2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611276

Distributionally Robust CVaR-Based Safety Filtering for Motion Planning in Uncertain Environments

Sleiman Safaoui and Tyler H. Summers

***Abstract*— Safety is a core challenge of autonomous robot motion planning, especially in the presence of dynamic and uncertain obstacles. Many recent results use learning and deep learning-based motion planners and prediction modules to predict multiple possible obstacle trajectories and generate obstacle-aware ego robot plans. However, planners that ignore the inherent uncertainties in such predictions incur collision risks and lack formal safety guarantees. In this paper, we present a computationally efficient safety filtering solution to reduce the collision risk of ego robot motion plans using multiple samples of obstacle trajectory predictions. The proposed approach reformulates the collision avoidance problem by computing safe halfspaces based on obstacle sample trajectories using distributionally robust optimization (DRO) techniques. The safe halfspaces are used in a model predictive control (MPC)-like safety filter to apply corrections to the reference ego trajectory thereby promoting safer planning. The efficacy and computational efficiency of our approach are demonstrated through numerical simulations.**

I. Introduction

Autonomous robots have many application areas including autonomous driving [1], warehouse management and logistics [2] drone delivery [3], and agriculture [4]. A core challenge facing autonomous robots is navigation in dynamic and uncertain environments, i.e. in the presence of moving obstacles whose future motion cannot be predicted exactly. This complicates the robot safety requirements: the ego robot must presume the obstacles' intentions and predict their future trajectories for use in computing its own motion plan. Thus, safety hinges on how accurately the dynamic obstacles' behavior can be predicted [5], [6]. Failing to account for prediction uncertainties may incur undue risk of severe collisions [7].

Various methods have been studied for predicting how obstacles will behave, but it is still an active area of research. In [8], a hidden Markov model is used for better understanding urban scenarios for autonomous vehicles (AVs). In another work, [9] uses a support vector machine and Bayesian filtering to predict lane

This work was supported in part by the United States Air Force Office of Scientific Research under Grants FA2386-19-1-4073 and FA9550-23-1-0424, and in part by the National Science Foundation under Grant ECCS-2047040.

S. Safaoui and T. H. Summers are with the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX, USA. E-mail: {sleiman.safaoui, tyler.summers}utdallas.edu.

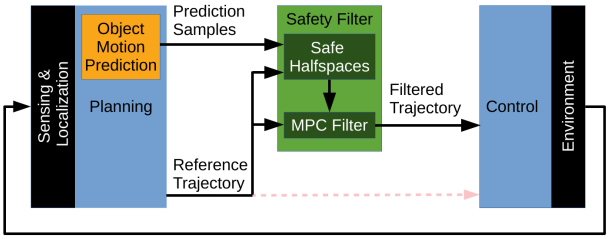


Fig. 1: An autonomy stack with the proposed *Safety Filter* module. The module intercepts the reference trajectory and corrects it to enforce the safety requirement.

change intentions for AVs. Furthermore, deep learning approaches have also been used. End-to-end motion planners, such as [10], implicitly account for future predictions, but they fail to explicitly capture the environment uncertainties which may lead to collisions. FIERY [11] generates a birds-eye-view probabilistic future predictions map which estimates environmental uncertainties, but it still requires a formal approach to use this data to enforce safety. In [12], a neural network ensemble is employed to estimate prediction uncertainty and identify rare cases. However, ensembles are resource-intensive to train and deploy.

Optimization-based methods are commonly used to formally guarantee safety requirements. Using samples of obstacle motion, [13] creates an empirical distribution then formulates a *distributionally robust optimization* (DRO) problem to ensure safety and avoid collisions under any distribution that is "close" to the empirical one. However, the solution solves a non-convex problem which is computationally demanding and not suitable for real-time operation. In [14], a similar DRO problem is formulated with application to multi-robot systems. The collision avoidance constraints are reformulated using a result from [15] to produce a convex nonlinear MPC problem with DRO constraints. Another recent solution uses conformal prediction to guarantee safety when using learning-based planners [16]. Here, prediction regions that satisfy a given probability bound are found and used in an MPC optimization problem.

A related approach to enforcing safety uses a *safety filter*. Instead of adding constraints to one of the modules of the autonomy stack, a standalone module takes the

reference trajectory from the motion planner and outputs a corrected *filtered* trajectory, as illustrated in Fig. 1. The filtered motion plan is guaranteed to satisfy certain safety requirements. Safety filters have been used in *deterministic* settings for autonomous racing [17], multi-agent motion planning [18], and autonomous driving [19] to filter unsafe learning-based motion plans.

Motion planners with prediction uncertainty often suffer from two issues. 1) They consider average behavior or limit the chance of unsafe events. These are suitable for real-time deployment but fail under edge cases (in the prediction distribution tail). 2) They rigorously tackle uncertainties and edge cases, but they are computationally intensive. In this work, we address this gap by proposing a computationally efficient solution using axiomatic risk theory to handle uncertainties and deal with edge cases. We assume that the ego robot has a planned reference trajectory and samples of the obstacles' future trajectories (e.g. through [20]). Our solution starts by finding safe halfspaces based on a distributionally robust conditional value-at-risk (DR-CVaR) risk metric for each obstacle. Then, the DR-CVaR safe halfspaces are used in an MPC-based safety filter to enforce safety. Compared to [14], our method is for *safety filtering* existing motion plans. It can solve all DR-CVaR problems in parallel before solving the MPC problem, simplifying the latter. The main contributions of this work are:

- We extend the notion of a safe halfspace and define data-driven DR-CVaR safe halfspaces that bound the risk of violating a safety specification.
- We verify the efficiency of the DR-CVaR halfspaces through a numerical analysis and show that they can be computed in a few milliseconds with up to a few hundred samples.
- We formulate an MPC-based safety filter that uses the DR-CVaR safe halfspaces to constrain the motion planning problem and bound collision risks.
- We demonstrate the efficacy of our solution and its ability to handle edge-cases through numerical simulations in a variety of motion planning scenarios.

NOTATION

The d-dimensional zero vector (matrix) and identity matrix are denoted by $\mathbb{0}_d$ ($\mathbb{0}_{d,d}$) and I_d, respectively. We use $a : b$ to denote all integers between $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ (inclusive). The Minkowski sum is denoted by $\oplus$. The transpose of a vector or matrix is denoted by $(\cdot)^\top$. The inner product of vectors z_1 and z_2 is denoted by $z_1 \cdot z_2 = z_1^\top z_2$. The support function of a compact set $\mathcal{C}$ is given by $S_\mathcal{C}(z) := \sup_{x\in\mathcal{C}} z \cdot x$. Random variables/vectors are denoted in $\boldsymbol{bold}$ and $\mathbb{E}[\cdot]$ is the expected value operator. Given a loss $\boldsymbol{l}$, the CVaR metric is $\mathrm{CVaR}_\alpha^\mathbb{P}(\boldsymbol{l}) := \inf_{\tau\in\mathbb{R}} \mathbb{E}^\mathbb{P}[\tau + \frac{1}{\alpha}\max\{\boldsymbol{l}-\tau, 0\}]$ which is evaluated with respect to the α worst-cases of the distribution $\mathbb{P}$ (the $1-\alpha$ quantile in the upper tail). For N_s samples of the loss $\{l^1, \ldots, l^{N_s}\}$, we use a sample average approximation to evaluate the expected value in the CVaR metric:
$\mathrm{CVaR}_\alpha^\mathbb{P}(\boldsymbol{l}) \approx \inf_{\tau\in\mathbb{R}} \frac{1}{N_s}\sum_{i=1}^{N_s}(\tau + \frac{1}{\alpha}\max\{l^i - \tau, 0\})$.

II. DR-CVAR SAFE MOTION PLANNING WITH UNCERTAIN DYNAMIC OBSTACLES

A. Motion Planning Safety Filtering Problem

We consider the problem of motion planning for an ego robot in the presence of N_{ob} dynamic obstacles whose future behavior is uncertain. The ego robot dynamics are assumed linear and described by:

$$x(t+1) = Ax(t) + Bu(t) \tag{1a}$$
$$y(t) = Cx(t) \tag{1b}$$

where $x(t) \in \mathbb{R}^n$ is the robot state at time t, $u(t) \in \mathbb{R}^m$ is the control, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are the state and input matrices, $y(t) \in \mathbb{R}^d$ is the position, and $C \in \mathbb{R}^{d\times n}$ is the matrix that extracts the position from the state. The ego robot is modeled by a convex and compact set $\mathcal{A}$. We also assume that the ego robot has a desired reference trajectory over a horizon of length T given by $\mathcal{T}^r(t) := \{x^r(t), \ldots, x^r(t+T)\}$. This reference trajectory may be obtained from any planning module (e.g. neural network, sampling-based or optimization-based methods, etc.).

Each obstacle is modeled as a convex and compact set $\mathcal{O}_i$ $\forall i \in [1 : N_{ob}]$. Their dynamics are unknown and the motion is uncertain. Furthermore, the prediction distribution is inherently unknown. Instead, we assume access to a module that can generate *sample predictions* of the obstacles' trajectories. The s-th sample trajectory of the i-th obstacle for a horizon of T time steps is given by $\mathcal{T}_i^s(t) := \{p_i^s(t), \ldots, p_i^s(t+T)\}$ and includes only position information ($p_i^s \in \mathbb{R}^d$).

An autonomy stack may pass the reference trajectory directly to a tracking controller (Fig. 1 faded red arrow) while assuming that it accounts for the obstacles' future trajectories. However, safety requirements desired in a reference trajectory may not be guaranteed, especially if the planner is based on a neural network. A formal safety guarantee is defined as follows.

Definition 1 (Safety Guarantee)**.** *Given a reference trajectory $\mathcal{T}^r$, a chosen risk metric $\mathcal{R}$, and a risk bound δ, safety is guaranteed iff $\mathcal{R}(\mathcal{T}^r) \leq \delta$.*

To enforce this notion of safety, we advocate for the usage of a safety filter that takes a reference trajectory and outputs a filtered trajectory as depicted in Fig. 1. This safety filtering problem is formalized below.

Problem 1 (Motion Planning Safety Filtering)**.** *Given an ego reference trajectory $\mathcal{T}^r(t)$ and obstacle trajectory samples $\mathcal{T}_i^s(t)$, $s \in [1 : N_s]$ for every obstacle*

$i \in [1 : N_{ob}]$, *find a filtered ego trajectory that ensures the safety of the ego robot per Definition 1.*

We now proceed with specifying the adopted risk metric $\mathcal{R}$ for the motion planning problem.

B. Signed Collision Loss Function

Consider an ego robot and a single dynamic obstacle. Denote by y the ego robot position, by y^r the ego robot reference position, and by p the obstacle position.

The deterministic collision avoidance condition is given by $(y \oplus \mathcal{A}) \cap (p \oplus \mathcal{O}) = \emptyset$. Using computational geometry arguments and convexifying the constraint via separating halfspaces, similar to [18, III-A.2], we have:

$$\begin{aligned}&(y \oplus \mathcal{A}) \cap (p \oplus \mathcal{O}) = \emptyset \iff (y - p) \notin \mathcal{O} \oplus (-\mathcal{A}) \\ &\impliedby z \cdot (y - p) \geq S_{\mathcal{O}}(z) + S_{-\mathcal{A}}(z) \end{aligned} \tag{2}$$

where z is any chosen unit vector per [21].

The constraint (2) is sufficient for ensuring collision avoidance for a deterministic system. Additionally, we can define a deterministic *safe halfspace*

$$\mathcal{H}(h, g) := \{y \mid h \cdot y + g - (S_{\mathcal{O}}(h) + S_{-\mathcal{A}}(h)) \leq 0\}$$

such that if $y \in \mathcal{H}$, then y is collision free and satisfies (2). For the deterministic case, the parameters $h,\ g$ of $\mathcal{H}$ can be directly mapped to the values in (2). Furthermore, using $\mathcal{H}$ we can define a signed distance function that quantifies the violation or satisfaction amount of a point relative to the collision avoidance constraint. In particular, consider the following loss function:

$$\ell(p, h, g) = -(h \cdot p + \underbrace{g - (S_{\mathcal{O}}(h) + S_{-\mathcal{A}}(h))}_{\tilde{g} :=}). \tag{3}$$

Given $h,\ g,\ p$, if $\ell(p, h, g) \geq 0$ then the obstacle p intrudes into the safe halfspace $\mathcal{H}$ and $\ell(p, h, g)$ is the intrusion amount. Otherwise, the obstacle is $|\ell(p, h, g)|$ units away from the boundary of the safe halfspace.

C. Collision Avoidance using DR-CVaR Safe Halfspaces

When the obstacle position is a random variable $\boldsymbol{p}$, (2) is ill-posed and the loss (3) becomes a random variable $\ell(\boldsymbol{p}, h, g)$. We now define a risk-based safe halfspace with respect to a risk metric $\mathcal{R}$ applied to $\ell(\boldsymbol{p}, h, g)$.

Definition 2 (Risk-Based Safe Halfspace)**.** *Given a halfspace normal h and a risk metric $\mathcal{R}$, a risk-based halfspace is given by $\mathcal{H}^{\mathcal{R}}(\tilde{g}) := \{p \mid h \cdot p + \tilde{g} \leq 0\}$, where $\tilde{g} = g^* - (S_{\mathcal{O}}(h) + S_{-\mathcal{A}}(h))$, and g^* is the optimal value of the following optimization problem:*

$$\min_{g} \quad g \tag{4a}$$

$$\textit{subject to} \quad \mathcal{R}(\ell(\boldsymbol{p}, g)) \leq \delta. \tag{4b}$$

It is common to approximate the distribution of $\boldsymbol{p}$ then pose (4b) as a chance constraint on the probability of collision hence bounding the value-at-risk (VaR) (e.g. [16]). However, VaR is not a coherent risk metric in the sense of Artzner et al. [22, Def 2.4], while CVaR is and it has been advocated for in robotics [23], [24]. Intuitively, CVaR measures the expected cost in the tail of the distribution. Thus, it not only accounts for the *frequency* of undesirable events, but also their *severity*.

Since we only have samples of $\boldsymbol{p}$ generated by the prediction module, and its underlying distribution is unknown, we use a data-driven *distributionally robust* CVaR risk metric, i.e. DR-CVaR, in (4b). In particular, the samples of $\boldsymbol{p}$ define an empirical distribution $\hat{\mathbb{P}}$. But, instead of treating $\hat{\mathbb{P}}$ as the true distribution, which can give large sampling errors for small N_s, we consider a Wasserstein distance-based ambiguity set $\mathcal{P}$ around $\hat{\mathbb{P}}$. Formally, $\mathcal{P} = \mathbb{B}_\epsilon(\hat{\mathbb{P}}) := \{\mathbb{Q} \in \mathcal{M}(\Xi) \mid \mathrm{d_w}(\hat{\mathbb{P}}, \mathbb{Q}) \leq \epsilon\}$ is the Wasserstein ball containing all distributions with a Wasserstein distance of at most ϵ from $\hat{\mathbb{P}}$. Here, $\mathcal{M}(\Xi)$ is the set of all finite mean distributions supported on Ξ and $\mathrm{d_w}(\cdot, \cdot)$ is the Wasserstein distance. Consider $\mathbb{Q}_1, \mathbb{Q}_2 \in \mathcal{M}(\Xi)$ and a norm $\|\cdot\|$ (we use the 2-norm), the Wassertein distance is defined by $\mathrm{d_w}(\mathbb{Q}_1, \mathbb{Q}_2) := \int_{\Xi^2} \|\xi_1 - \xi_2\| \, \Pi(d\xi_1, d\xi_2)$ where Π is a joint distribution of ξ_1 and ξ with marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$ respectively [25, Definition 3.1]. Intuitively, the Wasserstein distance represents the minimum transportation cost of transporting mass from one distribution into another. Accordingly, a DR-CVaR safe halfspace is defined as follows.

Definition 3 (DR-CVaR Safe Halfspace)**.** *Consider an empirical distribution $\hat{\mathbb{P}}$ supported on samples of a predicted obstacle position, a Wasserstein-based ambiguity set $\mathcal{P} = \mathbb{B}_\epsilon(\hat{\mathbb{P}})$ and a halfspace normal h. The* DR-CVaR *safe halfspace is defined as $\mathcal{H}^{dr}(\tilde{g}) := \{p \mid h \cdot p + \tilde{g} \leq 0\}$ where $\tilde{g} = g^* - (S_{\mathcal{O}}(h) + S_{-\mathcal{A}}(h))$, and g^* is the optimal value of the following problem:*

$$\min_{g} \quad g \tag{5a}$$

$$\textit{subject to} \quad \text{DR-CVaR}_\alpha^\epsilon(\ell(\boldsymbol{p}, g)) \leq \delta \tag{5b}$$

with $\text{DR-CVaR}_\alpha^\epsilon(\ell(\boldsymbol{p}, g)) := \sup_{\mathbb{P} \in \mathcal{P}} \text{CVaR}_\alpha^{\mathbb{P}}(\ell(\boldsymbol{p}, g))$.

Here, (5b) is an infinite dimensional constraint. However, since ℓ is affine, we can utilize tools from [25] to obtain a finite-dimensional convex reformulation.

Proposition 1. *The problem in (5) with support $\Xi := \{p \mid Vp \leq v\}$ for the random variable $\boldsymbol{p}$ and dual norm $\|\cdot\|_*$ admits the finite-dimensional reformulation:*

$$\begin{aligned} \inf_{g, \tau, \lambda, \eta_i, \gamma_{ik}} \quad & g \\ \textit{subject to} \quad & \lambda\epsilon + \frac{1}{N_s} \sum_{i=1}^{N_s} \eta_i \leq \delta, \\ & a_k \cdot p^i + b_k \tilde{g} + c_k \tau + \gamma_{ik} \cdot (v - V^\top p^i) \leq \eta_i \\ & \left\| V^\top \gamma_{ik} - a_k \right\|_* \leq \lambda, \quad \gamma_{ik} \geq 0. \end{aligned} \tag{6}$$

Proof. Using the CVaR definition with (3) we have:

$$\mathrm{CVaR}_\alpha^{\mathbb{P}}(\ell(\boldsymbol{p}, g)) \tag{7a}$$

$$= \inf_\tau \mathbb{E}^{\mathbb{P}}\left[\max\left(-\frac{h\cdot\boldsymbol{p}+\tilde{g}}{\alpha} + (1-\frac{1}{\alpha})\tau, \tau\right)\right] \tag{7b}$$

$$= \inf_\tau \mathbb{E}^{\mathbb{P}}\left[\underbrace{\max_k (a_k\boldsymbol{p} + b_k\tilde{g} + c_k\tau)}_{\mathbb{M}:=}\right] \tag{7c}$$

where $k \in \{1, 2\}$ and $a_1 = \frac{-h}{\alpha}$, $b_1 = \frac{-1}{\alpha}$, $c_1 = 1 - \frac{1}{\alpha}$, $a_2 = \mathbb{0}_d$, $b_2 = 0$, $c_2 = 1$. Then, the DR-CVaR constraint (5b) becomes

$$\text{DR-CVaR}_\alpha^\epsilon(\ell(\boldsymbol{p}, g)) \leq \delta \iff \sup_{\mathbb{P}\in\mathcal{P}} \mathrm{CVaR}_\alpha^{\mathbb{P}}(\ell(\boldsymbol{p}, g)) \leq \delta$$

$$\iff \sup_{\mathbb{P}\in\mathcal{P}} \inf_\tau \mathbb{E}^{\mathbb{P}}[\mathbb{M}] \leq \delta \Leftarrow \inf_\tau \underbrace{\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}^{\mathbb{P}}[\mathbb{M}]}_{\text{WCE}:=} \leq \delta \tag{8}$$

where the last line follows by the minimax inequality $\sup\inf(\cdot) \leq \inf\sup(\cdot)$. The worst-case expectation (WCE) term matches that from [25, (10)] with piecewise affine loss functions in the random variable $\boldsymbol{p}$ and hence applying [25, Corollary 5.1-(i)] results in (6). □

Example 1. *Consider an ego reference position* $y^r = [-0.9, -0.8]^\top$ *and a nominal obstacle at* $p = [0.5, 0]^\top$, *in 2D space, both of radius* $r = 0.3$. *Fig. 2 illustrates safe halfspaces using the expected value (mean),* CVaR *and* DR-CVaR *risk metrics with* $h = (p - y^r)/\|p - y^r\|$ *as depicted by the arrow. The halfspaces use 100 samples of the obstacle position sampled from the Gaussian random vector* $\mathcal{N}(p, diag(0.01, 0.01))$ *where* $diag(\cdot)$ *is the diagonal matrix. We use* $\alpha = 0.2$, $\delta = 0.1$, *and* $\epsilon \in \{0.05,\ 0.1,\ 0.2\}$. *As* ϵ *increases, the Wasserstein ball becomes larger, including more distributions, and hence the* DR-CVaR *halfspaces become more conservative. When* $\epsilon \to 0$, *the* DR-CVaR *safe halfspace converges to the* CVaR *case. Here,* y^r *is safe with respect to all halfspaces except the* DR-CVaR *with* $\epsilon = 0.2$. *Note that increasing* δ *relaxes the safety constraint and would make the halfspaces less conservative.*

D. DR-CVaR Safe Halfspace Guarantees

If the position of the ego robot is constrained to the DR-CVaR safe halfspaces, we obtain a safety guarantee that bounds collision risk, however, we require a technical assumption on the lightness of the tails[1] for the underlying true probability distribution $\mathbb{P}_{true}$ of predicted obstacle positions. Thus, we have the following lemma.

Lemma 1 (Concentration Inequality [26, Theorem 2]). *For a light-tailed distribution* $\mathbb{P}_{true}$ *with* $\chi := \mathbb{E}^{\mathbb{P}_{true}}[exp(\|\boldsymbol{p}\|^\rho)] < \infty$ *for* $\rho > 1$ *[25, Assumption 3.3] or* $\rho > 0$ *per [26], we have*

$$\mathbb{P}(\mathrm{d_w}(\mathbb{P}_{true}, \hat{\mathbb{P}}) \geq \epsilon) \leq \beta \tag{9}$$

where β *is a constant term that depends on* χ, ρ, N_s, *and* ϵ. *Alternatively,* $\mathbb{P}(\mathbb{P}_{true} \in \mathbb{B}_\epsilon(\hat{\mathbb{P}})) \geq 1 - \beta$ *for a carefully chosen value of* N_s.

[1]This assumption hold trivially if Ξ is compact. Since the distribution in our case represents the position of the obstacle, then Ξ is compact as obstacles can always be limited to a finite detection range.

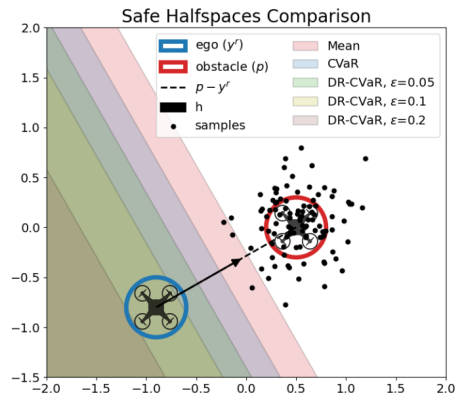


Fig. 2: Comparison between safe halfspaces based on the mean, CVaR, and DR-CVaR with different ϵ values.

Therefore, for a desired concentration bound, a minimum N_s can be computed to guarantee that satisfying the DR-CVaR constraint (5b) ensures the satisfaction of the CVaR variant of it for the true distribution with probability $1 - \beta$. However, this is usually conservative and generally requires N_s to be large making (6) computationally expensive. Instead, we advocate for treating ϵ, N_s, and δ as tuneable parameters to achieve a desired safety level that can be validated experimentally.

III. MPC-Based Safety Filter with DR-CVaR Safe Halfspaces

We now return to the safe motion planning problem described in Problem 1. Our solution has two steps:

1) Computing the safe halfspaces: We interpret the safety constraint $\mathcal{R}(\mathcal{T}^r) \leq \delta$ from Definition 1 point-wise in time and per obstacle. Thus, the obstacle trajectory samples are used to solve (5) per obstacle per time step over the horizon T resulting in $\mathcal{H}_i^{dr}(t+t')\ \forall i \in [1 : N_{ob}],\ t' \in [1 : T]$ (we drop the dependence of $\mathcal{H}$ on g for simplicity).
2) MPC Filter: The computed halfspaces are used as constraints in an MPC optimization problem that computes a minimally deviating trajectory from the reference trajectory. The MPC optimization problem is formalized below.

Problem 2 (MPC-Based Safety Filter). *Given an ego reference trajectory* $\mathcal{T}^r(t_0)$ *at time* t_0 *and linear ego vehicle dynamics (1), find the filtered trajectory* $\mathcal{T}(t_0) = \{x(t_0), \ldots, x(t_0 + T)\}$ *for the ego vehicle that satisfies the* DR-CVaR *safe halfspaces* $\mathcal{H}_i^{dr}(t),\ t \in [t_0+1 : t_0+$

TABLE I: CVaR and DR-CVaR safe halfspace average Solve and Call times (ms).

Risk Metric	$N_s = 50$		$N_s = 500$		$N_s = 1500$	
	Solve	Call	Solve	Call	Solve	Call
CVaR	0.144	2.23	1.91	4.32	5.98	8.95
DR-CVaR	0.236	9.10	2.58	73.4	8.53	216

T], by solving the finite-horizon optimization problem (11) with the objective function (10).

$$J(x(t_0 : t_0+T), u(t_0 : t_0+T-1)) = \sum_{t=t_0}^{T+t_0-1} u(t)^\top R(t)u(t) + \sum_{t=t_0+1}^{T+t_0} (x(t)-x^r(t))^\top Q(t)((x(t)-x^r(t)) \tag{10}$$

$$\min_{x,u} \quad J(x(t_0 : t_0+T), u(t_0 : t_0+T-1)) \tag{11a}$$

$$\textit{subject to} \quad x(t+1) = Ax(t) + Bu(t),\ u(t) \in \mathcal{U}(t) \quad \forall t \in [t_0 : t_0+T-1] \tag{11b}$$

$$y(t) = Cx(t),\ x(t_0) = x^r(t_0) \tag{11c}$$

$$y(t) \in \mathcal{Y}(t) \cap \left(\bigcap_{i=1}^{N_{ob}} \mathcal{H}_i^{dr}(t) \right) \tag{11d}$$

with $\forall t \in [t_0+1 : t_0+T]$ when not stated. Here, $\mathcal{Y}(t) \subseteq \mathbb{R}^d$ is a convex position constraint for the ego robot (e.g. environment bounds), $\mathcal{U}(t) \subseteq \mathbb{R}^m$ is the convex control input constraint set, and $R \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{m\times m}$ are symmetric, positive semidefinite cost matrices.

The MPC safety filter 11 is a quadratic program (QP) and can be modeled with tools such as CVXPY [27] and solved with many solvers, such as ECOS [28].

Remark 1. *We assume that (11) is always feasible at $t = 0$. The conjunction in (11d) may become empty or unreachable due to $\mathcal{U}(t)$ rendering the problem infeasible. In this work, we use the most recent optimal control $u^*(t)$ from solving (11) and proceed with the next available control, $u^*(t+1)$, until the problem is solved again or we run out ($u^*(t+T-1)$). Future works will address alternative infeasibility handling approaches.*

IV. Numerical Validation

For simplicity and clarity, we analyze the proposed approach in 2D space ($d = 2$). All experiments use CVXPY with ECOS and were executed on a Dell Precision 7520 computer with an Intel Xeon E3-1535M v6 CPU and 32GB RAM. Experiment code can be found at: https://github.com/TSummersLab/dr-cvar-safety_filtering.

A. Computation Cost Analysis

We perform a numerical analysis for the DR-CVaR safe halfspace computation time. We find the safe halfspaces as done in Example 1 for various number of samples N_s. For each N_s, (5) is solved 500 times with different random samples. We report three times: 1) the CVXPY reported setup time (Setup Time), 2) the CVXPY reported solver solve time (Solve Time), and 3) the time for executing the CVXPY solve method (Call Time). The results, in milliseconds, are reported in Fig. 3. Since (5) is a linear program (LP), it can be solved efficiently within a few milliseconds even for a few hundred samples. The Call Time includes a large overhead not captured by the Setup Time. Additionally, we compare the mean times for finding the DR-CVaR and CVaR safe halfspaces in Table I. This reveals that while the Solve Time is only about 50% higher for the DR-CVaR safe halfsapces, the Call Time grows quickly compared to CVaR counterpart. We conclude that the DR-CVaR safe halfspace formulation is suitable for real-time operation especially if a solver is used directly.

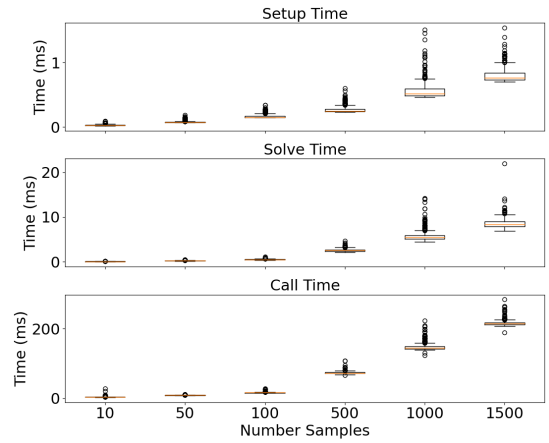
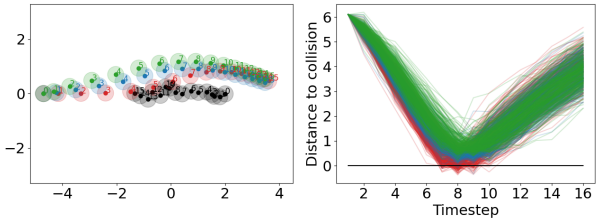


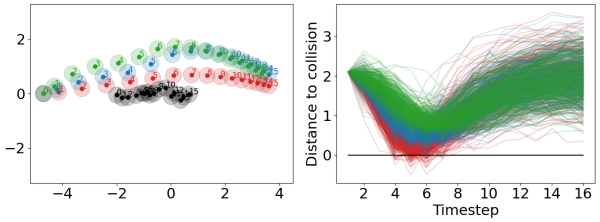
Fig. 3: DR-CVaR safe halfspace computation time

B. Motion Planning Safety Filtering Simulations

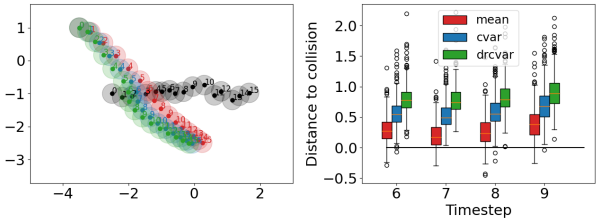
1) Simulation Setup: We model the ego and obstacle geometries as circles of radii $r_{\mathcal{A}} = r_{\mathcal{O}} = 0.3$. The ego robot uses double integrator dynamics with $A = \begin{bmatrix} I_2 & I_2T_s \\ \mathbb{0}_{2,2} & I_2 \end{bmatrix}$, $B = \begin{bmatrix} \frac{1}{2}I_2T_s^2 \\ I_2T_s \end{bmatrix}$, and $C = \begin{bmatrix} I_2 & \mathbb{0}_{2,2} \end{bmatrix}$ where $T_s = 0.2$sec is the discrete time step. The reference trajectory is generated using an obstacle-agnostic MPC-based motion planner with a target goal state. Its details are not discussed since our safety filter is agnostic to the chosen motion planning algorithm. Unbeknown to the ego robot, the obstacles are single integrators with $A = I_2,\ B = I_2T_s,\ C = I_2$. Sample trajectories are generated by adding a Gaussian random noise $\mathcal{N}(\mathbb{0}_2, diag(0.01, 0.01))$ to a nominal trajectory that keeps the vehicle aligned with the x-axis at a desired speed. However, when realizing the true position of the obstacle, a Laplace distribution with the same mean and covariance as the Gaussian is sampled. We use $T = 10$, $\alpha = 0.2$, $\delta = 0.1$, and $\epsilon = 0.05$.



(a) Head-on Collision



(b) Overtaking



(c) Intersection

Fig. 4: Motion Planning Scenarios and their distance to collision statistics across 300 Monte Carlo simulation. Red: Mean safe halfspace $\mathcal{H}^{\mathbb{E}}$. Blue: CVaR safe halfspace $\mathcal{H}^{cvar}$. Green: DR-CVaR safe halfspace $\mathcal{H}^{dr}$ Black: Obstacle.

2) Filtering in Different Scenarios: We use three types of reach-avoid motion planning scenarios: 1) Head-on collision, 2) Overtaking, and 3) Intersection. The left column of Fig. 4 overlays the trajectories of one experiment using safe halfspaces based on the mean value, CVaR, and DR-CVaR risk metrics. In all three cases, trajectories using DR-CVaR safe halfspaces achieve the lowest risk, while those using the expected value-based safe halfspace have the highest risk.

To demonstrate the advantage of using DR-CVaR halfspaces, we perform a Monte Carlo simulation repeating each experiment 300 times. The distance to collision $\|y-p\|-r_{\mathcal{A}}-r_{\mathcal{O}}$ is plotted in the right column of Fig. 4 with the last row showing a box plot zoomed-in view. Clearly, the performance using safe halfspaces based on the mean value result in more frequent collisions in all three cases. In the box plot, the worst-case scenario using the mean value safe halfspace is around -0.45. With this collision amount, both the ego robot and obstacle would be significantly damaged. The worst-case collision using CVaR results in a -0.22 distance to collision. Here, it may lead to a less severe collision.

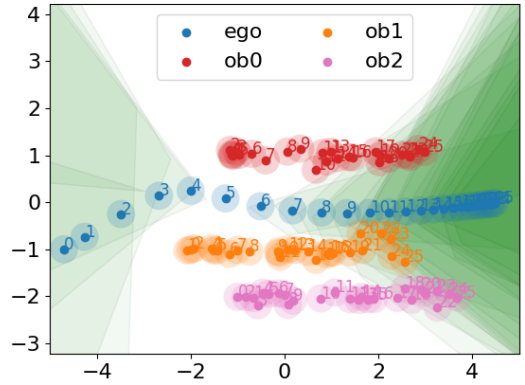


Fig. 5: Collision Avoidance with Multiple Obstacles. Green polytopes: DR-CVaR safe halfspaces. Blue: ego. All other colors: obstacles

On the other had, in all three cases, the DR-CVaR safe halfspace-based formulations avoid collisions with the ego vehicle, avoiding the obstacle even in the worst case scenario. Thus, our proposed solution can secure the robot's safety and systematically reduce collision risks, even with edge cases in the prediction distribution tail.

3) Safety Filtering With Multiple Obstacles: Consider the motion planning problem in Fig. 5, where the ego robot must avoid 3 obstacles. Fig. 5 plots the overall trajectories as well as the DR-CVaR safe halfspaces (green polytopes). Since the MPC safety filter is a QP, it can be solved efficiently in a few milliseconds. Here, the filter call time takes 7ms on average. The safety filter becomes infeasible only once, so we fall back to the previously computed set of optimal controls. This is illustrated at time step 5 when the ego robot is not inside a safe polytope. Throughout the experiment, the robot remains sufficiently far away from the obstacles and successfully reaches its goal.

V. Conclusion

We presented a safety filtering solution that improves a robot's safety when operating in a dynamic environment with prediction uncertainties. We posed a DRO problem that computes DR-CVaR safe halfspaces which are subsequently used as linear constraints in an MPC safety filter that corrects an ego reference trajectory. We performed numerical analyses demonstrating that DR-CVaR safe halfspaces are efficient to compute and can improve safety particularly in edge cases. The presented solution depends on a reference trajectory whose quality may influence the performance of the safety filter, and is limited to linear dynamics. Future directions will address these limitations and include a better approach to handling MPC infeasibilities, relaxation of the halfspaces to trade-off safety for performance, and implementation of the method on hardware.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[2] L. Wawrla, O. Maghazei, and T. Netland, "Applications of drones in warehouse operations," *Whitepaper. ETH Zurich, D-MTEC*, p. 212, 2019.

[3] S. Jung and H. Kim, "Analysis of amazon prime air uav delivery service," *Journal of Knowledge Information Technology and Systems*, vol. 12, no. 2, pp. 253–266, 2017.

[4] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar, "Devices, systems, and methods for automated monitoring enabling precision agriculture," in *2015 IEEE international conference on automation science and engineering (CASE)*. IEEE, 2015, pp. 462–469.

[5] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2017.

[6] E. Yurtsever, Y. Liu, J. Lambert, C. Miyajima, E. Takeuchi, K. Takeda, and J. H. Hansen, "Risky action recognition in lane change video clips using deep spatiotemporal networks with segmentation mask transfer," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3100–3107.

[7] A. Davies, "Google's self-driving car caused its first crash," in *Wired*, 2016.

[8] X. Geng, H. Liang, B. Yu, P. Zhao, L. He, and R. Huang, "A scenario-adaptive driving behavior prediction approach to urban autonomous driving," *Applied Sciences*, vol. 7, no. 4, p. 426, 2017.

[9] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 797–802.

[10] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[11] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 273–15 282.

[12] W. Zhou, Z. Cao, Y. Xu, N. Deng, X. Liu, K. Jiang, and D. Yang, "Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1275–1282.

[13] 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.

[14] A. Navsalkar and A. R. Hota, "Data-driven risk-sensitive model predictive control for safe navigation in multi-robot systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1442–1448.

[15] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[16] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *arXiv preprint arXiv:2210.10254*, 2022.

[17] B. Tearle, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "A predictive safety filter for learning-based racing control," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7635–7642, 2021.

[18] A. P. Vinod, S. Safaoui, A. Chakrabarty, R. Quirynen, N. Yoshikawa, and S. Di Cairano, "Safe multi-agent motion planning via filtered reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7270–7276.

[19] T. Phan-Minh, F. Howington, T.-S. Chu, S. U. Lee, M. S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin *et al.*, "Driving in real life with inverse reinforcement learning," *arXiv preprint arXiv:2206.03004*, 2022.

[20] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

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

[23] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2020, pp. 75–84.

[24] A. Hakobyan and I. Yang, "Distributionally robust optimization with unscented transform for learning-based motion control in dynamic environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3225–3232.

[25] 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, pp. 115–166, 2018.

[26] 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.

[27] S. Diamond and S. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[28] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *2013 European control conference (ECC)*. IEEE, 2013, pp. 3071–3076.