $[x]_+ = \max(x, 0)$.

2) *Conditional Value-at-Risk:* We review notions on conditional value-at-risk (CVaR) from [13]. Given a real-valued random variable Z with probability distribution \mathbb{P} and $\beta \in (0, 1)$, the *value-at-risk* of Z at level β , denoted $\text{VaR}_\beta^\mathbb{P}[Z]$, is the left-side $(1 - \beta)$ -quantile of Z . Formally,

$$\text{VaR}_\beta^\mathbb{P}[Z] = \inf\{\zeta \mid \mathbb{P}(Z \leq \zeta) \geq 1 - \beta\}.$$

The *conditional value-at-risk (CVaR)* of Z at level β , denoted $\text{CVaR}_\beta^\mathbb{P}[Z]$, is given as

$$\text{CVaR}_\beta^\mathbb{P}[Z] = \inf_{t \in \mathbb{R}} \left\{ t + \beta^{-1} \mathbb{E}^\mathbb{P}[Z - t]_+ \right\}, \quad (1)$$

where $\mathbb{E}^\mathbb{P}[\cdot]$ denotes expectation under \mathbb{P} . Under continuity of the CDF of Z , we have $\text{CVaR}_\beta^\mathbb{P}[Z] := \mathbb{E}^\mathbb{P}[Z \geq \text{VaR}_\beta^\mathbb{P}[Z]]$. The parameter β characterizes the risk-averseness. When β is close to unity, the decision-maker is risk-neutral, whereas, β close to the origin implies high risk-averseness.

3) *Wasserstein metric:* Given a compact set $\mathcal{W} \subset \mathbb{R}^{n_w}$, let $\mathcal{P}(\mathcal{W})$ be the set of Borel probability measures supported on \mathcal{W} . Following [14], the 1-Wasserstein metric between measures $\mu, \nu \in \mathcal{P}(\mathcal{W})$ is

$$d_{\mathcal{W}}(\mu, \nu) := \min_{\gamma \in \mathcal{H}(\mu, \nu)} \left\{ \int_{\mathcal{W} \times \mathcal{W}} \|w_1 - w_2\| \gamma(dw_1, dw_2) \right\}, \quad (2)$$

where $\mathcal{H}(\mu, \nu)$ is the set of all distributions on $\mathcal{W} \times \mathcal{W}$ with marginals μ and ν .

III. PROBLEM STATEMENT

Consider the following discrete-time system:

$$x_{t+1} = f(x_t, u_t), \quad (3)$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ represents the dynamics and $x_t \in \mathbb{R}^{n_x}$ and $u_t \in \mathbb{R}^{n_u}$ are the state and control input at time t , respectively. The system state and control input are required to satisfy the following deterministic constraints:

$$x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad \forall t \geq 0, \quad (4)$$

where $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$ are assumed to be compact convex sets. We assume without loss of generality that $0 \in \mathcal{U}$. Our objective in the motion planning problem is to drive the system from an initial state to a target equilibrium point $x_F \in$

\mathcal{X} while ensuring a suitable safety requirement. We encode this task as the following infinite-horizon risk-constrained optimal control problem:

$$\min \sum_{t=0}^{\infty} r(x_t, u_t) \quad (5a)$$

$$\text{s.t. } x_{t+1} = f(x_t, u_t), \quad \forall t \geq 0, \quad (5b)$$

$$x_t \in \mathcal{X}, u_t \in \mathcal{U}, \quad \forall t \geq 0, \quad (5c)$$

$$x_0 = x_S, \quad (5d)$$

$$\text{CVaR}_\beta^\mathbb{P}[g(x_t, w)] \leq 0, \quad \forall t \geq 0. \quad (5e)$$

Here, $x_S \in \mathcal{X}$ is the initial state and the stage-cost $r : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}_{\geq 0}$ is assumed to be continuous satisfying $r(x, u) = 0$ if and only if $(x, u) = (x_F, 0)$. Further, the constraint (5e) represents the safety guarantee, where CVaR stands for the conditional value-at-risk (see Section II-2 for details), w is a random variable with distribution \mathbb{P} supported on the polyhedral convex compact set $\mathcal{W} := \{w \in \mathbb{R}^{n_w} \mid Hw \leq h\}$, the parameter $\beta > 0$ is the risk averseness coefficient, and the continuous function $g : \mathcal{X} \times \mathcal{W} \rightarrow \mathbb{R}$ is referred to as the constraint function. Next, we will make g more precise for the case of avoiding polyhedral obstacles.

Let $C \in \mathbb{R}^{n_p \times n_x}$ be a matrix that gives the components corresponding to the position of a state $x \in \mathbb{R}^{n_x}$ as Cx . Consequently, the feasible region in the position coordinates is denoted by $C\mathcal{X}$. We assume the presence of one uncertain obstacle in $C\mathcal{X}$. At the unperturbed position, the space occupied by the obstacle is defined by $\mathcal{O} := \{p \in \mathbb{R}^{n_p} \mid Ap \leq b\}$, where $A \in \mathbb{R}^{M \times n_p}$, $b \in \mathbb{R}^M$, and \mathcal{O} is assumed to be compact. As the position of the obstacle is uncertain we assume that given a realization $w \in \mathcal{W} \subset \mathbb{R}^{n_p}$ of the random variable, the occupancy of the obstacle is given as

$$\mathcal{O}_w := \mathcal{O} + w = \{p + w \mid Ap \leq b\}.$$

We assume that $\mathcal{O}_w \subset C\mathcal{X}$ for all realizations $w \in \mathcal{W}$. We say that the state x is safe in terms of collision with \mathcal{O}_w if the following constraint is met

$$g(x, w) := d_{\min} - \text{dist}(Cx, \mathcal{O}_w) \leq 0, \quad (6)$$

where $\text{dist}(Cx, \mathcal{O}_w)$ stands for the Euclidean distance of the point Cx from the set \mathcal{O}_w . Specifically,

$$\text{dist}(Cx, \mathcal{O}_w) = \min_{y \in \mathcal{O}_w} \|Cx - y\|.$$

Note that using the structure of the \mathcal{O}_w , the distance can be equivalently written as

$$\text{dist}(Cx, \mathcal{O}_w) := \max_{m \in [M]} [A_m(Cx - w) - b_m]_+,$$

where A_m and b_m are the m -th row of A and m -th component of b , respectively. For the sake of simplicity and without loss of generality, we assume that $\|A_m\| = 1$ for all m . Throughout the paper, we assume g of the above form. In lieu of the above definition, we call x to be \mathbb{P} -risk safe if

$$\text{CVaR}_\beta^\mathbb{P}[g(x, w)] = \text{CVaR}_\beta^\mathbb{P}[d_{\min} - \text{dist}(Cx, \mathcal{O}_w)] \leq 0.$$

We approach solving the infinite-horizon problem (5) using approximations of the iterative MPC scheme proposed

in [1]. The method given in [1] combines ideas from iterative learning MPC [15] and distributionally robust (DR) risk-constrained optimization problems [16]. However, in doing so, the optimization problem at the core of the MPC problem becomes computationally cumbersome, especially when the iteration count goes high, the underlying distribution is continuous, and the ambiguity set is defined using the Wasserstein metric. Specifically, in this case, the optimization problem has mixed-integer decision variables, where the number of constraints and the number of decision variables grow linearly with the number of samples. The objective of this paper is to ease this computational burden by rendering the size of the optimization problem solved in the MPC independent of the number of samples while maintaining safety guarantees.

IV. DR-BASED SAFETY-CONSTRAINED ITERATIVE MPC

Here we present our general iterative framework that is borrowed from [1]. We present this scheme in a form where the safety constraints embedded in the MPC problem are abstract. In the following section, we specify three ways of generating these safety constraints, all of which form different subsets of the set of points that satisfy the risk constraint (5e) in a distributionally robust manner. We then discuss their computational and statistical properties.

A. Building blocks of the iterative scheme

An iteration involves generating a trajectory of the system (3). In particular, the trajectory of the j -th iteration is denoted as

$$\begin{aligned} x^j &:= [x_0^j, x_1^j, \dots, x_t^j, \dots, x_{T_j}^j], \\ u^j &:= [u_0^j, u_1^j, \dots, u_t^j, \dots, u_{T_j-1}^j], \end{aligned} \quad (7)$$

where x_t^j and u_t^j are the system state and the control input at time t , respectively. We assume that $x_0^j = x_S$ for all $j \geq 1$ and that each trajectory j consists of a finite number of time steps $T_j \in \mathbb{N}_{\geq 1}$. We also refer to this quantity as the length of the trajectory.

Recall that the distribution \mathbb{P} of the random variable w is supported on a polyhedral convex compact set $\mathcal{W} := \{w \in \mathbb{R}^{n_w} \mid Hw \leq h\}$. At the start of iteration j , we assume that we have $N_{j-1} \in \mathbb{N}_{\geq 1}$ number of samples of the uncertainty available to us. We denote this dataset as $\widehat{\mathcal{W}}^{j-1} := \{\widehat{w}_1, \dots, \widehat{w}_{N_{j-1}}\}$. We assume that the system gathers $M_j \in \mathbb{N}_{\geq 1}$ number of samples of the uncertainty during the j -th iteration. Thus, $N_j = N_{j-1} + M_j$. If one sample is gathered at each time step of the iteration, then $M_j = T_j$. The data could either be drawn from the distribution \mathbb{P} in an i.i.d. manner or could be obtained from other distributions that are close to \mathbb{P} in some appropriate metric. In both cases, our approach of enforcing risk constraint (5e) for all distributions in an appropriately defined set helps ensure the system's safety, even when the number of available samples is low. To this end, we impose an assumption on the data-gathering process. First, we define the empirical distribution corresponding to the dataset $\widehat{\mathcal{W}}^{j-1}$ as

$$\widehat{\mathbb{P}}_{N_{j-1}} := \frac{1}{N_{j-1}} \sum_{i=1}^{N_{j-1}} \delta_{\widehat{w}_i},$$

where $\delta_{\widehat{w}_i}$ is the dirac-delta distribution placed at the point \widehat{w}_i . Given this distribution and a radius $\theta_{j-1} > 0$, we construct the Wasserstein ambiguity set as

$$\mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1}) := \{\mathbb{Q} \in \mathcal{P}(\mathcal{W}) \mid d_W(\mathbb{Q}, \widehat{\mathbb{P}}_{N_{j-1}}) \leq \theta_{j-1}\}, \quad (8)$$

where d_W is the Wasserstein metric (see Section II-3 for the definition) and $\mathcal{P}(\mathcal{W})$ are all distributions supported on \mathcal{W} . We then assume:

Assumption IV.1. (*Confidence guarantee of \mathbb{P} contained in the ambiguity set*): For any iteration j , we are given a radius $\theta_{j-1} > 0$ such that for any dataset $\widehat{\mathcal{W}}^{j-1}$ gathered by the end of the $(j-1)$ -th iteration, we have

$$\text{Prob}[\mathbb{P} \in \mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})] \geq \zeta, \quad (9)$$

where $\zeta \in (0, 1)$ is a pre-specified confidence level. •

Note that if samples are drawn i.i.d., then $\text{Prob} = \mathbb{P}^{N_{j-1}}$ in the above definition, where $\mathbb{P}^{N_{j-1}}$ represents the N_{j-1} -fold product of the underlying distribution \mathbb{P} . Then, one can derive the relationship between ζ , θ_{j-1} , and N_{j-1} such that (9) holds, see e.g., [14, Theorem 3.4]. Bearing the above definition in mind, a point $x' \in \mathcal{X}$ satisfies the risk constraint (5e) with probability ζ if we ensure that

$$x' \in \mathcal{X}_{\text{DR}}^{j-1} := \{x \in \mathcal{X} \mid \sup_{\mathbb{Q} \in \mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})} \text{CVaR}_{\beta}^{\mathbb{Q}}[g(x, w)] \leq 0\}. \quad (10)$$

The aim of our algorithm is to seek such trajectories, those that satisfy the risk constraint with probability ζ . One way is to impose (10) in our MPC routine. However, the resulting optimization comes with a significant computational burden. Thus, in Section V we define approximations of the set given in (10) with the purpose of balancing computational ease and optimality while ensuring safety throughout.

We now describe other key elements of the iterative learning MPC. Given the trajectory (x^j, u^j) generated in iteration j , the cost-to-go at time t is denoted as:

$$J_{(t:\infty)}^j := \sum_{k=t}^{\infty} r(x_k^j, u_k^j). \quad (11)$$

Thus, the cost of the j -th iteration is $J_{(0:\infty)}^j$. Since we assume that the j -th trajectory has a finite length T_j , for every time step $t \geq T_j$, we assume that the system remains at x_F and the control input is zero. Thus, the infinite sum in (11) is well-defined as $r(x_F, 0) = 0$. In our iterative method, information about the system and the environment grows as iterations progress. The latter is owing to the fact that more data regarding the uncertainty becomes available in each iteration. On the other hand, the former is acquired by means of exploring the state space incrementally. To this end, our method maintains a set of safe states (along with the minimum cost that it takes to go to the target from them) that were explored in the previous iterations and uses them in an iteration as terminal constraints in the MPC scheme (as proposed in [15]). Specifically, the sampled safe set obtained at the end of iteration j and to be used in iteration $j+1$, denoted $\mathcal{S}^j \subseteq [j] \times \mathcal{X} \times \mathbb{R}_{\geq 0}$, is defined recursively as

$$\mathcal{S}^j = \mathbb{S}^j \left(\mathcal{S}^{j-1} \cup \{(j, x_t^j, J_{(t:\infty)}^j)\}_{t=1}^{T_j} \right), \quad (12)$$

where the set $\{(j, x_t^j, J_{(t:\infty)}^j)\}_{t=1}^{T_j}$ collects the set of states visited in iteration j , along with the associated cost-to-go. The counter j is maintained in this set to identify the iteration to which a state with a particular cost-to-go is associated with. The set \mathcal{S}^{j-1} is the set used in iteration j . The map \mathbb{S}^j only keeps the states that are safe with respect to the new dataset $\widehat{\mathcal{W}}^j$. The exact map \mathbb{S}^j is explained in our algorithm.

For ease of exposition, we define maps $\Pi_{\text{traj}}(\cdot)$, $\Pi_{\text{state}}(\cdot)$, and $\Pi_{\text{cost}}(\cdot)$, such that, given a safe set \mathcal{S} , $\Pi_{\text{traj}}(\mathcal{S})$, $\Pi_{\text{state}}(\mathcal{S})$, and $\Pi_{\text{cost}}(\mathcal{S})$ return the set of all trajectory indices, states, and cost-to-go values that appear in \mathcal{S} , respectively. The following assumption is required to initialize our iterative procedure with a nonempty sampled safe set.

Assumption IV.2. (*Initialization with robust trajectory*): The first iteration starts with the sampled safe set \mathcal{S}^0 containing a finite-length robustly safe trajectory x^0 that starts from x_S and reaches x_F . This means that the trajectory x^0 in \mathcal{S}^0 robustly satisfies all constraints of problem (5), that is, $x \in \mathcal{X}$, $g(x, w) \leq 0$ for all $w \in \mathcal{W}$, and all $x \in \Pi_{\text{state}}(\mathcal{S}^0)$. •

Note that for every state stored in the sampled safe set \mathcal{S}^j , we store the cost-to-go from it. However, it is possible that a state appears in multiple trajectories and given that \mathcal{S}^j contains several trajectories, it is beneficial to maintain a minimum cost-to-go from every state in it. That being the case, we define the map

$$Q^j(x) := \begin{cases} \min_{J \in F^j(x)} J, & x \in \Pi_{\text{state}}(\mathcal{S}^j), \\ +\infty, & x \notin \Pi_{\text{state}}(\mathcal{S}^j), \end{cases} \quad (13)$$

where

$$F^j(x) = \{J_{(t:\infty)}^i \mid \Pi_{\text{state}}(\{(i, x_t^i, J_{(t:\infty)}^i)\}) = \{x\}, \\ (i, x_t^i, J_{(t:\infty)}^i) \in \mathcal{S}^j\}. \quad (14)$$

Here, the set $F^j(x)$ contains all cost-to-go values associated with the state $x \in \Pi_{\text{state}}(\mathcal{S}^j)$ and consequently, the function Q^j determines the minimum among these.

Given the above-described elements, we now present the optimization problem that lies at the core of our method. For generality, we write the problem for generic current state x , sampled safe set $\bar{\mathcal{S}}$, and safety constraint $x \in \bar{\mathcal{X}} \subset \mathcal{X}$. Let $K \in \mathbb{N}_{\geq 1}$ be the length of the horizon and consider

$$\mathcal{J}_{(\bar{\mathcal{S}}, \bar{\mathcal{X}})}(x) := \begin{cases} \min & \sum_{k=0}^{K-1} r(x_k, u_k) + \bar{Q}(x_K) \\ \text{s. t.} & x_{k+1} = f(x_k, u_k), \forall k \in [K-1]_0, \\ & x_k \in \bar{\mathcal{X}}, u_k \in \mathcal{U}, \forall k \in [K-1]_0, \\ & x_0 = x, \\ & x_K \in \Pi_{\text{state}}(\bar{\mathcal{S}}), \end{cases} \quad (15)$$

where $\bar{Q} : \mathcal{X} \rightarrow \mathbb{R}$ gives the minimum cost-to-go for all states in $\bar{\mathcal{S}}$ and is calculated in a similar manner as in (13). The decision variables in the above problem are (x_0, x_1, \dots, x_K) and $(u_0, u_1, \dots, u_{K-1})$. The set $\bar{\mathcal{S}}$ defines the terminal constraint $x_K \in \Pi_{\text{state}}(\bar{\mathcal{S}})$. Finally, the constraint $x_k \in \bar{\mathcal{X}}$ encodes the safety guarantee. For iteration j and time step t , the MPC scheme solves the finite-horizon

problem (15) with $x = x_t^j$, $\bar{\mathcal{S}} = \mathcal{S}^{j-1}$, and $\bar{Q} = Q^{j-1}$, while $\bar{\mathcal{X}}$ takes one of the following values: $\mathcal{X}_{\text{Wass}}^{j-1}$, $\mathcal{X}_{\text{cl-Wass}}^{j-1}$, or $\mathcal{X}_{\text{inn-Wass}}^{j-1}$. The set $\mathcal{X}_{\text{Wass}}^{j-1}$ is defined by the reformulation of the Wasserstein DR risk-constraint, the set $\mathcal{X}_{\text{inn-Wass}}^{j-1}$ is an inner approximation of $\mathcal{X}_{\text{Wass}}^{j-1}$, and finally, $\mathcal{X}_{\text{cl-Wass}}^{j-1}$ is defined in a similar way as $\mathcal{X}_{\text{Wass}}^{j-1}$ but with clustered data. We explain these further in Section V.

B. Algorithm describing the iterative scheme

Here we put together the building blocks of our method that were outlined above. The resulting scheme is given in Algorithm 1 and is similar to the algorithm from our previous work [1]. Each iteration $j \geq 1$ of Algorithm 1 starts with a sampled safe set \mathcal{S}^{j-1} and a safe set $\mathcal{X}_{\text{safe}}^{j-1}$, where the latter is determined using the dataset $\widehat{\mathcal{W}}^{j-1}$ and the radius θ_{j-1} . We represent this association via the map \mathbb{X} (Line 3). As explained above, the set $\mathcal{X}_{\text{safe}}^{j-1}$ takes value as $\mathcal{X}_{\text{Wass}}^{j-1}$, $\mathcal{X}_{\text{cl-Wass}}^{j-1}$, or $\mathcal{X}_{\text{inn-Wass}}^{j-1}$. The precise definition of these sets are given in Section V. Given \mathcal{S}^{j-1} and $\mathcal{X}_{\text{safe}}^{j-1}$, the first step of the iteration (Line 2) involves generating a trajectory (x^j, u^j) using `Safe_MPC` routine (described in Algorithm 2). The dataset is updated to $\widehat{\mathcal{W}}^j$ and the set $\mathcal{X}_{\text{safe}}^j$ is computed for the next iteration using $\widehat{\mathcal{W}}^j$ and the radius θ_j in Line 3. The trajectory x^j along with its associated cost-to-go is appended to the sampled safe set in Line 4. The set \mathcal{UI}^j collects in Line 5 all previous trajectories for which one of the states is not safe with respect to the newly determined set $\mathcal{X}_{\text{safe}}^j$. In Line 6, the set of trajectories in $\mathcal{I}^{j-1} \cup \{j\}$ that are not in \mathcal{UI}^j are collected in the set \mathcal{I}^j . Consequently, the states visited in trajectories in \mathcal{I}^j are stored in \mathcal{S}^j in Line 7 and their minimum cost-to-go is computed in Line 8. Collectively, Lines 5 to 7 represent the map \mathbb{S} defined in (12).

Algorithm 1: Iterative MPC with DR-based safety constraints

Input : \mathcal{S}^0 – Initial sampled safe set
 $\widehat{\mathcal{W}}^0$ – Initial set of samples
 \mathcal{I}^0 – Index of trajectory in \mathcal{S}^0

Initialize : $j \leftarrow 1$, $\mathcal{X}_{\text{safe}}^0$, $\mathcal{UI}^0 \leftarrow \emptyset$

- 1 **while** $j > 0$ **do**
- 2 Set $(x^j, u^j) \leftarrow \text{Safe_MPC}(\mathcal{S}^{j-1}, \mathcal{X}_{\text{safe}}^{j-1})$;
 $T^j \leftarrow \text{length}(x^j)$; $\widehat{\mathcal{W}}^j \leftarrow \widehat{\mathcal{W}}^{j-1} \cup \{\widehat{w}_i\}_{i=1}^{M_j}$
- 3 Set $\mathcal{X}_{\text{safe}}^j \leftarrow \mathbb{X}(\widehat{\mathcal{W}}^j, \theta_j)$
- 4 Set $\mathcal{S}^{j-1} \leftarrow \mathcal{S}^{j-1} \cup \{(j, x_t^j, J_{(t:\infty)}^j)\}_{t=1}^{T_j}$
- 5 Set $\mathcal{UI}^j \leftarrow \{i \in (\mathcal{I}^{j-1} \cup \{j\}) \mid (i, x, J) \in \mathcal{S}^{j-1} \text{ and } x \notin \mathcal{X}_{\text{safe}}^j\}$
- 6 Set $\mathcal{I}^j \leftarrow (\mathcal{I}^{j-1} \cup \{j\}) \setminus \mathcal{UI}^j$
- 7 Set $\mathcal{S}^j \leftarrow \{(i, x, J) \in \mathcal{S}^{j-1} \mid i \in \mathcal{I}^j\}$
- 8 Compute $Q^j(x)$ for all $x \in \Pi_{\text{state}}(\mathcal{S}^j)$ using (13)
- 9 Set $j \leftarrow j + 1$

Algorithm 1 calls the `Safe_MPC` routine in each iteration. This procedure is given in Algorithm 2 where at each time step t , the finite-horizon problem (15) is solved with $x = x_t$. The optimal solution is denoted as

$$\begin{aligned} x_{\text{vec},t}^* &= [x_{t|t}^*, \dots, x_{t+K|t}^*], \\ u_{\text{vec},t}^* &= [u_{t|t}^*, \dots, u_{t+K-1|t}^*], \end{aligned} \quad (16)$$

where $x_{t+k|t}$ is the prediction made at time t regarding the state at time $t+k$. The control at time t is set as the first element $u_{t|t}^*$ (Line 5) and it is appended to the trajectory \mathbf{u} . The state is updated and added to \mathbf{x} in Line 6. The procedure moves to the next time step with the updated state as x_{t+1} .

Algorithm 2: Safe MPC function

```

1 Function Safe_MPC( $\bar{\mathcal{S}}, \bar{\mathcal{X}}$ ):
   Initialize :  $t \leftarrow 0$ ;  $x_0 \leftarrow x_S$ ;  $\mathbf{x} \leftarrow [x_0]$ ,  $\mathbf{u} \leftarrow []$ 
2   Set  $\bar{Q}$  as minimum cost-to-go in  $\bar{\mathcal{S}}$  (use (13))
3   while  $x_t \neq x_F$  do
4     Solve (15) with  $x = x_t$  and obtain optimal
       solutions  $x_{\text{vec},t}^*$  and  $u_{\text{vec},t}^*$ 
5     Set  $u_t \leftarrow u_{t|t}^*$ ;  $\mathbf{u} \leftarrow [u, u_t]$ 
6     Set  $x_{t+1} \leftarrow f(x_t, u_t)$ ;  $\mathbf{x} \leftarrow [x, x_{t+1}]$ ;  $t \leftarrow t + 1$ 
7   return ( $\mathbf{x}, \mathbf{u}$ )
8 end

```

Remark IV.1. (Comparison with [1]): We note that the general structure of our algorithm is similar to that in [1]. However, there are two key differences that highlight the contribution of this paper. First, the algorithm in [1] is written in a general form for any type of ambiguity set generated using data. There is no emphasis on computational tractability and the simulations only focus on discrete distributions. On the other hand, we here specify the Wasserstein ambiguity set and focus more on the computational issues. Second, our algorithm and the approximations presented in Section V are tailored for the motion planning problem; they exploit the structure of the obstacle avoidance constraint to derive fast MPC routines that can be implemented in real-time. •

V. APPROXIMATIONS OF DR-BASED SAFETY CONSTRAINT

In this section, we explain the three safety sets that are used as constraints in the MPC routine of Algorithm 1. All these sets lead to trajectories that satisfy the risk constraint (5e) with probability ζ . Before proceeding further we first provide a finite-dimensional representation of an upper bound of the worst-case CVaR over a Wasserstein ambiguity set. This will depict the computational issues of solving the finite horizon problem (15) with two DR-based safety sets $\mathcal{X}_{\text{Wass}}^{j-1}$ and $\mathcal{X}_{\text{cl-Wass}}^{j-1}$. Consequently, it will also motivate the design of the third safety set $\mathcal{X}_{\text{inn-Wass}}^{j-1}$.

Lemma V.1. (Reformulation of the worst-case risk): Consider an atomic distribution

$$\mathbb{P}^{\text{cen}} := \sum_{\ell=1}^L p_\ell \delta_{w_\ell^{\text{cen}}},$$

where $w_\ell^{\text{cen}} \in \mathcal{W}$ for all $\ell \in [L]$ and $p_\ell \in (0, 1)$ for all $\ell \in [L]$ with $\sum_{\ell=1}^L p_\ell = 1$. Consider the ambiguity set

$$\mathcal{B}(\mathbb{P}^{\text{cen}}, \theta_{\text{cen}}) := \{\mathbb{Q} \in \mathcal{P}(\mathcal{W}) \mid d_W(\mathbb{Q}, \mathbb{P}^{\text{cen}}) \leq \theta_{\text{cen}}\},$$

where d_W is the Wasserstein metric (2) and $\mathcal{P}(\mathcal{W})$ are all distributions supported on \mathcal{W} . Then, for the function g given in (6), we have

$$\begin{aligned} & \sup_{\mathbb{Q} \in \mathcal{B}(\mathbb{P}^{\text{cen}}, \theta_{\text{cen}})} \text{CVaR}_\beta^{\mathbb{Q}}[g(x, w)] \quad (17) \\ & \leq \begin{cases} \inf & \lambda \theta_{\text{cen}} - \beta \eta + \sum_{\ell=1}^L s_\ell \\ \text{s. t.} & p_\ell^{-1} s_\ell - \eta \geq d_{\min} - [(ACx - b)^\top \nu \\ & \quad - [(A^\top \nu - H^\top \gamma_\ell)^\top w_\ell^{\text{cen}} + \gamma_\ell^\top h]], \forall \ell \in [L], \\ & \|A^\top \nu - H^\top \gamma_\ell\| \leq \lambda, \|A^\top \nu\| \leq 1, \quad \forall \ell \in [L], \\ & \nu, \gamma_\ell \in \mathbb{R}_{\geq 0}^M, \lambda, s_\ell \in \mathbb{R}_{\geq 0}, \eta \in \mathbb{R}, \forall \ell \in [L]. \end{cases} \end{aligned}$$

The proof follows from a similar result in [5] and hence is omitted. Note that the above result is written for a general ambiguity set defined with the center \mathbb{P}^{cen} and radius θ_{cen} . Such a choice is motivated by the fact that in defining the safety set $\mathcal{X}_{\text{cl-Wass}}^{j-1}$, we will use the distribution derived from clustered data instead of the empirical one.

A. Safety set $\mathcal{X}_{\text{Wass}}^{j-1}$

For obtaining the safety set $\mathcal{X}_{\text{Wass}}^{j-1}$, we replace \mathbb{P}^{cen} and θ_{cen} in Lemma V.1 with the empirical distribution $\hat{\mathbb{P}}_{N_{j-1}}$ and θ_{j-1} , respectively. Then, we define $\mathcal{X}_{\text{Wass}}^{j-1}$ as the set of x for which the upper bound on the worst-case risk given in (17) is nonnegative. That is, in this case

$$\begin{aligned} \mathbb{X}(\hat{\mathcal{W}}^{j-1}, \theta_{j-1}) &= \mathcal{X}_{\text{Wass}}^{j-1} \quad (18) \\ &:= \left\{ x \in \mathcal{X} \mid \exists \nu, \gamma_\ell \in \mathbb{R}_{\geq 0}^M, \lambda, s_\ell \in \mathbb{R}_{\geq 0}, \eta \in \mathbb{R}, \right. \\ & \quad \text{such that} \\ & \quad \lambda \theta_{j-1} - \beta \eta + \sum_{\ell=1}^{N_{j-1}} s_\ell \leq 0, \\ & \quad p_\ell^{-1} s_\ell - \eta \geq d_{\min} - [(ACx - b)^\top \nu \\ & \quad \quad - [(A^\top \nu - H^\top \gamma_\ell)^\top \hat{w}_\ell + \gamma_\ell^\top \hat{h}], \forall \ell \in [N_{j-1}], \\ & \quad \left. \|A^\top \nu - H^\top \gamma_\ell\| \leq \lambda, \|A^\top \nu\| \leq 1, \forall \ell \in [N_{j-1}] \right\}. \end{aligned}$$

The above definition and the inequality (17) imply that if $x \in \mathcal{X}_{\text{Wass}}^{j-1}$, then $\sup_{\mathbb{Q} \in \mathcal{B}(\hat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})} \text{CVaR}_\beta^{\mathbb{Q}}[g(x, w)] \leq 0$. Hence, when $\mathcal{X}_{\text{safe}}^{j-1} = \mathcal{X}_{\text{Wass}}^{j-1}$ in Line 3 of Algorithm 1, then the generated trajectory in the j -th iteration is \mathbb{P} -risk safe with probability ζ .

Notice that in one of the constraints on the right-hand side of (18), the decision variables x and ν appear in a bilinear term. Hence, enforcing the state to be in $\mathcal{X}_{\text{Wass}}^{j-1}$ renders the optimization problem nonconvex with the number of constraints and the decision variables scaling with the size of the dataset. This poses a computational challenge when solving the finite-horizon problem (15) with $\bar{\mathcal{X}}$ set as $\mathcal{X}_{\text{Wass}}^{j-1}$ given in (18), especially since the terminal constraint in (15) is equivalent to a mixed-integer one. To alleviate this roadblock, we propose the following two alternatives.

B. Safety set $\mathcal{X}_{\text{cl-Wass}}^{j-1}$

As explained above, using each sample in defining the ambiguity set increases the computational burden in the reformulated problem. Hence, we use clustering following the ideas outlined in [6] and [7]. In particular, we assume that the designer selects the number of clusters $N^{\text{clu}} \in \mathbb{N}_{\geq 1}$ based on the trade-off between accuracy and computational cost. Given N^{clu} and samples $\{\widehat{w}_1, \dots, \widehat{w}_{N_{j-1}}\}$, we determine the centers of clusters $\{w_\ell^{\text{clu}, j-1}\}_{\ell=1}^{N^{\text{clu}}} \subset \mathcal{W}$ and an association map $\text{id} : \{1, \dots, N_{j-1}\} \rightarrow \{1, \dots, N^{\text{clu}}\}$ such that each sample \widehat{w}_i is associated with some cluster given by $\text{id}(i)$. Then, we form the clustered empirical distribution as

$$\mathbb{P}^{\text{clu}, j-1} := \sum_{\ell=1}^{N^{\text{clu}}} p_\ell \delta_{w_\ell^{\text{clu}, j-1}},$$

where for each $\ell \in \{1, \dots, N^{\text{clu}}\}$, we have

$$p_\ell = \frac{|\{i \in \{1, \dots, N_{j-1}\} \mid \text{id}(i) = \ell\}|}{N_{j-1}}.$$

That is, the fraction of points associated with the cluster ℓ . Further, we select the radius of the ambiguity set as $\theta_{\text{clu}, j-1} = \theta_{j-1} + \bar{d}$, where $\bar{d} = \max_{i \in \{1, \dots, N_{j-1}\}} \|\widehat{w}_i - w_{\text{id}(i)}^{\text{clu}, j-1}\|$. We then use $\mathbb{P}^{\text{clu}, j-1}$ and $\theta_{\text{clu}, j-1}$ in place of \mathbb{P}^{cen} and θ_{cen} , respectively, in Lemma V.1 and define the safety set

$$\begin{aligned} \mathbb{X}(\widehat{\mathcal{W}}^{j-1}, \theta_{j-1}) &= \mathcal{X}_{\text{cl-Wass}}^{j-1} \\ &:= \left\{ x \in \mathcal{X} \mid \exists \nu, \gamma_\ell \in \mathbb{R}_{\geq 0}^M, \lambda, s_\ell \in \mathbb{R}_{\geq 0}, \eta \in \mathbb{R}, \right. \\ &\quad \text{such that} \\ &\quad \lambda \theta_{\text{clu}, j-1} - \beta \eta + \sum_{\ell=1}^{N^{\text{clu}}} s_\ell \leq 0, \\ &\quad p_\ell^{-1} s_\ell - \eta \geq d_{\min} - ([ACx - b]^\top \nu \\ &\quad - [(A^\top \nu - H^\top \gamma_\ell)^\top w_\ell^{\text{clu}, j-1} + \gamma_\ell^\top h]), \forall \ell \in [N^{\text{clu}}], \\ &\quad \left. \|A^\top \nu - H^\top \gamma_\ell\| \leq \lambda, \|A^\top \nu\| \leq 1, \forall \ell \in [N^{\text{clu}}] \right\}. \end{aligned} \quad (19)$$

Note that the advantage of the above safety set lies in the fact that the set is defined with fewer constraints and this number only depends on the number of clusters N^{clu} instead of N_{j-1} . While we have reduced the number of points in the support of the center of the ambiguity set, we have increased the radius. In the process we have retained the guarantee:

Lemma V.2. (Confidence guarantee of \mathbb{P} contained in clustered data ambiguity set): Consider any iteration j and suppose Assumption IV.1 holds. Then, we have

$$\text{Prob}[\mathbb{P} \in \mathcal{B}(\mathbb{P}^{\text{clu}, j-1}, \theta_{\text{clu}, j-1})] \geq \zeta.$$

The proof follows directly from that of [6, Lemma VIII.2]. Owing to the above result, by ensuring $\mathcal{X}_{\text{safe}}^{j-1} = \mathcal{X}_{\text{cl-Wass}}^{j-1}$ for iteration j , we obtain a trajectory that satisfies the risk constraint (5e) with probability ζ . Finally, we comment that there are several ways of obtaining the clustered distribution \mathbb{P}^{clu} from the dataset $\widehat{\mathcal{W}}^{j-1}$. In our numerical example, we use K -means clustering. Another interesting approach is to use the I-Cover algorithm from [6]. However, in this case, the number of clusters grows as the number of samples increases.

C. Safety set $\mathcal{X}_{\text{inn-Wass}}^{j-1}$

Here we take a different approach as compared to the earlier methods. Instead of imposing distributionally robust constraints in the finite-horizon problem of the MPC, we create an inner estimate of the set $\mathcal{X}_{\text{DR}}^{j-1}$ (see (10)) at the beginning of the j -th iteration. We then force our trajectory to lie in this inner-estimated set during the MPC implementation. To achieve this, we present the following result that serves as a tool for constructing the inner estimation of $\mathcal{X}_{\text{DR}}^{j-1}$.

Proposition V.1. (Inner-estimating $\mathcal{X}_{\text{DR}}^{j-1}$): Consider the j -th iteration and the ambiguity set $\mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})$ given in (8). Define for each $m \in [M]$ the set

$$\begin{aligned} \mathcal{X}_m^{\text{inn}, j-1} &:= \{x \in \mathcal{X} \mid \sup_{\mathbb{Q} \in \mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})} \text{CVaR}_\beta^\mathbb{Q}[b_m + A_m^\top w \\ &\quad + d_{\min} - A_m^\top Cx] \leq 0\}, \end{aligned}$$

where we recall that A_m and b_m are the m -th row of A and m -th component of b , respectively. Let $\mathcal{X}^{\text{inn}, j-1} := \bigcup_{m=1}^M \mathcal{X}_m^{\text{inn}, j-1}$. If $x \in \mathcal{X}^{\text{inn}, j-1}$, then

$$\sup_{\mathbb{Q} \in \mathcal{B}(\widehat{\mathbb{P}}_{N_{j-1}}, \theta_{j-1})} \text{CVaR}_\beta^\mathbb{Q}[g(x, w)] \leq 0,$$

where g is given in (6).

Proof. Note that for any $y \in \mathcal{O}_w$ we have $Ay \leq b + Aw$. We will establish the result by showing that for any $x \in \mathcal{X}$ and $w \in \mathcal{W}$, if

$$b_m + A_m^\top w + d_{\min} - A_m^\top Cx \leq 0, \quad (20)$$

for some $m \in \{1, \dots, M\}$, then $g(x, w) \leq 0$. To this end, note that

$$\begin{aligned} &b_m + A_m^\top w + d_{\min} - A_m^\top Cx \\ &= b_m + A_m^\top w + \max_{\|f\| \leq d_{\min}} f^\top A_m - A_m^\top Cx \\ &\stackrel{(a)}{\geq} \max_{y \in \mathcal{O}_w} A_m^\top y + \max_{\|f\| \leq d_{\min}} f^\top A_m - A_m^\top Cx \\ &= \max_{y \in \mathcal{O}_w + d_{\min} \mathcal{B}} A_m^\top y - A_m^\top Cx, \end{aligned}$$

where \mathcal{B} is the unit ball centered at the origin, the first equality is due to $\|A_m\| = 1$, and (a) follows from the fact that $A_m^\top y \leq b_m + A_m^\top w$ for any $y \in \mathcal{O}$. From the above inequality, if (20) is satisfied, then $\max_{y \in \mathcal{O}_w + d_{\min} \mathcal{B}} A_m^\top y - A_m^\top Cx \leq 0$. This implies $Cx \notin \text{int}(\mathcal{O}_w + d_{\min} \mathcal{B})$, where $\text{int}(\cdot)$ represents the interior. That is, $\max_{y \in \mathcal{O}_w + d_{\min} \mathcal{B}} \|Cx - y\| \geq 0$ and so, $\text{dist}(Cx, \mathcal{O}_w) \geq d_{\min}$. This completes the proof. \square

Roughly speaking, in the above result we have constructed a safe region $\mathcal{X}^{\text{inn}, j-1}$ as the union of sets $\mathcal{X}_m^{\text{inn}, j-1}$ that are themselves intersection of half-spaces and the set \mathcal{X} . The outward normal defining the half-space is chosen to be the same that defines the occupancy of the obstacle. For example, in two dimensions, if the obstacle is a square with normal directions aligning with the axes, then the region $\mathcal{X}^{\text{inn}, j-1}$ will turn out to be the whole space $C\mathcal{X}$ except for the set of points that belong to a rectangular region. This fact will become more clear in our simulation section.

In light of Proposition V.1, given the dataset $\widehat{\mathcal{W}}^{j-1}$ and radius θ_{j-1} , we set

$$\mathbb{X}(\widehat{\mathcal{W}}^{j-1}, \theta_{j-1}) = \mathcal{X}_{\text{inn-Wass}}^{j-1} = \mathcal{X}^{\text{inn},j-1} \quad (21)$$

in Algorithm 1. Then, the trajectory generated in the j -th iteration is \mathbb{P} -risk safe with probability ζ . We next comment on the procedure of determining the constraints defining the set $\mathcal{X}_{\text{inn-Wass}}^{j-1}$. We first compute

$$b_m^{\text{inn},j-1} = \sup_{\mathbb{Q} \in \mathcal{B}(\mathbb{P}_{N_{j-1}}, \theta_{j-1})} \text{CVaR}_{\beta}^{\mathbb{Q}}[A_m^{\top} w]$$

for all $m \in \{1, \dots, M\}$. Consequently, we set

$$\mathcal{X}_{\text{inn-Wass}}^{j-1} = \cup_{m=1}^M \{x \in \mathcal{X} \mid b_m + b_m^{\text{inn},j-1} + d_{\min} - A_m^{\top} C x \leq 0\}.$$

While the above set has a simple form, we still require computing the scalar values $\{b_m^{\text{inn},j-1}\}$. To this end, we provide the following finite-dimensional optimization problem:

$$b_m^{\text{inn},j-1} = \begin{cases} \inf & \lambda \theta_{j-1} - \beta \eta + \sum_{\ell=1}^L s_{\ell} \\ \text{s. t.} & p_{\ell}^{-1} s_{\ell} - \eta \geq (A_m^{\top} - H_m^{\top} \xi_{\ell})^{\top} \widehat{w}_{\ell} + \xi_{\ell} h_m, \forall \ell \in [N_{j-1}], \\ & \|A_m^{\top} - H_m^{\top} \xi_{\ell}\| \leq \lambda, \quad \forall \ell \in [N_{j-1}], \\ & \lambda, s_{\ell}, \xi_{\ell} \in \mathbb{R}_{\geq 0}, \eta \in \mathbb{R}, \quad \forall \ell \in [N_{j-1}]. \end{cases}$$

We next conclude this section with the guarantees that our methods enjoy. The first result states the safety and recursive feasibility and then we present the asymptotic convergence. The proofs are omitted as they are similar to those given in [1] for analogous statements.

Proposition V.2. (Safety and recursive feasibility of Algorithm 1): *Let Assumption IV.1 and IV.2 hold. Then, for each of the safety sets represented by maps (18), (19), and (21), at each iteration $j \geq 1$ and time step $t \geq 0$, the finite-horizon problem (15) with $x = x_t^j$, $\overline{\mathcal{S}} = \mathcal{S}^{j-1}$, and $\overline{\mathcal{X}} = \mathbb{X}(\widehat{\mathcal{W}}^{j-1}, \theta_{j-1})$ solved in Algorithm 1 is feasible. Further, each point in the generated trajectory (x^j, u^j) satisfies the risk-constraint (5e) with probability ζ .*

Proposition V.3. (Asymptotic convergence of Algorithm 1): *Let Assumption IV.2 hold. Then, for each of the safety sets represented by maps (18), (19), and (21) and each iteration $j \geq 1$ of Algorithm 1, the trajectory (x^j, u^j) generated by Safe_MPC satisfies $x_t^j \rightarrow x_F$ as $t \rightarrow \infty$.*

VI. SIMULATION

In this section, we consider a motion planning problem in the presence of a randomly moving obstacle to compare the performance and efficiency of the presented approaches. In this problem, a circular mobile robot is navigating in a 2-D environment. We assume that at each time step, the position of the obstacle is observable and as the iterations progress, more information is revealed.

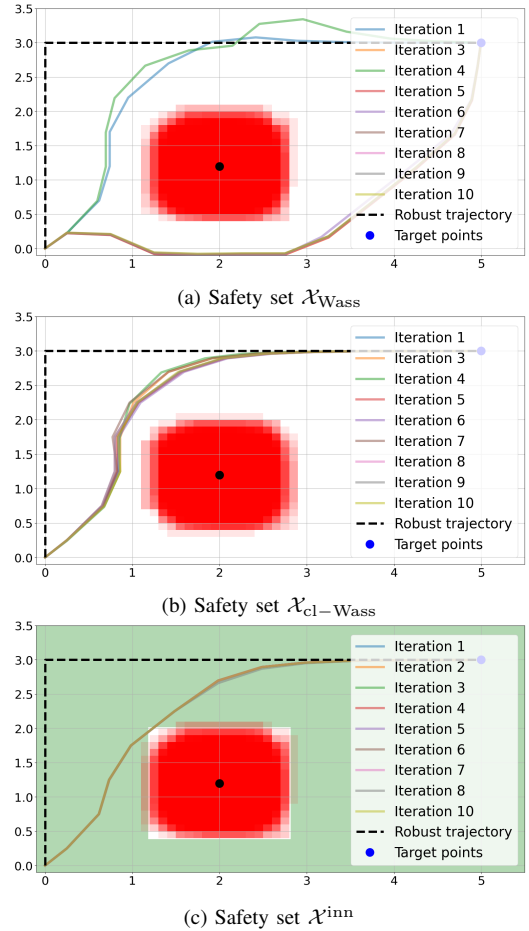


Fig. 1: Plots depicting the application of Algorithm 1 with the presented approximations for a path planning task in presence of an uncertain obstacle (see Section VI for details). The dashed black line represents the initial robust trajectory and the green area in Figure 1c shows \mathcal{X}^{inn} in the last iteration. The obstacle's occupancy is depicted by the shaded red heat map.

1) *Setup:* The deterministic linear dynamics of the mobile robot is represented as:

$$x_{t+1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} u_t,$$

where the state vector x_t consists of the position and velocity and the input vector u_t represents the acceleration. At each iteration, the agent starts at $x_s = [0, 0, 0, 0]$ and aims to reach the target at $x_F = [5, 3, 0, 0]$, while a square obstacle is moving randomly with the position of its center being $o_t = [2, 1.2] + w_t$. The distribution of $w_t \in \mathbb{R}^2$ is the product distribution, where for each axis, the component distribution is a truncated zero-mean normal distribution with support $[-0.45, 0.45]$ and variance $\sigma = 0.15$. Before starting the first iteration, the observations set is initialized with 15 i.i.d samples. The stage cost $r(x, u)$ is in quadratic form and given as $r(x_t, u_t) = (x_F - x_t)^{\top} Q (x_F - x_t) + u_t^{\top} R u_t$, where $Q = \text{diag}(1, 1, 0.01, 0.01)$ and $R = \text{diag}(0.01, 0.01)$. Other parameters are specified in Table I. The optimization problem (15) is solved using GEKKO [17] on a PC with an Intel Core i7-10610U 2.30-GHz processor and 16-GB

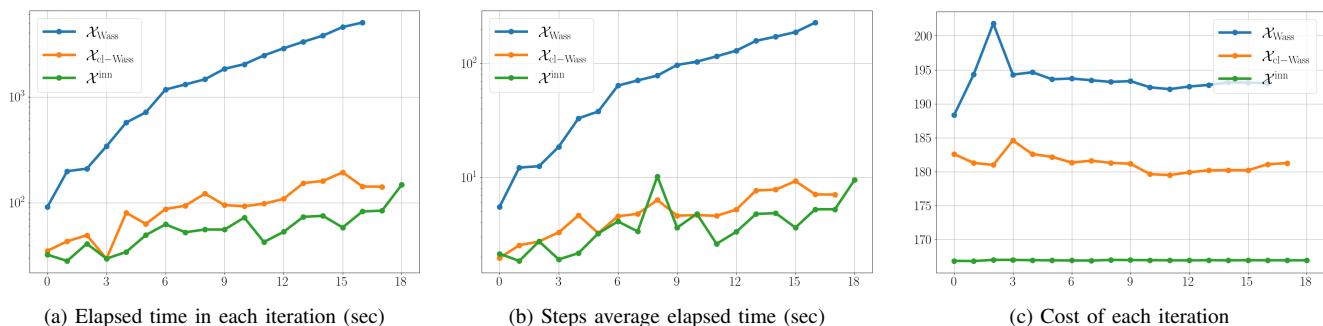


Fig. 2: The comparison of three presented approximation approaches in Algorithm 1 in terms of the computational time and cost efficiency. Plots illustrate the elapsed time in each iteration, elapsed time per step in each iteration, and the cost of each iteration over the progress of the algorithm.

Risk coefficient (β)	0.05	Distance threshold (d_{\min})	0.1
Obstacle length	1	Agent radius	0.2
Horizon length (K)	11	Ambiguity set radius (θ)	$1e^{-3}$
Number of iterations	20	Number of clusters (N^{clu})	5

TABLE I: Simulation parameters

RAM. For $\mathcal{X}_{\text{Wass}}^{j-1}$ and $\mathcal{X}_{\text{cl-Wass}}^{j-1}$, the problem is solved using an IPOPT-based solver and for $\mathcal{X}^{\text{inn},j-1}$ an APOPT-based solver is used. Each algorithm is executed for 20 iterations.

Results: Trajectories generated using three approximation approaches of the problem (15) are depicted in Figure 1. Resulted trajectories show that considering $\mathcal{X}_{\text{Wass}}$ as the safety set, provide the algorithm with more freedom to explore the environment. After expanding the safe set in the first iterations, the agent tries to find an efficient trajectory from the lower side of the obstacle. On the other hand, considering safety sets $\mathcal{X}_{\text{cl-Wass}}$ and \mathcal{X}^{inn} does not let the agent deviate much from the initial safe set. In Figure 1c, the green area that represents \mathcal{X}^{inn} includes some areas in which the probability of the obstacle’s presence is low. This shows the difference between the presented distributionally robust risk-averse set \mathcal{X}^{inn} and a normal robustly safe set. It is notable that the conservatism of the algorithms can be tuned via d_{\min} and β which is not the focus of this experiment.

In Figure 2, we compare the efficiency of the presented approaches in terms of both performance and computational effort. The first two plots show that using safety sets $\mathcal{X}_{\text{cl-Wass}}$ and \mathcal{X}^{inn} in problem (15) reduces the computational time notably and their difference with the $\mathcal{X}_{\text{Wass}}$ increase over iterations. In addition, the average elapsed time per step in each iteration indicates that the clustering approach and the inner approximation of the feasible set are comparable in terms of real-time decision-making, however, the inner approximation approach is slightly faster. Furthermore, bilinearity of constraints in $\mathcal{X}_{\text{Wass}}$ and $\mathcal{X}_{\text{cl-Wass}}$ limits the solver to be able to guarantee optimality. As a result, involving \mathcal{X}^{inn} leads to better trajectories in terms of cost efficiency.

VII. CONCLUSIONS

We have considered a risk-constrained optimal control problem for motion planning and explored using iterative DR MPC method as a solution strategy. Considering Wasserstein ambiguity sets in the DR MPC routine, we have formulated two approximations of the optimization problem driving the MPC. We have shown the approximations to be computationally efficient and resulting in safe trajectories. We

have illustrated the strength of our methods via a numerical example. Future work includes further bringing down the computational costs by considering reachable sets and exploring distributed implementation for multi-robot setup.

REFERENCES

- [1] A. Zolanvari and A. Cherukuri, “Data-driven distributionally robust iterative risk-constrained model predictive control,” in *European Control Conference*, (London, United Kingdom), pp. 1578–1583, 2022.
- [2] P. Coppens and P. Patrinos, “Data-driven distributionally robust MPC for constrained stochastic systems,” *IEEE Control Systems Letters*, vol. 6, pp. 1274–1279, 2022.
- [3] C. Mark and S. Liu, “Data-driven distributionally robust MPC: An indirect feedback approach,” 2021. arXiv preprint available at <https://arxiv.org/abs/2109.09558>.
- [4] A. Hakobyan and I. Yang, “Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 939–957, 2021.
- [5] A. Navsalkar and A. R. Hota, “Data-driven risk-sensitive model predictive control for safe navigation in multi-robot systems,” *arXiv preprint arXiv:2209.07793*, 2022.
- [6] D. Li and S. Martínez, “Data assimilation and online optimization with performance guarantees,” *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2115–2129, 2020.
- [7] F. Fabiani and P. J. Goulart, “The optimal transport paradigm enables data compression in data-driven robust control,” in *American Control Conference*, (New Orleans, LA), pp. 2412–2417, 2021.
- [8] I. Wang, C. Becker, B. V. Parys, and B. Stellato, “Mean robust optimization,” *arXiv preprint arXiv:2207.10820*, 2022.
- [9] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, “Scenario reduction in stochastic programming,” *Mathematical programming*, vol. 95, pp. 493–511, 2003.
- [10] C. Mark and S. Liu, “Stochastic MPC with distributionally robust chance constraints,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 7136–7141, 2020.
- [11] A. Dixit, M. Ahmadi, and J. W. Burdick, “Distributionally robust model predictive control with total variation distance,” *arXiv preprint arXiv:2203.12062*, 2022.
- [12] M. Fochesato and J. Lygeros, “Data-driven distributionally robust bounds for stochastic model predictive control,” in *IEEE Conf. on Decision and Control*, (Cancun, Mexico), pp. 3611–3616, 2022.
- [13] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming*. Philadelphia, PA: SIAM, 2014.
- [14] P. Mohajerin Esfahani and D. Kuhn, “Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations,” *Mathematical Programming*, vol. 171, no. 1-2, pp. 115–166, 2018.
- [15] U. Rosolia and F. Borrelli, “Learning model predictive control for iterative tasks. A data-driven control framework,” *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, 2017.
- [16] A. Cherukuri and A. R. Hota, “Consistency of distributionally robust risk-and chance-constrained optimization under Wasserstein ambiguity sets,” *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1729–1734, 2020.
- [17] L. Beal, D. Hill, R. Martin, and J. Hedengren, “GEKKO optimization suite,” *Processes*, vol. 6, no. 8, p. 106, 2018.