2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Learning-based Safety Critical Model Predictive Control using Stochastic Control Barrier Functions

Hossein Nejatbakhsh Esfahani, Sajad Ahmadi, Javad Mohammadpour Velni

Abstract— **This paper presents a learning-based safety-critical Model Predictive Control (MPC) design approach based on stochastic Control Barrier Functions (CBFs). To address the safety concerns and tackle model uncertainties in both the MPC and CBF, we first propose to use a parameterized stochastic CBF in the MPC scheme. We next devise a Reinforcement Learning (RL)-based algorithm based on the proposed stochastic CBF-MPC scheme to learn the approximate version of the proposed stochastic CBF for coping with an unknown CBF model, which cannot capture the correct structure of the CBF used in the real environment. To illustrate the performance of the proposed safety-critical control approach, we examine two test cases including trajectory tracking and path planning for a wheeled mobile robot.**

I. Introduction

Control barrier functions (CBFs) have become a popular tool for synthesizing safety-critical controllers due to their generality and relative ease of synthesis and implementation [1]. Barrier functions made their debut in optimization, and now they are frequently mentioned in control and verification literature because of their bond with Lyapunov-like functions, their ability to establish safety and collision avoidance, and their association with multi-objective control [2], [3], [4]. Although CBF is a popular tool to achieve provable safety guarantees, designing CBFs and calculating the corresponding safe control inputs may be nontrivial if the dynamics are complex.

In the last decade, there have been studies on incorporating safety measures into model predictive control (MPC). MPC is a popular, widely used and practical approach to optimal control design. This optimization-based control approach is often desired for its capability to handle both input and state constraints [5]. Previous studies have explored the inclusion of safety considerations within MPC framework to ensure that the controlled system operates within predefined safety boundaries. For example, barrier functions were utilized to develop a safety-critical MPC with CBFs [6].

Safety criteria within the scope of model predictive control (MPC) are commonly expressed as constraints within the underlying optimization problem, as demonstrated in previous works [6], [7]. These constraints include factors such as obstacle constraints and actuation limits. An example of a specific situation where safety criteria are relevant is obstacle avoidance. However, they only restrict the movement of an agent when it is in a close proximity to obstacles. In order to prompt the robot to take preventive actions even when obstacles are far away, a larger prediction horizon is typically required. Nevertheless, this elongated horizon leads to increased computational time during the optimization process. As a result, there is a motivation to develop a new form of model predictive control that ensures safety within the framework of set invariance. This approach employs the CBF constraints to confine the robot's movement throughout the optimization process [3].

Reinforcement learning (RL) has attracted high attention in the control systems community as a tool for solving Markov Decision Processes (MDPs) without prior knowledge of the process to be formulated as an MDP. The RL algorithms typically rely on Deep Neural Networks (DNNs) as function approximators [8], [9]. An MPC-based RL framework was proposed in [10] which showed that by adjusting not only the MPC model parameters but also the parameters in the MPC cost (terminal and stage costs) and constraints, the MPC scheme can, theoretically, generate the optimal closed-loop policy even if a simple and inaccurate predictive model is used. Moreover, a parameterized MPC scheme was used as a function approximator instead of a DNN required in both the Q-learning and the policy gradient methods [11], [12], [13].

In this paper, we leverage the core idea of using an MPC-based RL framework combined with the control barrier functions in order to provide a safety-critical MPC design scheme. In the proposed learning-based CBF-MPC, we consider a stochastic setting upon the CBF-MPC to cope with the problems induced by the uncertainties in both the MPC and CBF models. However, the proposed stochastic framework is constructed based on an approximate model of the uncertainty propagation via a deterministic model of the stochastic chance-constraint CBF. Moreover, we assume that the CBF model used in the MPC scheme cannot capture the true CBF in the real environment. To tackle these problems, a parameterized version of the stochastic CBF and MPC scheme is then introduced where the corresponding parameters are adjusted by RL to improve the closed-loop performance in the presence of model uncertainties and unknown CBF.

This paper is structured as follows. Introductory information on the CBFs is provided in Section II. Section III describes a parameterization of the stochastic CBF-MPC scheme, which is learned by RL. To achieve the best closed-loop performance, an MPC-based policy gradient algorithm is detailed in Section IV in order to adjust the parameterized CBF-MPC and learn a policy captured from the proposed stochastic CBF-MPC scheme. Finally, numerical examples

This work was financially supported by the United States National Science Foundation under award CMMI-2302217.

Authors are with Department of Mechanical Engineering, Clemson University, Clemson, SC, USA. {hnejatb, sahmadi, javadm}@clemson.edu.

979-8-3503-8265-5/$31.00 ©2024 AACC

are presented in Section V.

II. Control Barrier Functions

In the context of safety-critical systems, CBFs are adopted to provide an admissible control input space for safety assurance of dynamical systems. More specifically, safety can be formulated by enforcing invariance of a set, i.e., not leaving a safe set. Let us consider a set $\mathcal{C}$ defined as the super-level set of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$ such that

$$\begin{aligned} &\mathcal{C} = \{\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}, \\ &\partial\mathcal{C} = \{\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^n : h(\boldsymbol{x}) = 0\}, \\ &\mathrm{Int}(\mathcal{C}) = \{\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^n : h(\boldsymbol{x}) > 0\}, \end{aligned} \tag{1}$$

where $\partial\mathcal{C}$ and $\mathrm{Int}(\mathcal{C})$ are the boundary of $\mathcal{C}$ and the interior of $\mathcal{C}$, respectively. We additionally assume that $\mathrm{Int}(\mathcal{C}) \neq \emptyset$. We then refer to $\mathcal{C}$ as the *safe set* so that a CBF certifies whether a control policy achieves forward invariance of $\mathcal{C}$ by evaluating if the system trajectory remains away from the boundary of $\mathcal{C}$. Let us consider a control-affine system as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz continuous functions, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$ are the system states and control inputs. The closed-loop dynamics of the system then is

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_{\mathrm{cl}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + g(x)\boldsymbol{u}|_{\boldsymbol{\pi}(\boldsymbol{x})} \tag{3}$$

where the control policy (feedback controller) $\boldsymbol{\pi} : \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitz continuous. Then, one can consider a maximum interval of existence $I(\boldsymbol{x}_0) = [t_0, t_{\max})$ for any initial condition $\boldsymbol{x}_0 \in \mathcal{D}$ such that $\boldsymbol{x}(t)$ is the unique solution to (3) on $I(\boldsymbol{x}_0)$. In the case $t_{\max} = \infty$, the closed-loop system $\boldsymbol{f}_{\mathrm{cl}}$ is forward complete.

Definition 1. *(Forward Invariance) The closed-loop system* (3) *is forward invariant w.r.t. the set $\mathcal{C}$ if for every $\boldsymbol{x}_0 \in \mathcal{C}$, we have $\boldsymbol{x}(t) \in \mathcal{C}$ for all $t \in I(\boldsymbol{x}_0)$.*

Definition 2. *(Control Barrier Function) Given a dynamical system* (2) *and the safe set $\mathcal{C}$ with a continuously differentiable function $h : \mathcal{D} \to \mathbb{R}$, then h is a CBF if there exists a class $\mathcal{K}_\infty$ function κ for all $\boldsymbol{x} \in \mathcal{D}$ such that*

$$\sup_{\boldsymbol{u}\in\mathcal{U}} \left\{\dot{h}(\boldsymbol{x},\boldsymbol{u})\right\} \geq -\kappa(h(\boldsymbol{x})) \tag{4}$$

where $\dot{h}(\boldsymbol{x},\boldsymbol{u}) = L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u}$ with $L_f h, L_g h$ being the Lie derivatives of h along the vector fields $\boldsymbol{f}$ and g, respectively [3]. As a common choice of κ, one can use a linear form $\kappa(h(\boldsymbol{x})) = \alpha h(\boldsymbol{x})$, where $\alpha \geq 0$ is a parameter controlling the system behavior near the boundary of $\mathcal{C}$.

Remark 1. *In this paper, we consider a discrete-time CBF used in the MPC scheme with imperfect CBF and MPC models to control a discrete-time nonlinear system as*

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}_k) \tag{5}$$

where $\boldsymbol{d}_k \in \mathbb{R}^{n_d}$ is the process disturbance. This disturbance $\boldsymbol{d}_k \sim \mathcal{N}\left(\bar{\boldsymbol{d}}, \Lambda\right)$ is assumed to be normally distributed with the mean of $\bar{\boldsymbol{d}}$ and variance of Λ. Considering a linear representation of κ as $\alpha h(\boldsymbol{x}_k), 0 < \alpha \leq 1$, the condition (4) *can be extended to the discrete-time case as*

$$\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq -\alpha h(\boldsymbol{x}_k). \tag{6}$$

III. Parameterization of Stochastic CBF-MPC

A. Parameterization of CBF

Although most CBF-based control methods in the literature assume that a complete knowledge of the unsafe regions is available to express a CBF, it is not straightforward to capture a perfect model of CBFs due to model uncertainty and unknown environment, e.g., imperfect model of obstacles. To tackle this issue, we propose to learn the unknown CBFs using an MPC-based reinforcement learning (RL) algorithm. To this end, we first provide a parameterized CBF in which a parameterized class $\mathcal{K}_\infty$ function κ_θ is used. In the simplest case, one can consider a linear form of κ and choose α as a learning parameter such that $\kappa_\theta(h(\boldsymbol{x})) = \theta h(\boldsymbol{x})$. In the present paper, we propose a linear combination of candidate CBFs. Moreover, we use this parametric CBF in the context of MPC, where the mapping function κ is constructed as a polynomial function containing independent odd-powered safety functions, which are candidate class $\mathcal{K}$ functions [14]. Although this parametric CBF can regulate how fast the state of the system can approach the boundary of the safe set $\mathcal{C}$, it has an approximate structure so that one needs to adjust its parameters to achieve the desired performance. We then propose to learn an approximate parametric CBF by RL. Considering an approximate safety function h_a, a parametric version of (6) then reads as

$$\underbrace{h_a(\boldsymbol{x}_{k+1}) - h_a(\boldsymbol{x}_k) + \kappa_{\boldsymbol{\theta}}(h_a(\boldsymbol{x}_k))}_{\mathrm{CBF}^{\boldsymbol{\theta}}(h_a(\boldsymbol{x}_k))} \geq 0, \tag{7a}$$

$$\kappa_{\boldsymbol{\theta}}(h_a(\boldsymbol{x}_k)) = \boldsymbol{H}(\boldsymbol{x}_k)^\top \boldsymbol{\theta}, \qquad 0 < \boldsymbol{\theta} \leq 1 \tag{7b}$$

where

$$\begin{aligned} &\boldsymbol{H}(\boldsymbol{x}_k) = \\ &\left[h_a(\boldsymbol{x}_k), (h_a(\boldsymbol{x}_k))^3, \ldots, (h_a(\boldsymbol{x}_k))^{2p-1}\right]^\top, \quad p \in \mathbb{N}. \end{aligned} \tag{8}$$

Note that the parameterized function $\kappa_{\boldsymbol{\theta}}$ can be perfectly approximated using a rich parameterization, e.g., one can use a Neural Network (NN) under certain conditions. This method will be investigated in the future work.

B. Parameterized MPC

Next, we will formulate a parameterized nominal nonlinear MPC for some constant noises $\bar{\boldsymbol{d}}$ (possibly chosen as $\bar{\boldsymbol{d}} = \mathbb{E}[\boldsymbol{d}_k]$) as

$$\min_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}} \quad \gamma^N T_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_N) + \sum_{k=0}^{N-1} \gamma^k l_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k) \tag{9a}$$

$$\text{s.t.} \quad \bar{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k, \bar{\boldsymbol{d}}\right), \quad \bar{\boldsymbol{x}}_0 = \boldsymbol{x}_0, \tag{9b}$$

$$\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h_a^i(\bar{\boldsymbol{x}}_k)\right) \geq 0, \quad i = 1, \ldots, n_{\mathrm{cbf}} \tag{9c}$$

$$q(\bar{\boldsymbol{u}}_k) \leq 0, \tag{9d}$$

where $0 < \gamma \leq 1$ is a discount factor. The sequences $\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}$ denote the nominal state and control input trajectories, respectively. $T_{\boldsymbol{\theta}}$ and $l_{\boldsymbol{\theta}}$ are the parameterized terminal and stage cost functions, respectively. The control input constraints are introduced by $q\left(\bar{\boldsymbol{u}}_k\right)$. An approximate propagation of the state covariance matrix then reads as [13]

$$\Sigma_{k+1} = A_k \Sigma_k A_k^\top + B_k \Lambda B_k^\top, \quad \Sigma_0 = \hat{\Sigma} \tag{10}$$

where Σ_k is the positive definite covariance matrix. The matrix $\hat{\Sigma}$ denotes the uncertainty of the current state estimation, e.g., $\bar{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}$, and the Jacobian matrices A_k and B_k are obtained as

$$A_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k}, \quad B_k = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}}\right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k}. \tag{11}$$

Taking the uncertainty into account in the CBF-MPC scheme above, we next present a probabilistic CBF.

C. Probabilistic Parameterized CBF

As discussed earlier, one can use a parameterized CBF in (7) to cope with an unknown model of the true h. Additionally, the CBFs are extremely affected by an imperfect model due to model mismatch and disturbances. To address this issue, the notion of state covariance upon the uncertainty propagation can be invoked to provide a robust (stochastic) CBF such that $\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right) \geq 0$ for each CBF, $i = 1, \ldots, n_{\text{cbf}}$, and at each time step $k = 0, \ldots, N$. To tackle the effect of the uncertainty on the CBFs, we then propose a stochastic CBF in the MPC scheme to ensure that the probability of violating each $\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right) \geq 0$ is below a certain level $\epsilon_i \in [0, 1)$. To this end, let us consider a safe parameterized CBF such that the following condition holds

$$\Pr\left(\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right) \geq 0\right) \geq 1 - \epsilon_i. \tag{12}$$

Proposition 1. *Given a user-defined probability level ϵ_i, we consider a back-off coefficient value $\beta_i = \sqrt{\frac{1-\epsilon_i}{\epsilon_i}}$ and choose $C_i = \frac{\partial CBF_i^{\boldsymbol{\theta}}\left(h_a^i\left(\boldsymbol{x}_k\right)\right)}{\partial \boldsymbol{x}}$. The stochastic CBF condition* (12) *is then approximated by a deterministic CBF condition as*

$$CBF_i^{\boldsymbol{\theta}}\left(h_a^i\left(\boldsymbol{x}_k\right)\right) \geq \beta_i \sqrt{C_i \Sigma_k C_i^\top}. \tag{13}$$

Proof. To provide a tractable deterministic version of (12), we employ Cantelli's inequality [15] so that for any scalar $\zeta_i \geq 0$, we have

$$\Pr\left(\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right) \geq \mathbb{E}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] - \zeta_i \middle| \boldsymbol{x}_k\right) \tag{14}$$

$$\geq 1 - \frac{\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right]}{\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] + \zeta_i^2}.$$

To capture the probability level ϵ_i in the stochastic condition $\Pr\left(\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right) \geq 0\right) \geq 1 - \epsilon_i$, we then select

$$\zeta_i = \mathbb{E}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right], \tag{15}$$

and consider the lower bound in the inequality (14) as follows

$$1 - \frac{\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right]}{\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] + \zeta_i^2} \geq 1 - \epsilon_i. \tag{16}$$

Cantelli's inequality then leads to

$$\begin{aligned} \mathbb{E}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] &= \zeta_i \\ &\geq \beta_i \sqrt{\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right]}, \end{aligned} \tag{17}$$

where

$$\mathbb{E}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] = \mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h_a^i\left(\bar{\boldsymbol{x}}_k\right)\right), \tag{18a}$$

$$\mathrm{Var}\left[\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h^i\left(\boldsymbol{x}_k\right)\right)\right] = C_i \Sigma_k C_i^\top, \tag{18b}$$

and the condition (13) then holds. ■

Note that the back-off coefficient value β_i is computed to ensure the probability level in the CBF condition (13). However, this β_i is obtained independently of the underlying probability distribution, which may lead to relatively conservative bounds. To address this issue, one can use an approximation, assuming normally distributed state trajectories, such that the back-off is

$$\beta_i = \sqrt{2}\mathrm{erf}^{-1}\left(1 - 2\epsilon_i\right), \tag{19}$$

where erf^{-1} denotes the inverse error function. The CBF vector C_i is computed as

$$\begin{aligned} C_i = \frac{\partial \mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h_a^i\left(\boldsymbol{x}_k\right)\right)}{\partial \boldsymbol{x}} = \\ \frac{\partial h_a^i}{\partial \boldsymbol{x}}|_{\bar{\boldsymbol{x}}_{k+1}} - \frac{\partial h_a^i}{\partial \boldsymbol{x}}|_{\bar{\boldsymbol{x}}_k} + \frac{\partial \kappa_{\boldsymbol{\theta}}\left(h_a^i\left(\boldsymbol{x}_k\right)\right)}{\partial h_a^i\left(\boldsymbol{x}_k\right)} \frac{\partial h_a^i\left(\boldsymbol{x}_k\right)}{\partial \boldsymbol{x}}|_{\bar{\boldsymbol{x}}_k}. \end{aligned} \tag{20}$$

D. Stochastic CBF-MPC Formulation

In this paper, a parameterized CBF-MPC scheme will be used to deliver a parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ using policy gradient methods. We then formulate an approximate stochastic parameterized CBF-MPC scheme as follows:

$$\min_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \Sigma, \delta^i} \gamma^N T_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_N\right) + \tag{21a}$$

$$\sum_{k=0}^{N-1} \gamma^k l_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k + K_{\boldsymbol{\theta}} \bar{\boldsymbol{x}}_k, \Sigma_k\right) + \left(p^i\right)^\top \delta_k^i$$

$$\text{s.t.} \quad \bar{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k + K_{\boldsymbol{\theta}} \bar{\boldsymbol{x}}_k, \bar{\boldsymbol{d}}\right), \quad \bar{\boldsymbol{x}}_0 = \boldsymbol{x}_0, \tag{21b}$$

$$\Sigma_{k+1} = \tilde{A}_k \Sigma_k \tilde{A}_k^\top + \tilde{B}_k \Lambda \tilde{B}_k^\top, \quad \Sigma_0 = \hat{\Sigma}, \tag{21c}$$

$$\mathrm{CBF}_i^{\boldsymbol{\theta}}\left(h_a^i\left(\bar{\boldsymbol{x}}_k\right)\right) + \delta_k^i \geq \beta_i \sqrt{C_i \Sigma_k C_i^\top}, \tag{21d}$$

$$q\left(\bar{\boldsymbol{u}}_k\right) \leq 0, \tag{21e}$$

$$\delta_k^i \geq 0, \quad i = 1, \ldots, n_{\text{cbf}} \tag{21f}$$

In practice, a slack variable, δ_k^i, is often introduced to ensure constraint feasibility. This relaxation is then penalized in the cost with a large coefficient p^i. The overall control input is in the feedforward/feedback form $\bar{\boldsymbol{u}}_k + K_{\boldsymbol{\theta}} \boldsymbol{x}_k$ due to the prestabilizing controller. Note that we consider a compensation term $\hat{K}_{\boldsymbol{\theta}}$ for the feedback gain such that

$K_{\boldsymbol{\theta}} = \hat{K}_{\boldsymbol{\theta}} + K$. We then let RL adjust the term $\hat{K}_{\boldsymbol{\theta}}$. The Jacobian matrices are computed as

$$\tilde{A}_k = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k + K\bar{\boldsymbol{x}}}, \quad \tilde{B}_k = \left.\frac{\partial f}{\partial \boldsymbol{d}}\right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k + K\bar{\boldsymbol{x}}}. \tag{22}$$

Remark 2. *To achieve a computationally tractable formulation, the stochastic CBF-MPC scheme above uses a feedback control law with a prestabilizing feedback gain K. However, this is an approximate gain, and we then propose to tune this term by adding the compensation term $\hat{K}_\theta$ adjusted by RL. Given the reference steady state and input $\boldsymbol{x}_r, \boldsymbol{u}_r$, we use $\boldsymbol{u}_k = K\boldsymbol{x}_k$ for the linearized system at the steady state, and a quadratic cost of $\boldsymbol{x}_k^\top Q \boldsymbol{x}_k + \boldsymbol{u}_k^\top R \boldsymbol{u}_k$ such that*

$$K = -\left(R + B_r X B_r\right)^{-1} B_r^\top X A_r, \tag{23}$$

where

$$X = A_r^\top X A_r - A_r^\top X B_r \left(R + B_r^\top X B_r\right)^{-1} B_r^\top X A_r + Q, \tag{24a}$$

$$A_r = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_r, \boldsymbol{u}_r}, \quad B_r = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{\boldsymbol{x}_r, \boldsymbol{u}_r}. \tag{24b}$$

Although we use a fixed linearization (K is a time-invariant feedback gain), one can use a time-varying sequence of affine feedback laws, which can be embedded in the MPC scheme. The initial guess for the covariance matrix of the state deviation is labeled by $\hat{\Sigma}$. Although we initialize $\hat{\Sigma}$ at zero in this paper, one can use an observer to estimate this initial matrix. The modified stage cost function is

$$l_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k, \Sigma_k\right) = L_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k\right) + \varphi_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k, \Sigma_k\right), \tag{25}$$

where the local cost term $L_{\boldsymbol{\theta}}$ can be, for instance, a quadratic function and

$$\varphi_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k, \Sigma_k\right) = \operatorname{trace}\left(\frac{\partial^2 L_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k\right)}{\partial \bar{\boldsymbol{x}}_k^2} M \Sigma_k\right) \tag{26}$$

is a cost modification term. More specifically, this term is considered to deliver the impact of the uncertainty on the adopted stage cost. We propose to use a quadratic form for this term, which includes the adjustable matrix M as an RL parameter. Note that $\boldsymbol{f}_{\boldsymbol{\theta}}$ is the nonlinear dynamics, which could be adjusted by RL through the model bias parameters.

IV. Policy Gradient using CBF-MPC

In this paper, we consider the parameterized CBF-MPC scheme (21) as an approximator of the policy, which is used in the context of policy gradient. The parameterized deterministic policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is then delivered by (21). Let us define the closed-loop performance of a parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ for a given stage cost $L\left(\boldsymbol{x}, \boldsymbol{u}\right)$ as the following total expected cost

$$J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k L\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \middle| \boldsymbol{u}_k = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)\right], \tag{27}$$

where the expectation $\mathbb{E}$ is taken over the distribution of the Markov chain in the closed-loop system under policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$. We will focus on Deterministic Policy Gradient (DPG) method that formally maximizes the policy performance based on the deterministic policy gradient theorem. The policy parameters $\boldsymbol{\theta}$ can be directly optimized by the gradient descent method such that the best expected closed-loop cost (a.k.a. policy performance index J) can be achieved by applying the policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\pi}_{\boldsymbol{\theta}}) \tag{28}$$

where $\alpha > 0$ is the learning rate. The policy gradient then reads as

$$\nabla_{\boldsymbol{\theta}} J\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}\right) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k\right) \nabla_{\boldsymbol{u}} Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\Big|_{\boldsymbol{u}_k = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)}\right]. \tag{29}$$

Under some conditions detailed in [16], one can use a *compatible* approximation of the action-value function $Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ in which a class of compatible function approximator $Q^{\boldsymbol{w}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ exists such that the policy gradient is preserved. The compatible state-action function then reads as

$$Q^{\boldsymbol{w}}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \underbrace{\left(\boldsymbol{u}_k - \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k\right)\right)^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^\top\left(\boldsymbol{x}_k\right) \boldsymbol{w}}_{A^{\boldsymbol{w}}} + V^{\boldsymbol{\nu}}\left(\boldsymbol{x}_k\right). \tag{30}$$

The first term in the above compatible function as critic part is an approximation for the advantage function $A^{\boldsymbol{w}} \approx A^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$ and the second is a baseline function approximating the value function $V^{\boldsymbol{\nu}} \approx V^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}$. Both functions can be computed by the linear function approximators as

$$V^{\boldsymbol{\nu}}\left(\boldsymbol{x}_k\right) = \boldsymbol{\Upsilon}\left(\boldsymbol{x}_k\right)^\top \boldsymbol{\nu}, \quad A^{\boldsymbol{w}}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) = \boldsymbol{\Psi}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)^\top \boldsymbol{w}, \tag{31}$$

where $\boldsymbol{\Upsilon}\left(\boldsymbol{x}_k\right)$ is the feature vector, and $\boldsymbol{\Psi}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) := \left(\boldsymbol{u}_k - \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k\right)\right)^\top \nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}^\top\left(\boldsymbol{x}_k\right)$ includes the state-action features. The parameters $\boldsymbol{w}$ and $\boldsymbol{\nu}$ of the action-value function approximation then become the solutions of the following Least Squares (LS) problem

$$\min_{\boldsymbol{w}, \boldsymbol{\nu}} \mathbb{E}\left[\left(Q^{\boldsymbol{\pi}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k) - Q^{\boldsymbol{w}}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right)^2\right]. \tag{32}$$

The problem above can be solved via the Least Square Temporal Difference (LSTD) method, which belongs to *batch methods*, seeking to find the best fitting value function and action-value function, and it is more sample efficient than other methods. To compute the sensitivity $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}$, we first define the Lagrange function $\mathcal{L}_{\boldsymbol{\theta}}$ associated with the CBF-MPC scheme in (21) as follows

$$\mathcal{L}_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right) = \Phi_{\boldsymbol{\theta}} + \boldsymbol{\lambda}^\top G_{\boldsymbol{\theta}} + \boldsymbol{\mu}^\top H_{\boldsymbol{\theta}}, \tag{33}$$

where $\Phi_{\boldsymbol{\theta}}$ is the total parameterized costs of the CBF-MPC scheme. The inequality constraints of (21) are collected by $H_{\boldsymbol{\theta}}$ while $G_{\boldsymbol{\theta}}$ collects the equality constraints. Let $\boldsymbol{\lambda}$ be the Lagrange multipliers associated with the equality constraints $G_{\boldsymbol{\theta}}$. Variables $\boldsymbol{\mu}$ are the Lagrange multipliers associated with the inequality constraints. We then label $\boldsymbol{\Gamma} = \left\{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \Sigma, \delta^i\right\}$

the primal variables. The associated primal-dual variables then read as $\boldsymbol{z} = \{\boldsymbol{\Gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}\}$. The sensitivity of the policy delivered by the MPC scheme (21) w.r.t the policy parameters can be obtained using the Implicit Function Theorem (IFT) on the Karush-Kuhn-Tucker (KKT) conditions underlying the parametric Nonlinear Programming (NLP) such that

$$\frac{\partial \boldsymbol{z}^\star}{\partial \boldsymbol{\theta}} = -\frac{\partial \Omega_{\boldsymbol{\theta}}}{\partial z}^{-1} \frac{\partial \Omega_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}, \quad \Omega_{\boldsymbol{\theta}} = \begin{bmatrix} \nabla_{\boldsymbol{\Gamma}} \mathcal{L}_{\boldsymbol{\theta}} \\ G_{\boldsymbol{\theta}} \\ \text{diag}\left(\boldsymbol{\mu}\right) \boldsymbol{H}_{\boldsymbol{\theta}} \end{bmatrix}, \qquad (34)$$

where $\Omega_{\boldsymbol{\theta}}$ is the KKT matrix associated with the proposed CBF-MPC scheme (21). As $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is a part of $\boldsymbol{z}^\star$, the sensitivity of the CBF-MPC policy $\nabla_{\boldsymbol{\theta}} \boldsymbol{\pi}_{\boldsymbol{\theta}}$ can be extracted from gradients $\frac{\partial \boldsymbol{z}^\star}{\partial \boldsymbol{\theta}}$.

V. Simulation Results and Discussion

To illustrate the proposed RL-based stochastic CBF-MPC design approach, we implement two different scenarios: (1) trajectory tracking; and (2) path planning. In the first test case, a Wheeled Mobile Robot (WMR) is to follow a desired trajectory while avoiding some unknown obstacles, where the WMR is affected by model uncertainties and disturbances. Let us define the WMR model as

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \cos(\psi) & 0 \\ \sin(\psi) & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}, \qquad (35)$$

where $\boldsymbol{x} = [x, y, \psi]^\top$ and $\boldsymbol{u} = [v, \omega]^\top$ are the state and control input vectors, respectively. The position coordinates of the WMR are labeled by x, y, and ψ is the robot orientation angle. The control inputs v and ω are the linear and angular velocities, respectively. To discretize the above continuous model, we use a fourth-order Runge-Kutta (RK4) integrator providing discretized function $\boldsymbol{f}_d$ of the WMR model as

$$\boldsymbol{x}(k+1) = \boldsymbol{f}_d\left(\boldsymbol{x}(k), \boldsymbol{u}(k) + \boldsymbol{\Gamma}_1(k)\right) + \boldsymbol{\Gamma}_2(k), \qquad (36)$$

where $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ model the uncertainties as

$$\boldsymbol{\Gamma}_1(k) = v(k) \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix}, \quad \boldsymbol{\Gamma}_2(k) = T_s v(k) \begin{bmatrix} 0 \\ 0 \\ d_2(k) \end{bmatrix}, \quad (37)$$

where $d_1(k) \sim \mathcal{N}\left(0, \Sigma_1^2\right)$ and $d_2(k) \sim \mathcal{N}\left(0, \Sigma_2^2\right)$. The sampling time is T_s= 0.1 s and disturbance variances Σ_1 and Σ_2 are set as 0.3 and 0.4, respectively. We initialize the adjustable process noise covariance matrix (3×3) as $\Lambda = \text{diag}(\Sigma_1^2, \Sigma_2^2, \Sigma_2^2)$. The expected value of the process noise $\bar{\boldsymbol{d}}$ and the matrix M in (25) as RL parameters are also initialized at zero. The parameters adjusted by RL in the CBF-MPC scheme (21) are $\boldsymbol{\theta} = \left\{l_{\boldsymbol{\theta}}, T_{\boldsymbol{\theta}}, K_{\boldsymbol{\theta}}, \Lambda, \bar{\boldsymbol{d}}, \text{CBF}_i^\theta\right\}$. The safety functions $h^i, i = 1, \ldots, n_{\text{cbf}}$ are defined by circle equation, where the radius of circle is unknown.

A. Trajectory Tracking

First, we consider a trajectory tracking scenario for the mobile robot while it must avoid two unknown static obstacles (the blue ovals as CBFs in the real environment) shown in Figure 1. As it is observed, the stochastic CBF-MPC scheme without learning (the baseline scheme) cannot guarantee a collision-free trajectory tracking even if the probability level is a small value, $\epsilon_i = 0.1$. The initial position of the mobile robot is $\boldsymbol{x}_0 = [0.5, 0, 0]^\top$. To achieve a safe trajectory tracking, we then propose to modify the imperfect CBF-MPC scheme by RL such that the WMR can track the desired trajectory and avoid the unknown obstacles as shown in Figure 2.

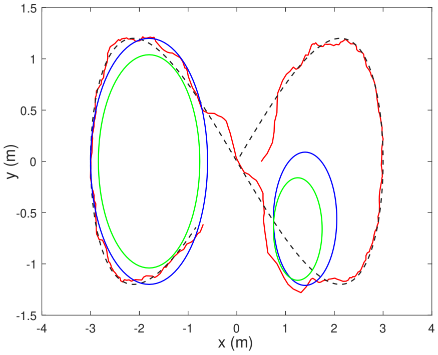


Fig. 1: Baseline CBF-MPC scheme with imperfect CBF. The green ovals are the imperfect CBF models used in the MPC scheme. The blue ovals are the obstacles (correct CBFs) in the real environment. The black dashed line is the desired trajectory.

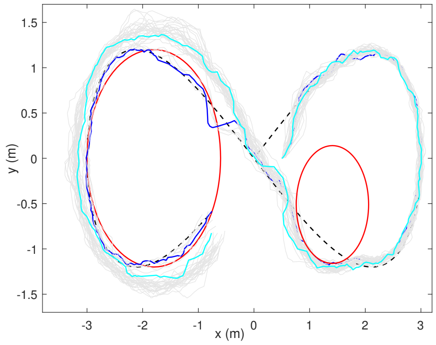
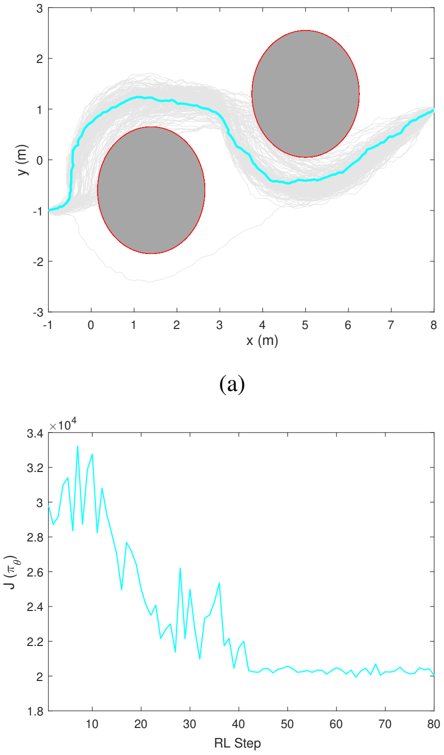


Fig. 2: Proposed stochastic CBF-MPC. The imperfect CBF models (green ovals in Figure 1) are used in the MPC scheme. The red ovals are the obstacles in the real environment. The blue line is the actual path obtained using a stochastic CBF-MPC without learning while the cyan line is generated by the proposed learning-based motion controller (RL-based stochastic CBF-MPC).

B. Path Planning

In this test case, we assume that the obstacles (CBFs) are modeled precisely. We consider a larger probability level $\epsilon_i = 0.3$ and let RL adjust the proposed stochastic CBF-MPC scheme. As shown in Figure 3, we observe that the stochastic CBF-MPC without learning still cannot provide a safe path planning even if we use an exact model of the true CBFs in the MPC scheme. Indeed, the approximate stochastic CBF-MPC uses an approximate chance-constraint CBF, and so this constraint may not be satisfied in the presence of uncertainties and stochastic disturbances. To tackle this issue, we then use the proposed learning-based CBF-MPC scheme such that the path generated after almost 40 RL steps can be regarded as the best path (a collision-free path). More specifically, the best closed-loop performance is achieved after 40 RL steps as there is no more significant improvement in the performance index $J(\boldsymbol{\pi}_{\boldsymbol{\theta}})$ shown in Figure 3. The optimal policies captured by the CBF-MPC after 80 RL steps are shown in Figure 4.



(a)

(b)

Fig. 3: The evolution of trajectories is shown in (a). The cyan line is the path obtained after 80 RL steps. The closed-loop performance is shown in (b).

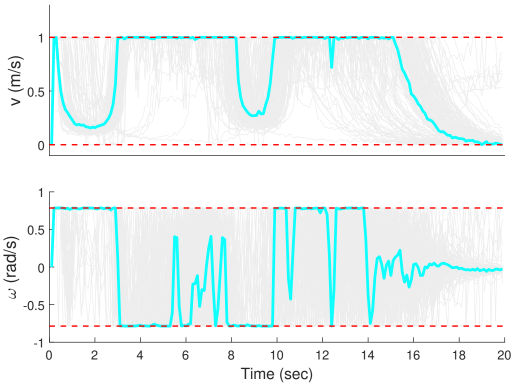


Fig. 4: The evolution of control inputs with the optimal policies shown in cyan.

VI. Conclusion

In this paper, a tractable safety-critical MPC scheme based on the stochastic CBFs has been proposed to cope with unknown systems, where both the CBF and MPC models used in the nominal CBF-MPC scheme cannot perfectly capture the real system due to stochastic disturbances and model uncertainties. In the proposed stochastic CBF-MPC scheme, an approximate chance-constraint CBF is formulated. We then use an MPC-based policy gradient algorithm to learn this approximate CBF-MPC scheme in order to achieve the optimal closed-loop performance. To demonstrate the performance of the learning-based stochastic CBF-MPC, we have used both trajectory tracking and path planning for a mobile robot in the presence of model uncertainties and unknown structure of CBFs.

References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[2] D. Panagou, D. M. Stipanovič, and P. G. Voulgaris, "Multi-objective control for multi-agent systems using Lyapunov-like barrier functions," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 1478–1483.

[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[4] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM review*, vol. 44, no. 4, pp. 525–597, 2002.

[5] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[6] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3882–3889.

[7] N. N. Minh, S. McIlvanna, Y. Sun, Y. Jin, and M. Van, "Safety-critical model predictive control with control barrier function for dynamic obstacle avoidance," *arXiv preprint arXiv:2211.11348*, 2022.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[9] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[10] S. Gros and M. Zanon, "Data-driven economic NMPC using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2020.

[11] H. N. Esfahani and S. Gros, "Policy gradient reinforcement learning for uncertain polytopic LPV systems based on MHE-MPC," *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 1–6, 2022, 6th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2022.

[12] H. N. Esfahani, A. B. Kordabad, W. Cai, and S. Gros, "Learning-based state estimation and control using MHE and MPC schemes with imperfect models," *European Journal of Control*, p. 100880, 2023.

[13] H. Nejatbakhsh Esfahani, A. Bahari Kordabad, and S. Gros, "Approximate robust NMPC using reinforcement learning," in *2021 European Control Conference (ECC)*, 2021, pp. 132–137.

[14] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[15] R. Pelessoni and P. Vicig, "Jensen's and cantelli's inequalities with imprecise previsions," *Fuzzy Sets and Systems*, vol. 458, pp. 50–68, 2023, uncertainty, Games and Decision (220 p.).

[16] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*. JMLR.org, 2014, p. I–387–I–395.