IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 December 2022, accepted 10 December 2022, date of publication 12 December 2022,
date of current version 19 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228922

RESEARCH ARTICLE

Safe Reinforcement Learning Using Wasserstein Distributionally Robust MPC and Chance Constraint

ARASH BAHARI KORDABAD[1], RAFAEL WISNIEWSKI[2], (Member, IEEE), AND SEBASTIEN GROS[1]

[1]Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway
[2]Department of Electronical Systems, Aalborg University, 9220 Aalborg, Denmark

Corresponding author: Arash Bahari Kordabad (arash.b.kordabad@ntnu.no)

This work was supported by the Research Council of Norway (RCN) Project "Safe Reinforcement Learning Using MPC" (SARLEM).

ABSTRACT In this paper, we address the chance-constrained safe Reinforcement Learning (RL) problem using the function approximators based on Stochastic Model Predictive Control (SMPC) and Distributionally Robust Model Predictive Control (DRMPC). We use Conditional Value at Risk (CVaR) to measure the probability of constraint violation and safety. In order to provide a safe policy by construction, we first propose using parameterized nonlinear DRMPC at each time step. DRMPC optimizes a finite-horizon cost function subject to the worst-case constraint violation in an ambiguity set. We use a statistical ball around the empirical distribution with a radius measured by the Wasserstein metric as the ambiguity set. Unlike the sample average approximation SMPC, DRMPC provides a probabilistic guarantee of the out-of-sample risk and requires lower samples from the disturbance. Then the Q-learning method is used to optimize the parameters in the DRMPC to achieve the best closed-loop performance. Wheeled Mobile Robot (WMR) path planning with obstacle avoidance will be considered to illustrate the efficiency of the proposed method.

INDEX TERMS Safe reinforcement learning, model predictive control, distributionally robust optimization, chance constraint, conditional value at risk, Q-learning.

I. INTRODUCTION

Enforcing safety in the presence of uncertainty and stochasticity of nonlinear dynamical systems is a challenging task [1]. Chance constraints are a common way of mathematical modeling of safety that requires a user-specified upper bound for the probability of the constraint violation [2]. However, it is challenging to handle a chance constraint from the computational point of view due to its nonconvexity. Conditional Value at Risk (CVaR) [3] is a convex risk measure that has received considerable attention in decision-making problems, such as Markov Decision Processes (MDPs) [4], [5].

The theory of stochastic optimal control typically assumes that the probability distribution of the disturbance is fully known. However, this assumption may not hold in many real-world applications, and one needs to estimate the probability distribution. However, stochastic optimization is challenging to solve, especially for non-convex problems [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Jinquan Xu.

In data-driven stochastic optimization, Sample Average Approximation (SAA) is a fundamental way to estimate the probability distribution of the random variables [7]. SAA typically needs quite an extensive data set to fulfill risk constraints accurately. Distributionally Robust Optimization (DRO) is an alternative that overcomes this problem. DRO tackles stochastic optimization by considering the worst-case distribution in an ambiguity set. There are several ways to construct ambiguity sets, e.g., moment ambiguity [8], Prohorov-based ball [9], Kullback–Leibler divergence-based ball [10] and Wasserstein-based ball [11]. The Wasserstein-based ball is a statistical ball in the space of probability distributions around the empirical distribution such that the radius of this ball is measured using Wasserstein distance. Then the radius of the ball represents the conservatism of the DRO problem. Unlike the SAA method, Wasserstein DRO provides a probabilistic guarantee based on finite samples in a tractable formulation [12].

Model Predictive Control (MPC) is an optimization-based control approach operating with a receding horizon [13]. MPC employs a (possibly inaccurate) model of the real

system dynamics to produce an input-state sequence over a given finite horizon. The resulting trajectory optimizes a given cost function while explicitly enforcing the system constraints. The optimization problem is solved at each time instance based on the current system state, and the first input of the optimal solution is applied to the system. Due to the finite-horizon scheme and (possibly) model mismatch, MPC usually delivers a reasonable but suboptimal approximation of the optimal policy. This paper uses the DRO in the chance-constrained nonlinear MPC. This approach has been known as Distributionally Robust MPC (DRMPC) [14].

Reinforcement Learning (RL) is a technique for solving problems involving MDPs. RL typically requires a function approximator to approximate the optimal policy, value function, or action-value function. For instance, Q-learning has been used in [15] for unmanned vehicle applications. In [16], the comparison of MPC and RL has been studied in the distributed setting. Recently, MPC has been used as a structured function approximator for RL algorithms. In this method, a parameterized MPC scheme is used in order to generate policy and/or value functions of the real system. Then RL algorithms can be used to adjust the MPC parameters to achieve the best closed-loop performance. The combination of MPC and RL has been proposed and justified in [17], where it is shown that an MPC scheme can theoretically generate the optimal policy and value functions for a given system even if the MPC model is inaccurate. Recent research have further developed and demonstrated this approach [18], [19].

A. RELATED WORKS

In [14], the authors have proposed to use DRMPC to utilize its benefits for motion control. A DRMPC has been applied to the multi-area dynamic optimal power flow in [20] to better hedge the uncertainties of distributed generation and loads. For the Gaussian processes, a learning-based DRMPC has been proposed in [21]. A learning-Based DRMPC has been developed for chance-constrained Markovian switching systems with unknown switching probabilities in [22]. The authors have shown that this framework provides mean-square stability of the system without requiring explicit knowledge of the transition probabilities. In [23], a DRO has been proposed for chance-constrained data-enabled predictive control with stochastic linear time-invariant systems. In [24], a DRMPC algorithm has been presented for spacecraft circular orbital rendezvous and docking problems. A soft-constrained DRMPC has been proposed for linear systems in [25].

A robust MPC scheme has been used as a function approximator for safe RL in [26]. Control Barrier Functions (CBF) have been used in the safe RL context in [27]. A safe RL-CBF framework has been developed to guarantee safety and improve exploration in [28]. Probabilistic safety in learning-based control methods has been provided in [29] based on probabilistic model predictive safety certification. In [30], the safe RL problem is formulated as a constrained MDP. Then a Lyapunov approach has been proposed to solve it.

B. CONTRIBUTIONS

There are a limited number of data from uncertainties and disturbances available in many real stochastic systems. Therefore, traditional methods such as SAA cannot accurately estimate the distribution of these random variables. An accurate distribution may be more important for safety-critical systems to design a safe controller for the system. In this paper, we propose to use a parameterized nonlinear DRMPC based on the Wasserstein metric as a function approximator for RL in order to generate a family of policies that are safe by construction. DRMPC is subject to the chance constraint, approximated by the CVaR risk measure. We reformulate Wasserstein DRMPC as a tractable optimization. Then we use the Q-learning technique to optimize the parameters of the DRMPC scheme to achieve the best closed-loop performance among the safe policies.

C. ORGANIZATION

The paper is structured as follows. Section II details safe RL and chance constraints. Section III provides safe policies based on the SMPC scheme, evaluated using the SAA method. Moreover, we formulate CVaR as a convex approximator of chance constraints. Section IV formulates a tractable DRMPC scheme and provides out-of-sample guarantees. Section V details Q-learning as an efficient way to optimize the parameters of the DRMPC scheme. Section VI provides a numerical simulation and section VII delivers a conclusion.

Notation: We denote the set of real numbers, non-negative real numbers, extended real numbers, non-negative integers, and natural numbers by $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$, $\mathbb{Z}$ and $\mathbb{N}$, respectively, while $\mathbb{I}_{i:j}$ refers to the set $\{i, i+1, \ldots, j\}$. Vectors in $\mathbb{R}^n$ are denoted by the bold letters, e.g., $\boldsymbol{a}$. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ denotes the usual inner product for given vectors $\boldsymbol{x}, \boldsymbol{y}$. A function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is proper if $f(\boldsymbol{x}) < +\infty$ for at least one $\boldsymbol{x}$ and $f(\boldsymbol{x}) > -\infty$ for every $\boldsymbol{x}$ in $\mathbb{R}^n$. The conjugate function of a function $f : \mathbb{R}^n \to \mathbb{R}$ is denoted by $[f]^\star(\boldsymbol{y}) := \sup_{\boldsymbol{x} \in \mathbb{R}^n} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x})$. Support function of set $\mathbb{W}$ is defined as $\Xi_{\mathbb{W}}(\boldsymbol{x}) := \sup_{\boldsymbol{y} \in \mathbb{W}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle$. For scalar a, we define $(a)_+ := \max\{a, 0\}$.

II. SAFE REINFORCEMENT LEARNING

In this section, we formulate safe Reinforcement Learning (RL) using chance constraints. Let us consider the following (possibly) nonlinear discrete-time stochastic dynamical system:

$$\boldsymbol{s}_{k+1} = \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{w}_k) \tag{1}$$

where $k \in \mathbb{Z}$ is the time index, $\boldsymbol{s}_k \in \mathbb{X} \subseteq \mathbb{R}^n$ is the system state, $\boldsymbol{a}_k \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input, $\boldsymbol{w}_k \in \mathbb{W} \subset \mathbb{R}^d$ is a random variable representing the stochastic disturbance of the system and $\boldsymbol{f} : \mathbb{R}^{n+m+d} \to \mathbb{R}^n$ is a Borel-measurable function. Note that the notation in (1) is standard in the literature of control, while the RL literature typically uses the conditional probability notation $\mathbb{P}[\boldsymbol{s}_{k+1} | \boldsymbol{s}_k, \boldsymbol{a}_k]$ for the state transition. We then make the following assumption on $\mathbb{W}$.

Assumption 1: The disturbance set $\mathbb{W}$ is convex and closed.

We will use this assumption in the rest of the paper to reformulate DRO as finite convex programming.

A deterministic policy $\boldsymbol{\pi} : \mathbb{X} \to \mathbb{U}$ maps the state space to the input space and determines how to choose input $\boldsymbol{a}_k$ at each state $\boldsymbol{s}_k$. We aim to find the optimal safe policy $\boldsymbol{\pi}^\star$, given by the solution of:

$$\boldsymbol{\pi}^\star \in \arg\min_{\boldsymbol{\pi}} \mathbb{E}_{\boldsymbol{s}_0 \sim \mu_0} \left[V^{\boldsymbol{\pi}}(\boldsymbol{s}_0) \right] \tag{2}$$

where μ_0 is the probability distribution of the initial state $\boldsymbol{s}_0$ and $V^{\boldsymbol{\pi}} : \mathbb{X} \to \mathbb{R}$ is the value function associated with the policy $\boldsymbol{\pi}$, defined as follows:

$$V^{\boldsymbol{\pi}}(\boldsymbol{s}_0) := \mathbb{E}_{\boldsymbol{w}} \left[\sum_{k=0}^{\infty} \gamma^k L(\boldsymbol{s}_k, \boldsymbol{\pi}(\boldsymbol{s}_k)) \right], \tag{3a}$$

$$\text{s.t. } \boldsymbol{s}_{k+1} = \boldsymbol{f}(\boldsymbol{s}_k, \boldsymbol{\pi}(\boldsymbol{s}_k), \boldsymbol{w}_k), \quad \forall k \in \mathbb{Z} \tag{3b}$$

$$\mathbb{P}[\boldsymbol{s}_{k+i} \in \mathcal{S} | \boldsymbol{s}_k] \geq \alpha, \quad \forall i \in \mathbb{I}_{1:I}, \quad \forall k \in \mathbb{Z} \tag{3c}$$

where $L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the stage cost, $\gamma \in (0, 1]$ is the discount factor, $\mathcal{S} \subseteq \mathbb{X}$ is a safe set and $\alpha \in (0, 1)$ is a user-chosen confidence level. The chance-constraint (3c) guarantees probabilistic safety of state trajectories $\boldsymbol{s}_{k+i}$ for a finite-horizon with length $I \in \mathbb{N}$ given state $\boldsymbol{s}_k$ at each time instance k. In fact, we generalize the common chance constraint in the literature not only to be satisfied for one step ahead but also to be satisfied for a finite horizon ahead at every time instance. This paper provides such policies using both an SMPC scheme and a DRMPC scheme with horizon I.

The safe set $\mathcal{S}$ can be defined as follows:

$$\mathcal{S} = \{\boldsymbol{s} \in \mathbb{X} | h_j(\boldsymbol{s}) \leq 0, \ \forall j \in \mathbb{I}_{1:J}\} \tag{4}$$

where $h_j : \mathbb{X} \to \mathbb{R}$ specifies a state constraint and J is the number of constraints. For the sake of simplicity and in order to avoid the complexity of joint constraints, we consider the following individual constraint:

$$\mathbb{P}[\max_j h_j(\boldsymbol{s}_{k+i}) \leq 0 | \boldsymbol{s}_k] \geq \alpha, \quad \forall i \in \mathbb{I}_{1:I} \tag{5}$$

Then one can verify that using (4), (5) implies (3c).

Assumption 2: Each function $-h_j$ is proper, convex and lower semi-continuous functions.

In the next section, we will use an SMPC scheme in order to provide a family of safe policies.

III. STOCHASTIC MPC-BASED POLICY

In the RL context, we consider a family of the parameterized policy given by $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$ and seek the best parameters $\boldsymbol{\theta}^\star$ that provide the best closed-loop performance. More specifically, (2) is reformulated as:

$$\boldsymbol{\theta}^\star \in \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}_0 \sim \mu_0} \left[V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{s}_0) \right] \tag{6}$$

Instead of solving (3) directly, we use a function approximator based on the MPC scheme to extract policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ that satisfies (3c) by construction for all parameters $\boldsymbol{\theta}$.

More specifically, consider the following parameterized SMPC at time instant k:

$$\min_{\boldsymbol{a}, \boldsymbol{s}} \ \mathbb{E} \left[T_{\boldsymbol{\theta}}(\boldsymbol{s}_{I+k|k}) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}) \,\middle|\, \boldsymbol{s}_{k|k} \right], \tag{7a}$$

$$\text{s.t. } \boldsymbol{s}_{i+k+1|k} = \boldsymbol{f}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}, \boldsymbol{w}_i), \quad \forall i \in \mathbb{I}_{0:I-1} \tag{7b}$$

$$\mathbb{P}[\max_j h_j(\boldsymbol{s}_{k+i|k}) \leq 0] \geq \alpha, \quad \forall i \in \mathbb{I}_{1:I} \tag{7c}$$

$$\boldsymbol{a}_{i+k|k} \in \mathbb{U}, \quad \boldsymbol{s}_{k|k} = \boldsymbol{s}_k, \tag{7d}$$

where $T_{\boldsymbol{\theta}} : \mathbb{X} \to \mathbb{R}$ and $l_{\boldsymbol{\theta}} : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is the parameterized terminal cost and stage cost, respectively. This parameterization allows one to provide a family of policies that are safe for all $\boldsymbol{\theta} \in \mathbb{R}^p$. Then by tuning the parameters $\boldsymbol{\theta}$ and reshaping the cost function and MPC-scheme, one can achieve the best closed-loop performance. Decision variables $\boldsymbol{a} = \{\boldsymbol{a}_{k|k}, \ldots, \boldsymbol{a}_{I+k-1|k}\}$ and $\boldsymbol{s} = \{\boldsymbol{s}_{k|k}, \ldots, \boldsymbol{s}_{I+k|k}\}$ are the input and state sequence, respectively. Then the parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ at time instance k is extracted as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\text{SMPC}}(\boldsymbol{s}_k) = \boldsymbol{a}_{k|k}^\star(\boldsymbol{\theta}, \boldsymbol{s}_k) \tag{8}$$

where $\boldsymbol{a}_{k|k}^\star$ is the solution of SMPC (7) corresponding to the first input $\boldsymbol{a}_{k|k}$.

The use of parameterized MPC scheme as a function approximator in order to capture the optimal policy and value function was proposed and justified in [17]. Moreover, the authors showed that RL methods such as Q-learning and policy gradient can be used in order to adjust the MPC scheme parameters and achieve the best closed-loop performance.

We ought to stress here that MPC scheme (7) provides a family of safe policy for all parameters $\boldsymbol{\theta}$ based on the best state-input sequence that minimizes a finite-horizon parameterized cost function of an MPC scheme. Obviously, a richer parameterization in the stage cost and terminal cost provides a more extensive set of policies. Then tuning the parameters $\boldsymbol{\theta}$ leads us to get the optimal policy among the provided policy families. We will detail Q-learning as a practical way of adjusting the parameters in Section V.

In order to tackle the chance constraint (7c), a natural measure of risk is value-at-risk VaR. For a random variable r and confidence level α, VaR_α is defined as follows:

$$\text{VaR}_\alpha(r) := \min\{\eta \in \mathbb{R} \,|\, \mathbb{P}(r \leq \eta) \geq \alpha\} \tag{9}$$

In fact, VaR represents the worst-case loss with probability α. Then one can show that:

$$\text{VaR}_\alpha(r) \leq 0 \Leftrightarrow \mathbb{P}(r \leq 0) \geq \alpha \tag{10}$$

Unfortunately, VaR is, in general, non-convex, and optimizing models involving VaR are numerically intractable for high-dimensional, non-normal distributions.

An alternative measure of risk is conditional value-at-risk CVaR, defined as follows:

$$\text{CVaR}_\alpha(r) := \min_{\eta \in \mathbb{R}} \mathbb{E} \left[\eta + \frac{(r - \eta)_+}{1 - \alpha} \right] \tag{11}$$

Indeed, CVaR is a *coherent* risk measure that satisfies conditions such as convexity and monotonicity [3]. Risk management with CVaR functions can be done quite efficiently. CVaR can be formulated with convex and linear programming methods, while VaR is comparably complicated to optimize. Detailed benefits and concepts of CVaR can be found in, e.g., [31].

It can be shown that for $\alpha \to 1$, CVaR can approximate VaR more accurately. i.e.:

$$\lim_{\alpha \to 1} \text{CVaR}_\alpha(r) - \text{VaR}_\alpha(r) = 0\,. \tag{12}$$

Note that in engineering applications, we often are interested in a very low probability of failure ($\alpha \to 1$). Then using CVaR, with the numerical and mathematical benefits, imposes a very low conservative on the problem. Using CVaR, MPC (7) can be approximated as follows:

$$\min_{\boldsymbol{a},\boldsymbol{s}} \quad \mathbb{E}\left[T_{\boldsymbol{\theta}}(\boldsymbol{s}_{I+k|k}) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}) \,\middle|\, \boldsymbol{s}_{k|k} \right], \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{s}_{i+k+1|k} = \boldsymbol{f}(\boldsymbol{s}_{i+k|k}, \boldsymbol{a}_{i+k|k}, \boldsymbol{w}_i), \quad \forall i \in \mathbb{I}_{0:I-1} \tag{13b}$$

$$\text{CVaR}_\alpha(\max_j h_j(\boldsymbol{s}_{k+i|k})) \le 0, \quad \forall i \in \mathbb{I}_{1:I} \tag{13c}$$

$$\boldsymbol{a}_{i+k|k} \in \mathbb{U}, \quad \boldsymbol{s}_{k|k} = \boldsymbol{s}_k, \tag{13d}$$

At each time k we first consider N_s, independent and identically distributed (i.i.d.) samples of the disturbance $\boldsymbol{w}_i$ and we denote these samples by $\boldsymbol{w}_i^m$, $i \in \mathbb{I}_{1:I}$ $m \in \mathbb{I}_{1:N_s}$. Then N_s scenarios are described as follows:

$$\boldsymbol{s}_{k+i|k}^m = \boldsymbol{f}(\boldsymbol{s}_{k+i-1|k}^m, \boldsymbol{a}_{k+i-1|k}^m, \boldsymbol{w}_i^m) \tag{14}$$

where $\boldsymbol{s}_{k+i|k}^m$ and $\boldsymbol{a}_{k+i|k}^m$ are the predicted state and input for m^{th} scenario at time $k+i$ given time k. We then define auxiliary variables x_i^m for $i \in \mathbb{I}_{1:I}$, $m \in \mathbb{I}_{1:N_s}$ in order to approximate CVaR, in (13c), in the following tractable Linear Programming (LP), $\forall i \in \mathbb{I}_{1:I}$:

$$\text{CVaR}_\alpha \quad (\max_j h_j(\boldsymbol{s}_{k+i|k})) \approx \tag{15a}$$

$$\min_{\eta_i, \boldsymbol{x}_i} \quad \eta_i + \frac{1}{(1-\alpha)N_s} \sum_{m=1}^{N_s} x_i^m \tag{15b}$$

$$\text{s.t.} \quad \max_j h_j(\boldsymbol{s}_{k+i|k}^m) - \eta_i \le x_i^m, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{15c}$$

$$0 \le x_i^m, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{15d}$$

where $\boldsymbol{x}_i = \{x_i^m\}_{m=1}^{N_s}$ and $\eta_i \in \mathbb{R}$. In [32], it has been shown that for $N_s \to \infty$ the approximation in (15) will converge to its exact value with probability one.

Substitution of (15) into (13) and using SAA, SMPC (13) reads as:

$$\min_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{x}} \quad \frac{1}{N_s} \sum_{m=1}^{N_s} \left(T_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+I|k}^m) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+i|k}^m, \boldsymbol{a}_{k+i|k}^m) \right) \tag{16a}$$

$$\text{s.t.} \quad \boldsymbol{s}_{k+i|k}^m = \boldsymbol{f}(\boldsymbol{s}_{k+i-1|k}^m, \boldsymbol{a}_{k+i-1|k}^m, \boldsymbol{w}_i^m), \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{16b}$$

$$\eta_i + \frac{1}{(1-\alpha)N_s} \sum_{m=1}^{N_s} x_i^m \le 0, \quad \forall i \in \mathbb{I}_{1:I} \tag{16c}$$

$$\max_j h_j(\boldsymbol{s}_{k+i|k}^m) - \eta_i \le x_i^m, \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{16d}$$

$$\boldsymbol{a}_{i+k|k}^m \in \mathbb{U}, \; 0 \le x_i^m, \; \boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k, \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{16e}$$

where $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^I, \boldsymbol{x} = \{\boldsymbol{x}_i\}_{i=1}^I$.

From a theoretical point of view, SMPC (16) requires $N_s \to \infty$ in order to provide an accurate approximation of the original MPC (13). In the next section, we will introduce DRMPC scheme to overcome this problem.

IV. DISTRIBUTIONALLY ROBUST MPC-BASED POLICY

In order to tackle the limited distributional information issue with finite-many sampling, we use Distributionally Robust Optimization (DRO) in the chance constraint of the MPC scheme. In this section, we suppress the subscript i, denoting the horizon index, to simplify the notations.

The core idea of the theoretical developments in this section was proposed in [12] for general optimization problems. For the sake of clarity, in the context of learning-based MPC, we detail these developments in this section.

We use the Wasserstein metric to define an ambiguity set as a ball around the empirical distribution $\hat{\mathcal{P}}$. Then the optimization will be solved with respect to the worst-case distribution in the ambiguity set. Empirical distribution $\hat{\mathcal{P}}$, evaluated from N_s i.i.d. samples $\{\boldsymbol{w}^m\}_{m=1}^{N_s}$, is defined as follows:

$$\hat{\mathcal{P}} = \frac{1}{N_s} \sum_{m=1}^{N_s} \delta_{\boldsymbol{w}^m} \tag{17}$$

where $\delta_{\boldsymbol{w}}$ is the Dirac measure concentrated at $\boldsymbol{w}$. Then we define the Wasserstein ball $\mathbb{D}$ around the empirical distribution $\hat{\mathcal{P}}$ as the ambiguity set as follows:

$$\mathbb{D} := \{\mathcal{P} \in \mathcal{P}(\mathbb{W}) \,|\, d_{\text{W}}(\mathcal{P}, \hat{\mathcal{P}}) \le \epsilon\} \tag{18}$$

where $\mathcal{P}(\mathbb{W})$ denotes the set of Borel probability measures on the support $\mathbb{W}$, $\epsilon \ge 0$ is the radius of the ball and $d_{\text{W}} : \mathcal{P}(\mathbb{W}) \times \mathcal{P}(\mathbb{W}) \to \mathbb{R}_{\ge 0}$ is the Wasserstein metric, defined as follows:

$$d_{\text{W}}(\mathcal{P}_1, \mathcal{P}_2) := \min_{\kappa \in \mathcal{P}(\mathbb{W}^2)} \left\{ \int_{\mathbb{W}^2} \|\boldsymbol{w}_1 - \boldsymbol{w}_2\| \mathrm{d}\kappa(\boldsymbol{w}_1, \boldsymbol{w}_2) \times \Big| \Pi^l \kappa = \mathcal{P}_l, l = 1, 2 \right\} \tag{19}$$

for all distributions $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{P}(\mathbb{W})$ where $\Pi^l \kappa$ denotes the lth marginal of the transportation plan κ for $l = 1, 2$ [33]. Indeed, the Wasserstein distance of $\mathcal{P}_1$ and $\mathcal{P}_2$ can be interpreted as the minimum transportation cost for moving the probability mass from $\mathcal{P}_1$ to $\mathcal{P}_2$. Then distributionally robust optimization minimizes the worst-case cost over all the distributions in the ambiguity set. Distributionally robust constraint (13c) can be written as follows:

$$\sup_{\mathcal{P} \in \mathbb{D}} \text{CVaR}_\alpha^{\mathcal{P}}(\max_j h_j(\boldsymbol{s})) \le 0 \tag{20}$$

For the sake of simplicity, we define a new variable $c := \max_j h_j(\boldsymbol{s})$. We then recall the definition of CVaR:

$$\text{CVaR}_\alpha^{\mathcal{P}}(c) = \min_\eta \mathbb{E}^{\mathcal{P}} \left[\eta + \frac{1}{1-\alpha}(c-\eta)_+ \right] \tag{21}$$

We then use the minimax inequality for (20):

$$\sup_{\mathcal{P}\in\mathbb{D}} \mathrm{CVaR}_\alpha^{\mathcal{P}}(c) \le \min_{\eta} \sup_{\mathcal{P}\in\mathbb{D}} \mathbb{E}^{\mathcal{P}}\left[\eta + \frac{1}{1-\alpha}(c-\eta)_+\right] = \min_{\eta} \eta + \frac{1}{1-\alpha} \sup_{\mathcal{P}\in\mathbb{D}} \mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] \tag{22}$$

on the other hand:

$$\sup_{\mathcal{P}\in\mathbb{D}} \mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] = \sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] \quad \text{s.t.} \quad d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}}) \le \epsilon \tag{23}$$

then using the Lagrangian function for the constrained optimization (23):

$$\sup_{\mathcal{P}\in\mathbb{D}} \mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] = \sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \inf_{\lambda\ge 0} \left\{\mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] + \lambda(\epsilon - d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}}))\right\} \tag{24}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier. Using Theorem 1 in [34]:

$$\begin{aligned}
&\sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \inf_{\lambda\ge 0} \left\{\mathbb{E}^{\mathcal{P}}\left[(c-\eta)_+\right] + \lambda(\epsilon - d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}}))\right\} \\
&= \inf_{\lambda\ge 0} \left\{\lambda\epsilon + \sup_{\mathcal{P}\in\mathcal{P}(\mathbb{W})} \left\{\mathbb{E}^{\mathcal{P}}[(c-\eta)_+] - \lambda d_{\mathrm{W}}(\mathcal{P},\hat{\mathcal{P}})\right\}\right\} \\
&= \inf_{\lambda\ge 0} \left\{\lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} \sup_{\boldsymbol{w}\in\mathbb{W}} \left\{(c-\eta)_+ - \lambda\|\boldsymbol{w}-\boldsymbol{w}^m\|\right\}\right\}
\end{aligned} \tag{25}$$

In fact, the first equality in (25) follows from the strong duality that has been shown in [34], and the second equality holds because $\mathcal{P}(\mathbb{W})$ contains all the Dirac distributions supported on $\mathbb{W}$.

Introducing a new auxiliary variable y^m, we can rewrite (25) as follows:

$$\inf_{\lambda,\boldsymbol{y}} \ \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m \tag{26a}$$

$$\text{s.t.} \ \sup_{\boldsymbol{w}\in\mathbb{W}} \{(c-\eta)_+ - \lambda\|\boldsymbol{w}-\boldsymbol{w}^m\|\} \le y^m, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{26b}$$

$$0 \le \lambda \tag{26c}$$

where $\boldsymbol{y} = \{y^m\}_{m=1}^{N_s}$. From the definition of dual norm, we decompose the expression inside $(\cdot)_+$ in constraint (26b) as follows [12]:

$$\sup_{\boldsymbol{w}\in\mathbb{W}} \left\{\min_{\|\boldsymbol{\xi}_1^m\|_\star\le\lambda} -\left\langle\boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle\right\} \le y^m \tag{27a}$$

$$\sup_{\boldsymbol{w}\in\mathbb{W}} \left\{\min_{\|\boldsymbol{\xi}_2^m\|_\star\le\lambda} -\left\langle\boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle + c\right\} - \eta \le y^m \tag{27b}$$

where $\|\cdot\|_\star := \sup_{\|\boldsymbol{\xi}\|\le 1}\langle\cdot,\boldsymbol{\xi}\rangle$ is the dual norm. Since $\{\boldsymbol{\xi} \mid \|\boldsymbol{\xi}\|_\star \le \lambda\}$ is a convex set, we use the minimax inequality, and (27) reads:

$$\min_{\|\boldsymbol{\xi}_1^m\|_\star\le\lambda} \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\left\langle\boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle\} \le y^m \tag{28a}$$

$$\min_{\|\boldsymbol{\xi}_2^m\|_\star\le\lambda} \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\left\langle\boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle + c\} - \eta \le y^m \tag{28b}$$

Then optimization (26) can be written as follows:

$$\inf_{\lambda,\boldsymbol{y},\boldsymbol{\xi}_1,\boldsymbol{\xi}_2} \ \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m$$

$$\text{s.t.} \ \sup_{\boldsymbol{w}\in\mathbb{W}} \{-\left\langle\boldsymbol{\xi}_1^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle\} \le y^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{29a}$$

$$\sup_{\boldsymbol{w}\in\mathbb{W}} \{-\left\langle\boldsymbol{\xi}_2^m, \boldsymbol{w}-\boldsymbol{w}^m\right\rangle + c\} - \eta \le y^m, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{29b}$$

$$\|\boldsymbol{\xi}_1^m\|_\star \le \lambda, \quad \|\boldsymbol{\xi}_2^m\|_\star \le \lambda, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{29c}$$

where $\boldsymbol{\xi}_l = \{\boldsymbol{\xi}_l^m\}_{m=1}^{N_s}$ for $l = 1, 2$. Changing $\boldsymbol{\xi}_l^m$ to $-\boldsymbol{\xi}_l^m$, we have:

$$\inf_{\lambda,y,\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\boldsymbol{v}} \quad \lambda\epsilon + \frac{1}{N_s}\sum_{m=1}^{N_s} y^m \tag{30a}$$

$$\text{s.t.} \quad -\left\langle\boldsymbol{\xi}_1^m, \boldsymbol{w}^m\right\rangle + \Xi_{\mathbb{W}}(\boldsymbol{\xi}_1^m) \le y^m, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{30b}$$

$$[-c]^\star(\boldsymbol{\xi}_2^m - \boldsymbol{v}^m) + \Xi_{\mathbb{W}}(\boldsymbol{v}^m) - \left\langle\boldsymbol{\xi}_2^m, \boldsymbol{w}^m\right\rangle - \eta \le y^m, \ \forall m \in \mathbb{I}_{1:N_s} \tag{30c}$$

$$\|\boldsymbol{\xi}_1^m\|_\star \le \lambda, \quad \|\boldsymbol{\xi}_2^m\|_\star \le \lambda, \quad \forall m \in \mathbb{I}_{1:N_s} \tag{30d}$$

where $[-c]^\star$ is the conjugate of $-c$ that is calculated at $\boldsymbol{\xi}_2^m - \boldsymbol{v}^m$. Note that under assumptions 1 and 2, (30) is a finite convex program [12]. Restoring the index i, the DRMPC scheme based on the Wasserstein metric reads as follows (31), shown at the bottom of the next page, where $\boldsymbol{\xi} = \{\boldsymbol{\xi}_{i,1}, \boldsymbol{\xi}_{i,2}\}_{i=0}^{I}$. Then the parameterized safe policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$ based on DRMPC scheme at time instance k is extracted as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}(\boldsymbol{s}_k) = \boldsymbol{a}_{k|k}^\star(\boldsymbol{\theta}, \boldsymbol{s}_k) \tag{32}$$

where $\boldsymbol{a}_{k|k}^\star$ is solution of DRMPC (31) corresponding to the first input $\boldsymbol{a}_{k|k}$. Note that all the optimal solutions of $\boldsymbol{a}_{k|k}^m$s are identical since the random samples are generated based on the first given state $\boldsymbol{s}_{k|k}^m = \boldsymbol{s}_k$ and the uncertainty cannot be anticipated [35]. Then we select one of the optimal solutions of $\boldsymbol{a}_{k|k}^m$s as $\boldsymbol{a}_{k|k}$.

A. OUT-OF-SAMPLE GUARANTEE

Unlike the SAA method, Wasserstein DRMPC provides a probabilistic guarantee on the out-of-sample performance with finitely many samples. More specifically, let us consider the following inequality:

$$\mathrm{CVaR}_\alpha^{\mathcal{P}}(\max_j h_j(\boldsymbol{s}_{k+i|k}^\star)) \le 0 \tag{33}$$

where $\boldsymbol{s}_{k+i|k}^\star$ is the optimal solution of (31) and $\mathcal{P}$ is an unknown arbitrary distribution. Then it is worth fulfilling

inequality (33) with high probability, i.e.:

$$\mathbb{P}\left\{\mathrm{CVaR}_{\alpha}^{\mathcal{P}}(\max_{j} h_j(\boldsymbol{s}^{\star}_{k+i|k})) \leq 0\right\} \geq 1-\beta \tag{34}$$

where β is a user-specified confidence level. It has been shown in [12], if the Wasserstein radius ϵ_i is chosen as follows:

$$\epsilon_i = \begin{cases} \left(\frac{\log(c_1\beta^{-1})}{c_2 N_s}\right)^{\frac{1}{\max\{d,2\}}} & \text{if} \quad N_s \geq \frac{\log(c_1\beta^{-1}))}{c_2} \\ \left(\frac{\log(c_1\beta^{-1})}{c_2 N_s}\right)^{\frac{1}{a}} & \text{if} \quad N_s < \frac{\log(c_1\beta^{-1}))}{c_2} \end{cases} \tag{35}$$

then (34) holds, where c_1, c_2 are positive constants. In fact, we have assumed that the measure concentration inequality holds [36], i.e., $B = \mathbb{E}^{\mathcal{P}}[\exp \|\boldsymbol{w}\|^a] \leq \infty$ for $a > 1$ (Light-tailed distribution), then c_1, c_2 depend on a, B and the disturbance dimension.

We must emphasize here that in practice, analysis and (probabilistic) finite sampling guarantees are essential in the context of RL and stochastic optimization because, in practice, there is typically limited access to real system data. This analysis can include various criteria in the context of RL, such as convergence rate [37], regret analysis [38], and performance [39].

The next Proposition summarizes the theoretical development of this section.

Proposition 1: Under assumptions 1 and 2, DRMPC has a tractable reformulation as (31) *and the extracted policy* $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$*, based on finite* N_s *i.i.d. samples, satisfies* (34) $\forall k \in \mathbb{Z}, \forall i \in \mathbb{I}_{1:I}, \forall \boldsymbol{\theta} \in \mathbb{R}^p$*, for a user-specified* β *and* α *and any distributions* $\mathcal{P}$*, if* ϵ_i *is selected from* (35).

B. FEASIBILITY PRE-FILTRATION

As discussed, satisfying a state-depend hard constraint with $\alpha = 1$ is generally impossible. The same problem exists when the required α is higher than the problem nature requirement. This problem arises in solving (31) when no solution is found. This problem is known as the infeasibility of optimization. A common way to solve the feasibility issue is to soften the constraints using slack variables. The slack variables are positive decision variables that allow inequalities to violate. However, the violation is penalized in the cost function.

A common way to use slack variables is by adding them to the original cost function. However, in this case, finding proper positive coefficients is still challenging. Another way to use slack variables is to build an optimization as a pre-filtration to find the feasible slack variables and then apply them to the original optimization problem. More specifically, we consider the following optimization problem:

$$\min_{\boldsymbol{s},\boldsymbol{a},\boldsymbol{\eta},\boldsymbol{\lambda},\boldsymbol{y},\boldsymbol{\xi},\boldsymbol{v},\boldsymbol{\sigma}} \quad \sum_{i=0}^{I} \sigma_i^2 \tag{36a}$$

$$\text{s.t.} \quad (31b)$$

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i\epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq \sigma_i,$$

$$0 \leq \sigma_i, \quad \forall i \in \mathbb{I}_{1:I} \tag{36b}$$

$$(31d) - (31g) \tag{36c}$$

with the optimal solutions $\sigma_i^{\star}$. We then replace constraint (31c) with the following constraint $\forall i \in \mathbb{I}_{1:I}$:

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i\epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq \sigma_i^{\star} \tag{37}$$

Then the DRMPC scheme always has a feasible solution. Note that inverting the procedure of obtaining DRMPC (31), we can see DRMPC (31) with the feasible constraint (37), equivalent to the following robust constraint:

$$\sup_{\mathcal{P}\in\mathbb{D}} \mathrm{CVaR}_{\alpha}^{\mathcal{P}}(\max_{j} h_j(\boldsymbol{s})) \leq \sigma_i^{\star} \tag{38}$$

while (20) may yield an infeasible solution. Note that DRMPC scheme provides a family of safe policies $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\mathrm{DRMPC}}$

$$\min_{\substack{\boldsymbol{s},\boldsymbol{a},\boldsymbol{\eta},\\ \boldsymbol{\lambda},\boldsymbol{y},\boldsymbol{\xi},\boldsymbol{v}}} \quad \frac{1}{N_s}\sum_{m=1}^{N_s}\left(T_{\boldsymbol{\theta}}(\boldsymbol{s}^m_{k+I|k}) + \sum_{i=0}^{I-1} l_{\boldsymbol{\theta}}(\boldsymbol{s}^m_{k+i|k}, \boldsymbol{a}^m_{k+i|k})\right)$$

(31a)

$$\text{s.t.} \quad \boldsymbol{s}^m_{k+i|k} = \boldsymbol{f}(\boldsymbol{s}^m_{k+i-1|k}, \boldsymbol{a}^m_{k+i-1|k}, \boldsymbol{w}^m_i), \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{31b}$$

$$\eta_i + \frac{1}{1-\alpha}\left[\lambda_i\epsilon_i + \frac{1}{N_s}\sum_{m=1}^{N_s} y_i^m\right] \leq 0, \quad \forall i \in \mathbb{I}_{1:I}$$

(31c)

$$-\left\langle \boldsymbol{\xi}^m_{i,1}, \boldsymbol{w}^m_i \right\rangle + \Xi_{\mathbb{W}}(\boldsymbol{\xi}_{i,1}) \leq y_i^m, \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{31d}$$

$$\left[-\max_{j} h_j\right]^{\star}(\boldsymbol{\xi}^m_{i,2} - \boldsymbol{v}^m_i) + \Xi_{\mathbb{W}}(\boldsymbol{v}^m_i) \tag{31e}$$

$$-\left\langle \boldsymbol{\xi}^m_{i,2}, \boldsymbol{w}^m_i \right\rangle - \eta_i \leq y_i^m, \ \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I}$$

$$\|\boldsymbol{\xi}^m_{i,1}\|_{\star} \leq \lambda_i, \quad \|\boldsymbol{\xi}^m_{i,2}\|_{\star} \leq \lambda_i, \ \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{31f}$$

$$\boldsymbol{a}^m_{i+k|k} \in \mathbb{U}, \ \boldsymbol{s}^m_{k|k} = \boldsymbol{s}_k, \quad \forall m \in \mathbb{I}_{1:N_s}, \quad \forall i \in \mathbb{I}_{1:I} \tag{31g}$$

for all tuning parameters $\boldsymbol{\theta}$. Therefore, in the next stage, it is necessary to update the parameters to achieve the best performance. The next section details the Q-learning method as a practical way of updating the parameters $\boldsymbol{\theta}$ to achieve the best closed-loop performance.

V. Q-LEARNING BASED ON DRMPC SCHEME

Q-learning is a powerful, well-known, and popular method in the field of RL, whose use is practical due to relatively low-cost computational efforts, especially in engineering and economic applications [40]. In fact, Q-learning is a classical model-free RL algorithm that tries to capture the optimal action-value function $Q_{\boldsymbol{\theta}} \approx Q^{\star}$ via tuning the parameters vector $\boldsymbol{\theta}$ where $Q_{\boldsymbol{\theta}}$ is the parameterized action-value function, and $Q^{\star}$ is the optimal action-value function [41]. The optimal action-value function $Q^{\star}$ is defined as follows:

$$Q^{\star}(\boldsymbol{s}_k, \boldsymbol{a}_k) = L(\boldsymbol{s}_k, \boldsymbol{a}_k) + \gamma \min_{\boldsymbol{\pi}} \mathbb{E}[V^{\boldsymbol{\pi}}(\boldsymbol{s}_{k+1}) | \boldsymbol{s}_k, \boldsymbol{a}_k] \quad (39)$$

The parameterized action-value function $Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k)$ based on DRMPC scheme (31) can be formulated as follows:

$$\begin{aligned} Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) = & \min_{\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}} && (31a) && (40a) \\ & \text{s.t.} && (31b),\ (37),\ (31d)-(31g) && \\ & && && (40b) \\ & && \boldsymbol{a}_{k|k} = \boldsymbol{a}_k, && (40c) \end{aligned}$$

while the approximation of the value function $V_{\boldsymbol{\theta}}$ can be extracted from (40) when constraint (40c) is removed. Then one can verify that the MPC-based action-value function and value function satisfy the fundamental Bellman equations [17]. Q-learning solves the following Least Square (LS) problem:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\left(Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) - Q^{\star}(\boldsymbol{s}_k, \boldsymbol{a}_k)\right)^2\right]. \quad (41)$$

In order to solve (41), Temporal-Difference (TD) method uses the following update rule for the parameters $\boldsymbol{\theta}$ at state $\boldsymbol{s}_k$ [42]:

$$\begin{aligned} \delta_k &= L(\boldsymbol{s}_k, \boldsymbol{a}_k) + \gamma V_{\boldsymbol{\theta}}(\boldsymbol{s}_{k+1}) - Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) && (42a) \\ \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} + \zeta \delta_k \nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}_k, \boldsymbol{a}_k) && (42b) \end{aligned}$$

where the scalar $\zeta > 0$ is the learning step-size, δ_k is labelled the TD error, and the input $\boldsymbol{a}_k$ is selected according to the corresponding parametric policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{\text{DRMPC}}(\boldsymbol{s}_k)$ with the possible addition of small random exploration such that it preserves the safety. The gradient $\nabla_{\boldsymbol{\theta}} Q_{\boldsymbol{\theta}}$ required in (42) can be obtained by a sensitivity analysis on the DRMPC scheme (40) as detailed in [17] for generic MPC schemes.

In order to generate $\boldsymbol{a}_k$, we first add a small exploration noise to the policy, i.e.:

$$\boldsymbol{a}_k^{\text{e}}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k) = \boldsymbol{\pi}_{\boldsymbol{\theta}}^{\text{DRMPC}}(\boldsymbol{s}_k) + \boldsymbol{\rho}_k \quad (43)$$

where $\boldsymbol{\rho}_k \in \mathcal{E}$ is a random variable providing the exploration. One can easily observe that $\boldsymbol{a}_k^{\text{e}}$ may not deliver a safe input. Therefore a safety filtration based on the DRMPC scheme is needed to provide safe exploration, more specifically consider the following parametric DRMPC scheme with the additional parameter $\boldsymbol{\rho}_k$:

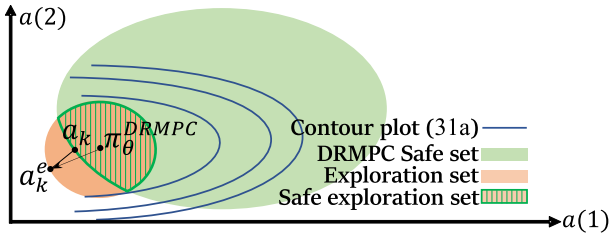

FIGURE 1. The illustration of the safe exploration for the Q-learning method in a 2-input system. Safe exploration input $a_k \in$ *safe exploration set*, while $a_k^{\text{e}} \in$ *Exploration set* and $\pi_\theta^{\text{DRMPC}}(s_k) \in$ *DRMPC Safe set*.

$$\begin{aligned} \min_{\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v}} & \quad \|\boldsymbol{a}_{k|k} - \boldsymbol{a}_k^{\text{e}}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k)\| && (44a) \\ \text{s.t.} & \quad (40b) && (44b) \end{aligned}$$

Then $\boldsymbol{a}_k(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k) = \boldsymbol{a}_{k|k}^{\star}(\boldsymbol{\theta}, \boldsymbol{s}_k, \boldsymbol{\rho}_k)$ delivers a safe input after exploration where $\boldsymbol{a}_{k|k}^{\star}$ is the optimal solution of (44) for the first input. Fig. 1 illustrates the safe exploration based on the DRMPC scheme. DRMPC safe set is defined as follows:

$$\text{DRMPC safe set} := \{\boldsymbol{a}_{k|k} \mid \exists \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{v} : (40b)\}$$

In the policy gradient method, the projection technique results in a bias in the gradient of the performance function [43]. Roughly speaking, this is because the safe exploration set may not be a centered ball, as shown in fig. 1. We have proposed a robust MPC scheme in [44] to solve the bias issue. The proposed method can be easily applied to the DRMPC scheme for the policy gradient method, but applying it is out of the focus of the current work.

Fig.2 illustrates the proposed safe learning method using the DRMPC scheme.

Remark 1: The proposed method can be applied to the general nonlinear stochastic dynamics with an unknown distribution of stochasticity. Obviously, the computational efforts are increased as the dimension and complexity of the dynamics grow.

Remark 2: In this paper, we do not focus on the convergence of the learning method. It is well-known that under the mild assumptions, the Q-learning technique generates a sequence of the parameters $\boldsymbol{\theta}$ that converge to the parameters that best estimate the exact optimal action-value function. Then the extracted policy is an optimal policy among the provided safe policies. The convergence conditions for the Q-learning method can be found in, e.g., [45].

Remark 3: Closed-loop stability of the policy for the nominal model used in the MPC scheme resulting from an MPC scheme is straightforward under some mild assumptions on the stage cost and terminal cost and constraints. However, these conditions are not painless for general stochastic systems and stochastic and robust MPC. This aspect is not the main scope of the current work. However, in the functional space, the closed-loop stability properties are recently addressed in [46] for general stochastic systems (MDP).

The proposed approach has been summarized in Algorithm 1.

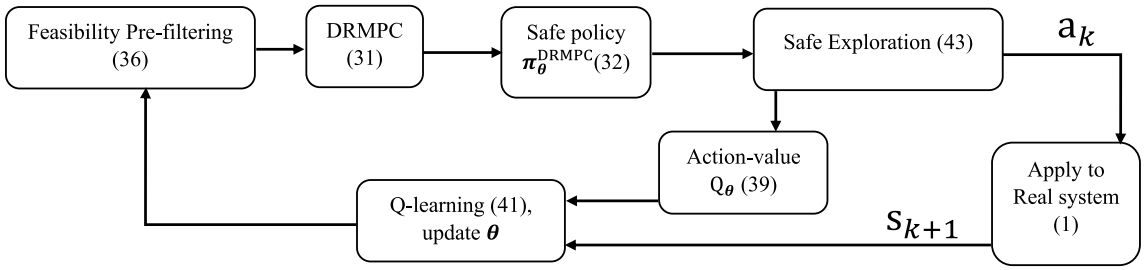

FIGURE 2. Schematics of the proposed safe RL using DRMPC scheme.

Algorithm 1 Using DRMPC Based Q-Learning to Provide Optimal Safe Policy

Input: α, β, I, γ, parameterize l_θ, T_θ
Initialize : s_0, θ_0
while θ *converges* **do**
  for $k = 0, \ldots, k$ *(end of the mission)* **do**
    Initialize : $s^m_{k|k} = s_k$,
    run feasibility pre-filtration (36) to get $\sigma_i^\star$,
    run DRMPC (31) with the parameters θ_k and relaxed constraint (37) to get safe policy $\pi_\theta^{\text{DRMPC}}(s_k)$,
    apply the safe exploration using (43) and (44) to get the input a_k,
    apply the input a_k to the dynamics (1) to get s_{k+1},
    update parameters $\theta_{k+1} \leftarrow \theta_k$ using Q-learning technique, e.g., (42) (ϵ_i s are among the parameters),
  Save the last parameters $\theta_0 = \theta_{k+1}$
end

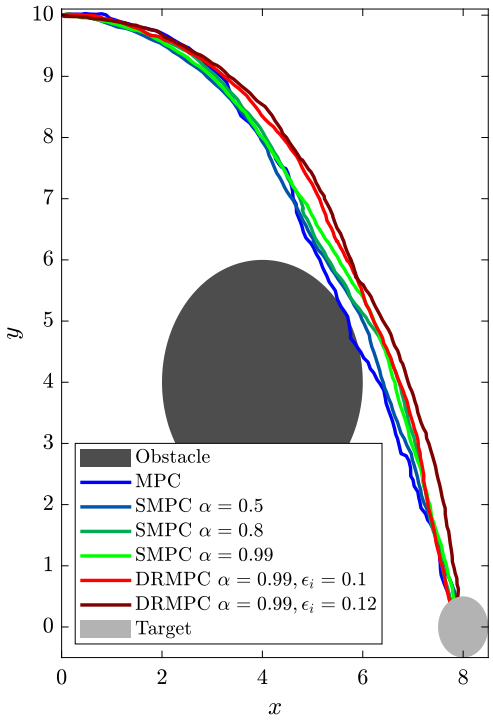

FIGURE 3. Sample average approximation of SMPC and DRMPC with CVaR constraints.

The next section provides a numerical case study for the proposed method.

VI. NUMERICAL SIMULATION

In this section, we consider Wheeled Mobile Robot (WMR) path planning while avoiding static obstacles. The stochastic nonlinear dynamics can be considered as follows:

$$s_{k+1} = \begin{bmatrix} t_e \cos(\phi_k) & 0 \\ t_e \sin(\phi_k) & 0 \\ 0 & t_e \end{bmatrix} a_k + s_k + w_k \tag{45}$$

where $s_k = [x_k, y_k, \phi_k]^\top$, $a_k = [v_k, \psi_k]^\top$ and $\|w_k\|_\infty \leq 0.1$ are the system state, input, and disturbance, respectively. x_k and y_k are the position of the robot in two dimensions and $\phi_k \in [-\pi, \pi]$ is the orientation angle. Sampling time t_e is selected 0.2sec for the simulation. The control inputs v_k and ψ_k are the linear and angular velocities, respectively. The control input is restricted as follows:

$$\begin{bmatrix} 0 \\ -1 \end{bmatrix} \leq a_k \leq \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \tag{46}$$

For simplicity, we consider obstacles of elliptic shape. Hence, the condition for obstacles avoidance can be seen as the following inequality:

$$h_j(s) = 1 - \left(\frac{x - o_{x,j}}{r_{x,j}}\right)^2 - \left(\frac{y - o_{y,j}}{r_{y,j}}\right)^2 \tag{47}$$

where $(o_{x,j}, o_{y,j})$ and $(r_{x,j}, r_{y,j})$ are the center and radii of the j^{th} ellipse ($j = 1, \ldots, J$), respectively, and J is number of obstacles.

First, we simulate SMPC with CVaR constraints based on Sample average approximation and DRMPC, and we compare them with deterministic MPC. As shown in figure 3, there are some constraint violations in the MPC scheme. As the probability level α increases, the distance from the path and obstacle increases in SMPC. As mentioned, this method usually requires a large number of data to capture the chance constraint accurately. Moreover, as shown in figure 3, the planned path using DRMPC is farther from the obstacle. We then consider the following stage cost for the

$$L(\boldsymbol{s},\boldsymbol{a}) = \|\boldsymbol{a}\| + |\phi| + \underbrace{|x-8| + |y| - \frac{1}{\tau}\log\left(\frac{1}{2}\left(|\max_j h_j(\boldsymbol{s})| - \max_j h_j(\boldsymbol{s})\right) + \omega\right)}_{r(x,y)} \tag{48}$$

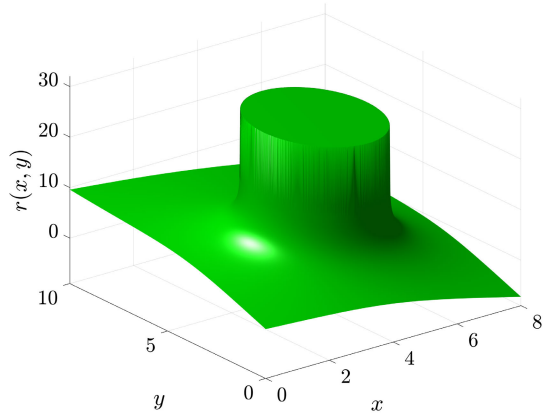

FIGURE 4. The function $r(x, y)$ for $\tau = 0.2$ and $\omega = 10^{-4}$.

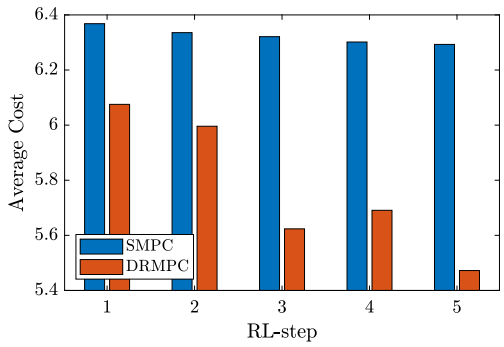

FIGURE 5. Average costs of five missions during Q-learning from SMPC and DRMPC.

RL (48), as shown at the top of the page, where τ and ω are small positive constants. Since h_j only depends on x, y, function r also depends on x, y. Note that the logarithmic barrier function has been inspired by the constrained optimization context [47]. Moreover, this function allows us to compute the logarithm for every $\boldsymbol{s}$, while it has a large value when the constraints violate. Figure 4 illustrates $r(x, y)$. We include the radius of the Wasserstein ball in the DRMPC parameters to tune it using Q-learning. Figure 5 shows the average stage costs during each mission. As can be seen, the average stage costs are decreasing in five missions in both SMPC and DRMPC. However, DRMPC has lower average costs, and Q-learning is more effective in the DRMPC scheme than in the SMPC scheme. The better improvement in the DRMPC scheme is due to the more freedom and parameters in the provided policies, such as the radius of the Wasserstein ball around the empirical distribution, whereas in the standard SMPC scheme, there is no such parameter. Obviously, tuning the radius of the Wasserstein ball and, consequently, adjusting the conservatism of the safe policy positively impacts the improvement of the closed-loop performance.

VII. CONCLUSION

In this paper, we proposed to use a tractable Distributionally Robust MPC (DRMPC) scheme in order to provide safe policy for Reinforcement Learning (RL) by construction. DRMPC optimized the cost function subject to the worst-case distribution in a given statistical ball around the empirical distribution. The radius of this ball was measured using the Wasserstein metric. Moreover, Conditional Value at Risk (CVaR) was used as a convex approximator of chance constraints in the DRMPC scheme. We used Q-learning to update the parameters of the DRMPC scheme. We showed the efficiency of the method in the path planning of a Wheeled Mobile Robot (WMR). Considering model mismatch, joint chance-constrained and Neural Network based cost functions in the DRMPC scheme will be the directions of future works.

REFERENCES

[1] B. Lutjens, M. Everett, and J. P. How, "Safe reinforcement learning with model uncertainty estimates," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8662–8668.
[2] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, 1999.
[3] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *J. Risk*, vol. 2, pp. 21–42, Feb. 2000.
[4] Y. Chow and M. Ghavamzadeh, "Algorithms for CVaR optimization in MDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[5] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-sensitive and robust decision-making: A CVaR optimization approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1522–1530.
[6] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization-aided energy-efficient information collection in internet of underwater things networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1775–1789, Feb. 2022.
[7] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM J. Optim.*, vol. 12, no. 2, pp. 479–502, 2002.
[8] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Oper. Res.*, vol. 58, no. 3, pp. 595–612, 2010.
[9] E. Erdoğan and G. Lyengar, "Ambiguous chance constrained problems and robust optimization," *Math. Programm.*, vol. 107, nos. 1–2, pp. 37–61, 2006.
[10] Z. Hu and L. J. Hong, "Kullback–Leibler divergence constrained distributionally robust optimization," *Optim. Online*, pp. 1695–1724, 2013.
[11] G. Pflug and D. Wozabal, "Ambiguity in portfolio selection," *Quant. Finance*, vol. 7, no. 4, pp. 435–442, Aug. 2007.
[12] P. M. Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, nos. 1–2, pp. 115–166, Sep. 2018.
[13] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Madison, WI, USA: Nob Hill, 2017.
[14] A. Hakobyan and I. Yang, "Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 939–957, Apr. 2022.
[15] W. Wei, J. Wang, Z. Fang, J. Chen, Y. Ren, and Y. Dong, "3U: Joint design of UAV-USV-UUV networks for cooperative target hunting," *IEEE Trans. Veh. Technol.*, early access, Nov. 9, 2022.

[16] I. Saeed, T. Alpcan, S. M. Erfani, and M. B. Yilmaz, "Distributed nonlinear model predictive control and reinforcement learning," in *Proc. Austral. New Zealand Control Conf. (ANZCC)*, Nov. 2019, pp. 1–3.
[17] S. Gros and M. Zanon, "Data-driven economic NMPC using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 636–648, Feb. 2020.
[18] A. Bahari Kordabad, W. Cai, and S. Gros, "MPC-based reinforcement learning for economic problems with application to battery storage," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 2573–2578.
[19] A. B. Kordabad, W. Cai, and S. Gros, "Multi-agent battery storage management using MPC-based reinforcement learning," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2021, pp. 57–62.
[20] W. Huang, W. Zheng, and D. J. Hill, "Distributionally robust optimal power flow in multi-microgrids with decomposition and guaranteed convergence," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 43–55, Jan. 2021.
[21] A. Hakobyan and I. Yang, "Learning-based distributionally robust motion control with Gaussian processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 7667–7674.
[22] M. Schuurmans and P. Patrinos, "Learning-based distributionally robust model predictive control of Markovian switching systems with guaranteed stability and recursive feasibility," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 4287–4292.
[23] J. Coulson, J. Lygeros, and F. Dorfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3289–3304, Jul. 2022.
[24] B. Li, Z. X. Li, and K. Zhang, "Distributionally model predictive control for spacecraft rendezvous and docking," in *Advances in Guidance, Navigation and Control*. Cham, Switzerland: Springer, 2022, pp. 4447–4457.
[25] S. Lu, J. H. Lee, and F. You, "Soft-constrained model predictive control based on data-driven distributionally robust optimization," *AIChE J.*, vol. 66, no. 10, Oct. 2020, Art. no. e16546.
[26] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, Aug. 2021.
[27] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *Int. J. Robust Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, Apr. 2021.
[28] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3387–3395.
[29] H. J. Wabersich, L. Hewing, A. Carron, and M. N. Zeilinger, "Probabilistic model predictive safety certification for learning-based control," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 176–188, Jan. 2022.
[30] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh, "A Lyapunov-based approach to safe reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8092–8101.
[31] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *J. Banking Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.
[32] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia, PA, USA: SIAM, 2021.
[33] L. V. Kantorovich and S. G. Rubinshtein, "On a space of totally additive functions," *Vestnik St. Petersburg Univ., Math.*, vol. 13, no. 7, pp. 52–59, 1958.
[34] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with Wasserstein distance," 2016, *arXiv:1604.02199*.
[35] E. Klintberg, J. Dahl, J. Fredriksson, and S. Gros, "An improved dual Newton strategy for scenario-tree MPC," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 3675–3681.
[36] N. Fournier and A. Guillin, "On the rate of convergence in Wasserstein distance of the empirical measure," *Probab. Theory Related Fields*, vol. 162, nos. 3–4, pp. 707–738, Aug. 2015.
[37] G. Dalal, G. Thoppe, B. Szörényi, and S. Mannor, "Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning," in *Proc. Conf. Learn. Theory*, 2018, pp. 1199–1233.
[38] Z. Zhou, Z. Zhou, Q. Bai, L. Qiu, J. Blanchet, and P. Glynn, "Finite-sample regret bound for distributionally robust offline tabular reinforcement learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3331–3339.
[39] Z. Yang, K. Zhang, M. Hong, and T. Başar, "A finite sample analysis of the actor-critic algorithm," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 2759–2764.
[40] A. B. Kordabad and S. Gros, "Q-learning of the storage function in economic nonlinear model predictive control," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105343. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197622003694
[41] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[42] C. Szepesväri, "Algorithms for reinforcement learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, 2010.
[43] S. Gros and M. Zanon, "Bias correction in reinforcement learning via the deterministic policy gradient method for MPC-based policies," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 2543–2548.
[44] A. B. Kordabad, H. Nejatbakhsh Esfahani, and S. Gros, "Bias correction in deterministic policy gradient using robust MPC," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 1086–1091.
[45] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[46] S. Gros and M. Zanon, "Economic MPC of Markov decision processes: Dissipativity in undiscounted infinite-horizon optimal control," *Automatica*, vol. 146, Dec. 2022, Art. no. 110602.
[47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

ARASH BAHARI KORDABAD received the B.Sc. degree in mechanical engineering from the University of Tabriz, Tabriz, Iran, in 2017, and the M.Sc. degree in mechanical engineering from the Sharif University of Technology, Tehran, in 2019. He is currently pursuing the Ph.D. degree with the Department of Cybernetics Engineering, Norwegian University of Science and Technology, Trondheim, Norway. His research interests include reinforcement learning, model predictive control, and optimization for autonomous vehicles and smart-grid applications.

RAFAEL WISNIEWSKI (Member, IEEE) received the Ph.D. degree in electrical engineering, in 1997, and the Ph.D. degree in mathematics, in 2005. He is currently a Professor and the Head of the Section of Automation and Control, Department of Electronic Systems, Aalborg University. From 2007 to 2008, he was a Control Specialist at Danfoss A/S. His research interest includes system theory, particularly in hybrid systems.

SEBASTIEN GROS received the Ph.D. degree from the Swiss Federal Institute of Technology Lausanne, Switzerland, in 2007. After a journey by bicycle from Switzerland to the Everest base camp in full autonomy, he joined a Research and Development Group, Strathclyde University, Glasgow, U.K., focusing on wind turbine control. In 2011, he joined the University of KU Leuven, Leuven, Belgium, where his main research focus was on optimal control and fast MPC for complex mechanical systems. In 2013, he joined the Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden, where he became an Associate Professor, in 2017. He is currently a Full Professor with NTNU, Norway, and a Guest Professor with Chalmers. His research interests include numerical methods, real-time optimal control, reinforcement learning, and the optimal control of energy-related applications.