2025 IEEE 19th International Conference on Control & Automation (ICCA)
June 30 - July 3, 2025. Tallinn, Estonia

An Efficient Bayesian Policy Exploration Approach for Reinforcement Learning Model Predictive Control

Yihao Qin, Yiding Ji*

Abstract— **Reinforcement Learning Model Predictive Control (RL-MPC) has achieved significant progress in recent years. However, existing approaches still have some limitations. This paper proposes a Bayesian policy exploration method for RL-MPC that substantially enhances its performance. Specifically, we implement Bayesian posterior estimation of value functions and introduce an optimistic exploration strategy tailored for efficient exploration of RL-MPC, which improves the sample efficiency of RL policy exploration. Then an optimistic Bayesian exploration strategy is proposed, which encourages the agent to leverage existing model information to achieve superior control performance. The soundness and effectiveness of our method are evaluated through an empirical study of controlling a drone to reach targets subject to uncertain model parameters and environmental perturbations. The results validate that our approach has superior performance compared with benchmarks.**

Index Terms— **reinforcement learning, model predictive control, Bayesian estimation, optimistic policy exploration.**

I. Introduction

Model Predictive Control (MPC) has been extensively studied due to its applicability in many engineering applications, providing reliable control performance and ensuring fulfillment of design constraints [1]–[5]. However, it is often not feasible to construct a precise mathematical model for real-word dynamic systems since they are intrinsically non-deterministic, complex and subject to disturbances. Although various robust and stochastic MPC approaches have been developed, they mainly focus on satisfying the constraints, often overlooking the impacts of stochastic uncertainties on the overall performance of the closed-loop system [6]–[10].

Reinforcement learning (RL) is an intelligent decision framework that does not rely on models of the given system to generate action policies. Instead, it utilizes observed state transitions and reward functions derived from interactions with the real system. MPC is particularly well-suited to execute action-value functions and policies in RL, as it naturally incorporates prior knowledge and explicitly handles constraints imposed on the system. Recently, the integration of RL and MPC has attracted substantial attention in both learning and control communities. [11] first applied MPC to approximate value functions in RL and manipulated the MPC scheme holistically with RL methods to maximize closed-loop performance. Notably, the developed approach effectively captures the optimal action-value functions and policies for the actual system, even when no accurate model is available. This framework was extended in [12] by ensuring safety guarantees on the system's closed-loop behaviors. RL-MPC also has convincing performance in various applications, such as traffic networks [13], renewable energy [14], as well as control of autonomous system [15].

Despite the comprehensive investigation of RL-MPC, some open problems remain to be tackled. Specifically, current methods heavily depend on the MPC solutions such as accumulative cost functions at current and future moments of the prediction horizon to approximate the value functions. However, due to model uncertainty and noises, the approximation may deviate from the real values, which causes the agent to explore an excessive number of samples before convergence. This, in turn, reduces the sample efficiency and limits applicability of the framework on many scenarios. To the best of our knowledge, there is currently no established results to facilitate efficient update of the state-action value function estimates based on the collected state costs.

Moreover, conventional RL-MPC methods typically implement the first step of the optimal control sequence calculated by MPC as the current action [16]. However, this contradicts with the core idea of RL exploration, which seeks to discover and implement new policies beyond the existing ones. Although some RL-MPC approaches have attempted to incorporate exploration by sampling actions from a broader space, see, e.g., [17], their exploration strategies still suffer from heavy randomness and fail to effectively leverage the model information, thus generating suboptimal policies.

To mitigate those issues, this paper proposes a Bayesian policy exploration method to enhance the performance of RL-MPC. Based on real-time rewards, we introduce posterior Bayesian estimation techniques to improve the accuracy and efficiency of value function approximation. Additionally, we propose an optimistic exploration strategy for RL-MPC, which calculates the gradient of the control objective with respect to the parameters the current model. This enables the agent to better exploit the updated model parameters for action exploration, ultimately improving control performance.

This paper is organized as follows. Section II reviews the preliminary knowledge of RL and MPC, then formulates the key problem for investigation. Section III analyzes the inefficiency issues of existing RL-MPC policy exploration methods, then develops a Bayesian framework for value

Yihao Qin and Yiding Ji (corresponding author) are with Robotics and Autonomous Systems Thrust, the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. (Email addresses: yqin637@connect.hkust-gz.edu.cn, jiyiding@hkust-gz.edu.cn)

This work is supported by the National Natural Science Foundation of China grants 62303389 and 62373289; Guangdong Basic and Applied Basic Research Funding grants 2022A151511076 and 2024A1515012586; Guangdong Scientific Research Platform and Project Scheme grant 2024KTSCX039; Guangzhou Basic and Applied Basic Research Scheme grant 2023A04J1067; Guangzhou-HKUST(GZ) Joint Funding Program grants 2023A03J0678, 2023A03J0011, 2024A03J0618 and 2024A03J0680.

function estimation and optimistic policy exploration to mitigate the issue. Section IV implements our method to control a drone with uncertain parameter and under disturbances and presents simulation results. Finally, Section V concludes the work and outlines several directions for future extensions.

II. Preliminaries and Problem Formulation

MPC is typically framed as an optimal control problem. At each time step, the controller forecasts future system states over a finite prediction horizon using the system's dynamic model, subsequently determining the initial control action by optimizing the anticipated accumulation of rewards over this horizon. More specifically, this optimal control problem can be solved iteratively in a receding horizon optimization manner, where the objective is to minimize a predefined cost function and simultaneously satisfy the constraints:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{x},\sigma} \quad & \lambda_\theta\left(x_0\right)+\sum_{k=0}^{N-1}\ell_\theta\left(x_k,u_k\right) \\ \text{s.t.} \quad & x_{k+1}=f_\theta\left(x_k,u_k\right), \\ & h_\theta\left(x_k,u_k\right)\le 0, \\ & x_0=s \end{aligned} \tag{1}$$

where λ_θ is the initial state cost serving as a storage function from Theorem 2 in [18]; $\gamma \in [0,1]$ is the discount factor for stage cost; N is the MPC horizon; $x_k \in \Re^m$ is the state trajectory prediction (m is the dimension of the state space); $u_k \in \Re^r$ is the sequence of control inputs; $f_\theta(x_k,u_k)$ represents the system dynamics; $\ell_\theta\left(x_k,u_k\right)$ is the stage cost.

Markov Decision Process (MDP) [19] provides a generic framework for RL problems. A MDP is denoted by a tuple (S,A,R,P) where $S\in\mathbb{R}^{n_s}$ is the state space, $A\in\mathbb{R}^{n_a}$ is the action space, $R:\mathbb{R}^{n_s}\to\mathbb{R}$ is the reward function and $P\left[s_{k+1}\mid s_k,a_k\right]:\mathbb{R}^{n_s}\times\mathbb{R}^{n_s}\times\mathbb{R}^{n_a}\to[0,1]$ is the transition probability, which is often unknown a priori in RL settings.

Instead of designing a control law from the model of the dynamic system, RL generates a parameterized control policy $\pi_\theta(s):\mathbb{R}^{n_s}\to\mathbb{R}^{n_a}$ from the collected data, which aims to maximize the expected discounted accumulative rewards:

$$J\left(\pi_\theta\right):=E_{\tau\sim\pi_\theta}\left[\sum_{k=0}^{\infty}\gamma^k r\left(s_k,a_k\right)\right] \tag{2}$$

where τ is the stochastic trajectory of the closed loop system under policy π_θ, γ represents the discount factor, E represents the expectation operation. In order to effectively optimize the performance objective (2) and find the optimal policy, sensitivity-based RL methods use approximations of the optimal value functions $V_\star(s)$, and optimal action-value function $Q_\star(s,a)$ of the real system, respectively, defined as:

$$V_\star(s_k)=E_{\tau\sim\pi_\star}\left[\sum_{i=k}^{\infty}\gamma^{i-k}r(s_i,a_i)\right] \tag{3a}$$

$$Q_\star(s_k,a_k)=r(s_k,a_k)+\gamma E\left[V_{\pi_\star}(s_{k+1})|s_k,a_k\right] \tag{3b}$$

Based on the above concepts, several algorithms can be designed, such as Q learning [20] and policy gradient [17].

Conventional deep RL methods often apply Deep Neural Networks to approximate the optimal value functions and learning policies. In contrast, MPC based function approximation parameterizes the output of the MPC solution as a value function estimator. Such a scheme was first introduced in [11] and adopted below to fit the settings of this work.

Problem 1 (Optimal value function approximation by MPC)**.** *Given an MDP with unknown transition dynamics, then the goal is to approximate the optimal value function V^* of RL using the following MPC scheme parameterized by θ:*

$$\begin{aligned} V_\theta(s)=\min_{\mathbf{u},\mathbf{x},\sigma} \quad & \lambda_\theta\left(x_0\right)+\sum_{k=0}^{Z-1}\gamma^k\left(\ell_\theta\left(x_k,u_k\right)+\kappa_k^\top\sigma_k\right) \\ & +\gamma^N\left(V_\theta^{\mathrm{f}}\left(x_N\right)+\kappa_n^\top\sigma_N\right) \\ \textit{s.t.} \quad & x_{k+1}=f_\theta\left(x_k,u_k\right), \\ & h_\theta\left(x_k,u_k\right)\le\sigma_k, \\ & h_\theta^{\mathrm{f}}\left(x_N\right)\le\sigma_N, \\ & x_0=s \end{aligned} \tag{4}$$

where $\gamma\in[0,1]$ is the discount factor for stage cost; κ_k and κ_n are the weight on slack variables σ_k and σ_N, respectively; $h_\theta^{\mathrm{f}}\left(x_N\right)$ and $h_\theta^{\mathrm{f}}\left(x_N\right)$ are constraints on states and control input, respectively; $V_\theta^f\left(x_N\right)$ is the approximated value function at the terminal state of the MPC horizon.

In addition, the action-value function is defined as $Q_\theta(s,a)=\min_{u,x}V_\theta(s)$ where we let $u_0=a$ and the other constraints in (4) be satisfied. Accordingly, the optimal value functions and policies are derived under the above scheme:

$$\pi_\theta(s)=\arg\min_a Q_\theta(s,a),\ V_\theta=\min_a Q(s,a) \tag{5}$$

The above scheme establishes a parameterization of the MPC, meanwhile encapsulates the associated costs and constraints characterize the optimal action-value functions and policies of RL. In the following section, we will develop an efficient policy exploration approach to address Problem 1.

III. Bayesian policy exploration in RL-MPC

In this section, we first comprehensively analyze the inefficiency issues of existing RL-MPC exploration methods. Next we propose an efficient Bayesian posterior state-action value function estimation approach based on the observed state rewards from history. Then an optimistic Bayesian policy exploration strategy is proposed, which involves information of the current estimated model to improve sample efficiency.

A. Inefficient exploration of existing RL-MPC methods

Conventional RL-MPC algorithms usually utilize exploration to enhance existing policies by executing new actions that may not align with the current policy. The exploration process allows the agent to improve its performance and a common exploration technique [17] is to add a stochastic perturbation term to the MPC cost function. Specifically, a term $\mathrm{d}^\top u_0$ is added to the MPC cost as follows:

$$\mathrm{a}^{\mathrm{d}}:=\pi_\theta(\mathrm{s})+\mathrm{d} \tag{6}$$

where $\mathrm{d}\in\Re^r$ denotes random perturbations. The cost then results in a perturbed MPC policy π_θ and retains the satisfaction of the constraints. However, random perturbation

may not lead to optimal exploration, and therefore cause a significant degradation of performance and information gain.

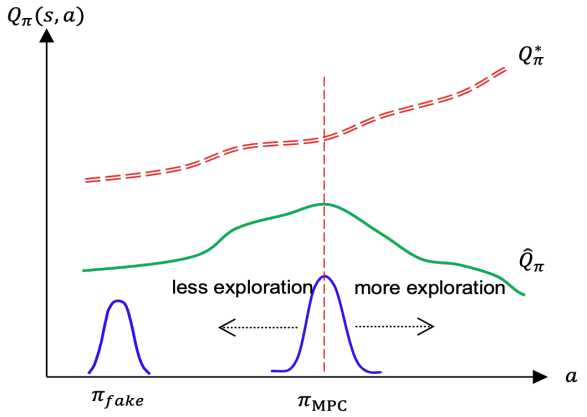

Fig. 1: A scenario of suboptimal exploration in RL-MPC.

For instance, consider the scenario where the state s is fixed, the relationship between the optimal action value function $Q_\pi^\star$, the estimated action value function $\hat{Q}_\pi$, and the action a is illustrated in Fig. 1, and $Q_\pi^\star$ is unknown. The primary RL objective is to select actions that maximize $Q_\pi(s,a)$. However, as shown above, introducing a random perturbation to the actions result in two major drawbacks:

- ***blind exploration***: In Fig. 1, the error between the estimated action value function $\hat{Q}_\pi$ and the optimal action value function Q_π^* varies between the positive and negative directions of π_{MPC}, which indicates the output action of the MPC in the current state. Notably, the estimation error in the positive direction demonstrate reduced magnitude, while their negative counterparts exhibit significantly greater amplitude. To efficiently sample actions that compensate for the discrepancy between $\hat{Q}_\pi$ and Q_π^*, it is essential to increase the exploration in the positive direction. However, most conventional exploration methods typically sample actions uniformly in both directions, which causes inefficient sampling in the negative direction, where $\hat{Q}_\pi$ is sufficiently accurate. The blindness of existing exploration strategies is a key limitation that adversely affects both sample efficiency and training performance.
- ***locally optimal policy:*** As illustrated in Fig. 1, the current estimate $\hat{Q}_\pi$ has a small estimation error near π_{fake}. If the step size of the random exploration is not sufficiently large, the agent will falsely conclude that π_{fake} is an optimal policy. However, the action value function Q_π^* implies that the agent is capable to achieve higher rewards by exploring actions in the positive direction. Consequently, the agent risks converging to a locally optimal policy, which hinders further exploration and refrains from greater rewards.

The above mentioned estimation error extends beyond the mere difference between the values of $\hat{Q}_\pi$ and Q_π^*. In the context of RL-MPC where action value functions are employed to select optimal actions, a similar trend in the changing behaviors of $\hat{Q}_\pi$ and Q_{π^*} will likely yield comparable optimal actions in the output. Furthermore, for the sake of a more generalized analysis, the notations used in the preceding discussion and accompanying schematic diagram may slightly differ from those employed in earlier sections of the article, when no confusion is iccurred.

B. Bayesian Approximation for RL-MPC

As previously discussed, existing RL-MPC methods primarily employs collected data to identify an optimal parameter θ such that $Q_\theta = Q_\star$ and consequently $\pi_\theta = \pi_\star$. This approach relies on approximating the optimal state value function and action value function, as described in equation (3b). However, the intrinsic stochastic nature of real systems often complicates the accurate estimation of these value functions, which in turn hinders the determination of optimal parameters. This challenge also causes approximation errors, which deteriorates the performance of the RL-MPC scheme.

$$\mathcal{E}_Q(s,a) = Q^*(s,a) - Q_\theta(s,a) \tag{7}$$

To effectively capture the uncertainty and efficiently minimize estimation errors, we draw inspiration from data-driven control and learning [21], [22], then propose a novel estimation scheme. The action value functions are approximated following a Gaussian prior distribution based on the MPC output, rather than directly determined by the MPC solution.

$$\hat{Q}(s,a) \sim \mathcal{N}(\mu, k(\cdot,\cdot)) \tag{8}$$

where μ represents the mean value, and $k(\cdot,\cdot)$ represents the covariance between state action pairs, e.g., (s,a) and (s',a').

We further define the system cost $\ell(s_t,a_t)$ at time t based on the temporal difference equation as follows [20].

$$\ell(s_t,a_t) = \hat{Q}(s_t,a_t) - \gamma \min_{a_+} \hat{Q}(s_{t+1},a_+). \tag{9}$$

Note that the expected value function at time $t+1$ is replaced by a single sample in (9), such that the estimator $\hat{Q}$ is inherently stochastic. In order to describe the stochasticity of (9), we introduce the following statistical generative model:

$$\begin{aligned}\ell(s_t,a_t) =& \hat{Q}(s_t,a_t) - \gamma \hat{Q}(s_{t+1},a_+) \\ &+ N((s_t,a_t),(s_{t+1},a_+))\end{aligned} \tag{10}$$

where $a_+ = \arg\min_{a_+} \hat{Q}(s_{t+1},a_+)$, and $N \sim \mathcal{N}(0,\Sigma)$ is an independent noise to measure the discrepancy between the observed stage cost and the temporal difference. For simplicity, the mean is set as 0 and the other values are set similarly. The covariance matrix is chosen as $\Sigma = \sigma\delta(s-s')$, where σ is a positive constant, δ is the Dirac function, and $s[\cdot]$ is a slice indexed for the multi-dimensional state s.

For convenience, one component w_t is defined to replace the state action pair (s_t,a_t), so (10) is further expressed as

$$\ell(w_t) = \hat{Q}(w_t) - \gamma \hat{Q}(w_{t+1}) + N(w_t, w_{t+1}) \tag{11}$$

According to (11), the relation between the observed stage costs $\ell(w_t)$ and the estimated action value function $\hat{Q}(w_t)$ can be regarded as a latent variable model, with $\hat{Q}(w_t)$ being input latent variables and $\ell(w_t)$ being observable output.

The latent variable model in Equation (11) characterizes the evolution of $\hat{Q}(w_t))$, that is, the differences between adjacent time moments. However, it is insufficient to accurately estimate $\hat{Q}(w_t)$. To address the issue, Theorem 1 establishes

an analytical expression of $\hat{Q}(w_t)$ based on history data $\ell(w_t)$, allowing more efficient and accurate exploration.

Theorem 1. *Given the stack vector of observable stage rewards $\mathbf{L}_t = [\ell(w_1), \ell(w_2), \cdots, \ell(w_t)]^{\mathrm{T}} \in \Re^{t\times 1}$ collected by time t, if the temporal difference equation is (10), then the posterior Bayesian estimate of the RL-MPC optimal action-value function satisfies the following Gaussian distribution:*

$$\hat{Q}(w) \mid \mathbf{L}_{t-1} \sim \mathcal{N}(\mu_t, \Sigma_{t-1}) \tag{12}$$

Proof. Given $\mathbf{L}_t$, the stack of $\hat{Q}(w_t)$ and $N(w_t, w_{t+1})$ can be constructed as follows respectively

$$\hat{\mathbf{Q}}_t = [\hat{Q}(w_1), \hat{Q}(w_2), \cdots, \hat{Q}(w_t)]^{\mathrm{T}} \in \Re^{t\times 1} \tag{13}$$

$$\mathbf{N}_t = [N(w_0, w_1), N(w_1, w_2), \cdots, N(w_{t-1}, w_t)]^{\mathrm{T}} \tag{14}$$

Next, an intermediate matrix is introduced and combined with (13), (14) to derive the the generative model (10):

$$\mathbf{H}_t = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ 0 & 1 & -\gamma & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -\gamma \end{bmatrix} \in \Re^{t-1\times t} \tag{15}$$

$$\mathbf{L}_{t-1} = \mathbf{H}_t \hat{\mathbf{Q}}_t + \mathbf{N}_{t-1} \tag{16}$$

The kernel function representing the covariance between the current state w and previous states w_i for $i = 1 \cdots t$ is

$$\mathbf{k}_t(w) = (k(w_1, w), \ldots, k(w_t, w))^{\top} \in \Re^{t\times 1} \tag{17}$$

which yields the stack of kernel functions collected by t:

$$\mathbf{K}_t = [\mathbf{k}_t(w_1), \ldots, \mathbf{k}_t(w_t)] \in \Re^{t\times t} \tag{18}$$

Due to the properties of multidimensional Gaussian variables [23], $\hat{\mathbf{Q}}_t$ and $\hat{\mathbf{N}}_t$ are expressed in the stacked forms:

$$\hat{\mathbf{Q}}_t \sim \mathcal{N}(Q_{\pi_\theta}(\mathbf{w_t}), \mathbf{K}_t) \tag{19}$$

$$\mathbf{N}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_t) \tag{20}$$

where $\mathbf{w}_t = [w_1, \ldots, w_t]$, $\mathbf{\Sigma}_t = diag\{\Sigma_1, \cdots, \Sigma_t\}$.

To derive the the Bayesian posterior estimate of state action value function $\hat{Q}(w)$ from stage loss functions $\mathbf{L_{t-1}}$, their joint Gaussian distribution is first represented as [24]:

$$\begin{bmatrix} \hat{Q}(w) \\ \mathbf{L_{t-1}} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_Q \\ \boldsymbol{\mu}_L \end{bmatrix}, \begin{bmatrix} \mathbf{Cov}^{Q} & \mathbf{Cov}^{QL} \\ \mathbf{Cov}^{LQ} & \mathbf{Cov}^{L} \end{bmatrix} \right) \tag{21}$$

where $\boldsymbol{\mu}_Q = Q_{\pi_\theta}(w)$ represents the prior mean value of current state value function and is the optimal state action value function output by MPC solution; $\boldsymbol{\mu}_L$ represents the mean value of $\mathbf{L}_{t-1}$. The covariance matrices are derived as:

$$\mathbf{Cov}^{Q} = \mathbf{E}(\hat{Q}(w)\hat{Q}(w)^T) = k(w, w) \tag{22}$$

$$\begin{aligned} \mathbf{Cov}^{L} =& \mathbf{E}(\mathbf{L}_{t-1}\mathbf{L}_{t-1}^T) \\ =& \mathbf{E}((\mathbf{H}_t\hat{\mathbf{Q}}_t + \mathbf{N}_{t-1})(\mathbf{H}_t\hat{\mathbf{Q}}_t + \mathbf{N}_{t-1})^T) \\ =& \mathbf{E}(\mathbf{H}_t\hat{\mathbf{Q}}_t\hat{\mathbf{Q}}_t^T\mathbf{H}_t^T + \mathbf{H}_t\hat{\mathbf{Q}}_t\mathbf{N}_{t-1}^T + \\ & \mathbf{N}_{t-1}\hat{\mathbf{Q}}_t^T\mathbf{H}_t^T + \mathbf{N}_{t-1}\mathbf{N}_{t-1}^T) \\ =& \mathbf{E}(\mathbf{H}_t\hat{\mathbf{Q}}_t\hat{\mathbf{Q}}_t^T\mathbf{H}_t^T + \mathbf{N}_{t-1}\mathbf{N}_{t-1}^T) \\ =& \mathbf{H}_t\mathbf{\Sigma}_t\mathbf{H}_t^T + \mathbf{\Sigma}_{t-1} \end{aligned} \tag{23}$$

where Σ_{t-1} is an identity matrix, $\mathbf{N}_{t-1}\hat{\mathbf{Q}}_t^T = \hat{\mathbf{Q}}_t\mathbf{N}_{t-1}^T = \overset{4}{0}$ due to the independence of process noise and action value functions. Also $\hat{\mathbf{Q}}_t$, $\mathbf{N}_t$, $\mathbf{H}_t$, and $\mathbf{L}_{t-1}$ are stacked vectors introduced in (13), (14), (15) and (16), respectively.

$$\begin{aligned} \mathbf{Cov}^{QL} &= \mathbf{E}(\hat{Q}(w)\mathbf{L}_{t-1}^T) \\ &= \mathbf{E}(\hat{Q}(w)\hat{\mathbf{Q}}_t^T\mathbf{H}_t^T + \hat{Q}(w)\mathbf{N}_{t-1}^T) \\ &= \mathbf{k}_t^T(w)\mathbf{H}_t^T \\ \mathbf{Cov}^{LQ} &= \mathbf{H}_t\mathbf{k}_t(w) \end{aligned} \tag{24}$$

Following the conditional distribution of multidimensional Gaussian variables (21), the Bayesian posterior estimate of state action value function conditioned on $\mathbf{L}_{t-1}$ is given as:

$$\begin{aligned} \hat{Q}(w) \mid \mathbf{L}_{t-1} \sim \mathcal{N}(&\boldsymbol{\mu}_Q + \mathbf{Cov}^{QL}\mathbf{Cov}^{L^{-1}}(\mathbf{L}_{t-1} - \boldsymbol{\mu}_L), \\ &\mathbf{Cov}^{Q} - \mathbf{Cov}^{QL}\mathbf{Cov}^{L^{-1}}\mathbf{Cov}^{LQ}) \end{aligned} \tag{25}$$

We plug (22), (23) and (24) into (25), and adopt the stacking form of kernel function shown in (18), then obtain

$$\hat{Q}(w) \mid \mathbf{L}_{t-1} \sim N(\mu, \mathbf{\Sigma}) \tag{26}$$

where the mean variable μ is given as:

$$Q_{\pi_\theta}(w) + \mathbf{k}_t(w)^T\mathbf{H}_t^T(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^T + \Sigma_{t-1})^{-1}(\mathbf{L}_{t-1} - \mu_{\mathbf{L}}) \tag{27}$$

and the variance variable Σ is given as

$$k(w, w) - \mathbf{k}_t^T(w)\mathbf{H}_t^T(\mathbf{H}_t\mathbf{K}_t\mathbf{H}_t^T + \Sigma_{t-1})^{-1}\mathbf{H}_t\mathbf{k}_t(w) \tag{28}$$

This concludes the proof. □

Theorem 1 indicates that the estimation of the optimal action-value function can be dynamically adjusted in real time, based on the stage costs collected throughout the process. In addition to the improved efficiency, our approach also increases the estimation accuracy by continuously refining the value function with the most current information.

C. Optimistic Bayesian policy exploration

Conventional RL-MPC methods often suffer from inefficient policy exploration, which impedes the improvement of their performance. That is to say, some random RL exploration approaches, such as adding random Gaussian noise, may promote the agent's exploration to a certain extent, however, they are usually not sufficiently efficient, thus not suitable for model-based control. Drawing inspiration from [25], we propose an optimistic Bayesian policy exploration method for RL-MPC, which turns out to be more efficient and is validated in the experiments presented later.

Suppose that the original policy follows a normal distribution $\mathcal{N}(\mu, \Sigma)$. Then the main idea of optimistic exploration is to perturb the original policy in a manner that enhances the agent's performance. This involves adjusting the policy to either maximize or minimize the corresponding Q function, depending on the specific objectives of the application scenarios. Meanwhile, it ensures that the perturbation does not cause significant deviation from the original policy. The mathematical formulation of our approach is detailed as:

$$\mu_E, \Sigma_E = \underset{\mathrm{KL}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu_E,\Sigma_E))\leq\delta}{\arg\min} E_{a_t\sim\mathcal{N}(\mu_E,\Sigma_E)}\left[Q(s,a)(w)\right] \tag{29}$$

where $KL(.,.)$ is the Kullback–Leibler divergence to ensure that the distribution of exploration policy does not differ wildly from the original policy. Next we repeat the lemma from [25] that gives an explicit solution of (29).

Lemma 1. *The mean μ_E and covariance Σ_E derived from the minimization of (29) are explicitly of the following forms:*

$$\mu_E = \mu + \frac{\sqrt{2\delta}}{\left\| [\nabla_a Q(s,a)]_{a=\mu} \right\|_\Sigma} \Sigma \left[\nabla_a Q(s,a)\right]_{a=\mu} \tag{30}$$
$$\Sigma_E = \Sigma$$

Lemma 1 calculates the gradient of $Q(s,a)$ with respect to action a, which is essential for optimistic exploration. In contrast, we utilize the rolling horizon MPC scheme to approximate the state-action value functions, which differs from conventional RL methods where $Q(s,a)$ is typically represented through the construction of a actor-critic neural network. To bridge the gap between conventional RL and the RL-MPC so as to analytically derive the exploration policy, we propose an alternative policy exploration approach specifically designed for RL-MPC, which is detailed below.

Our RL-MPC framework utilizes the initial system model to return the MPC solution, after which RL is employed to fine-tune the model parameters, thereby enhancing the overall control performance. Meanwhile, real-time information concerning states and deviations from the target is also collected during the training phase. Then we leverage such information to derive the following policy exploration to approximate the mean value in (30):

$$\mu_E = \mu + \alpha \nabla_a F_\theta(s,a) \mathbf{diff}(s) \tag{31}$$

where F_θ represents the dynamic model of system, α represents the conservative coefficient, **diff** represents the distance between the current state and the target.

By implementing the aforementioned exploration strategy, the agent "optimistically" utilizes the updated model parameters, facilitating a more targeted perturbation for exploration in the RL-MPC procedure. This approach enables the agent to navigate in a direction that enhances overall performance, effectively mitigating the inefficient exploration issues discussed in the previous subsection. Simulations in the next section further validate the sample efficiency of the policy.

Remark 1. *Note that the RL-MPC approach proposed in this paper is developed under the RL-driven EMPC framework in [11]. The stability of the closed-loop system under such a control scheme has been proven through strict dissipativity based analysis in [11]. The stability of our scheme can be shown similarly and is omitted here due to limit of space.*

IV. Experiment Results

To validate the performance of our method, we run simulations for a perturbed drone model with uncertain parameters, which is commonly seen in applications [26], [27]. The drone is modeled as a discrete-time, linear time-invariant (LTI) system subject to disturbances, whose dynamic model is expressed in (32). The objective of the drone is to reach a designated target location, with the RL-MPC reward measured by both the distance to the target and energy consumption. The second LSTD-Q algorithm is selected as the reinforcement learning backbone for training [28].

$$x_{t+1} = Ax_t + Bu_t + C\Psi(x_t) w_t + G + N \tag{32}$$

where the state includes the 3D positions and velocities, along with the roll and pitch attitudes and rates:

$$x = \begin{bmatrix} p_x & p_y & p_z & v_x & v_y & v_z & a_p & a_r & r_p & r_r \end{bmatrix}^\top$$

where A is the state transition matrix, B is the control input matrix and G is the additional external input and $C\Psi(x_t) w_t$ is the wind disturbance, for details of these variables, please refer to [26]. In addition, N is the external disturbance imposed on the position of the system, i.e., p_x, p_y and p_z. The learnable parameters are set as $[\delta, g, K_z]$, where δ is the backoff parameter of constraint ($h_\theta(x_k, u_k) = (1+\delta)h(x_k, u_k)$)), g is the gravitational pull constant, K_z is the vertical thruster coefficient, the setting of learning parameters remains the same as [26] for fair comparisons.

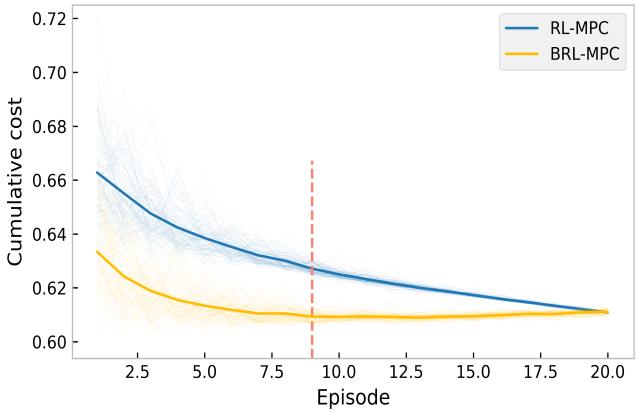

Fig. 2: Performance comparison between RL-MPC and BRL-MPC, where BRL-MPC can converge at around 8th episode.

We ran five experiments, each with different random seeds, and the primary results are presented in Fig. 2, where the solid line represents the mean cost. The Bayesian RL-MPC (BRL-MPC) approach proposed in this work, indicated by the red dotted line, employs Bayesian estimation of the optimal value function and converges at around the 8th episode. This method also demonstrates enhanced resilience to disturbances in position information during the training process, resulting in improved sample efficiency compared to conventional RL-MPC methods. Additionally, Fig. 3 details the learning process of BRL-MPC in terms of four metrics.

We conducted an additional experiment to demonstrate the performance of BRL-MPC with optimistic policy exploration (BRL-MPC-OE) and the results are depicted in Fig. 4. In summary, the cumulative cost of RL-MPC and BRL-MPC is 0.617 and 0.619, respectively, while it is 0.529 for BRL-MPC-OE. The results indicate that optimistic exploration enables the agent to explore more efficiently and results in a lower cumulative cost, which further validates that our method can accomplish the task with superior performance.

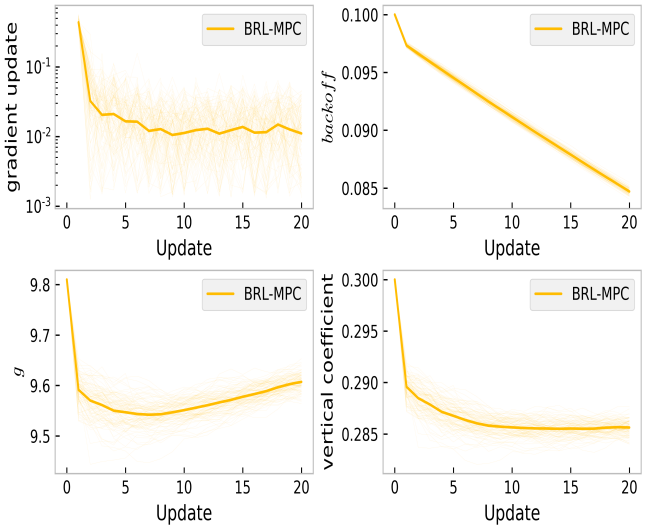

Fig. 3: Diagram of parameters learning in training process.

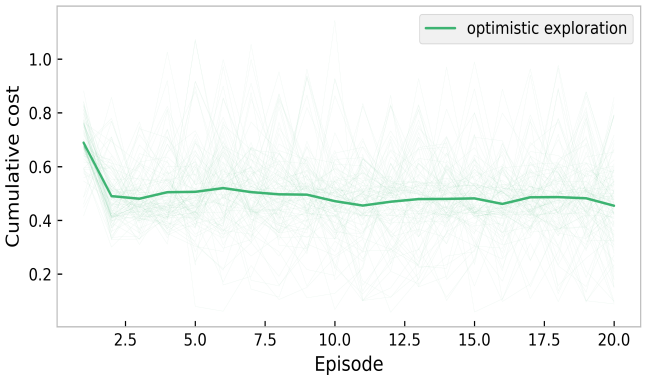

Fig. 4: Performance of BRL-MPC-OE in terms of cumulative cost: convergence at around 3rd episode

V. CONCLUSIONS

This paper introduced a Bayesian policy exploration approach for reinforcement learning model predictive control (RL-MPC). Our method improves the accuracy of value function estimation and sample efficiency through posterior Bayesian estimation. Additionally, we have developed an optimistic Bayesian policy exploration strategy that facilitates efficient policy exploration of agents. A series of simulations were conducted, whose results validate the efficiency and effectiveness of our proposed policy exploration approach. For future research directions, we intend to investigate the stability and adaptiveness of the RL-MPC algorithms.

REFERENCES

[1] J. Chen, A. Behal, Z. Li, and C. Li, "Active battery cell balancing by real-time model predictive control for extending electric vehicle driving range," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 4003–4015, 2024.

[2] D. Yuan, X. Yu, S. Li, and X. Yin, "Safe-by-construction autonomous vehicle overtaking using control barrier functions and model predictive control," *International Journal of Systems Science*, vol. 55, no. 7, pp. 1283–1303, 2024.

[3] K. Zhang, Z. Li, Y. Wang, and N. Li, "Privacy-preserving nonlinear cloud-based model predictive control via affine masking," *Automatica*, vol. 171, p. 111939, 2025.

[4] D. Li, K. Zhang, H. Dong, Q. Wang, Z. Li, and Z. Song, "Physics-augmented data-enabled predictive control for eco-driving of mixed traffic considering diverse human behaviors," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 4, pp. 1479–1486, 2024.

[5] Y. Li, J. Yang, J. Liu, X. Wang, and S. Li, "On stability of model predictive control with finite-control-set constraints and disturbances," *Automatica*, vol. 171, p. 111982, 2025.

[6] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Stochastic mpc with offline uncertainty sampling," *Automatica*, vol. 81, pp. 176–183, 2017.

[7] X. Yu, W. Dong, S. Li, and X. Yin, "Model predictive monitoring of dynamical systems for signal temporal logic specifications," *Automatica*, vol. 160, p. 111445, 2024.

[8] N. Li, K. Zhang, Z. Li, V. Srivastava, and X. Yin, "Cloud-assisted nonlinear model predictive control for finite-duration tasks," *IEEE Trans. on Automatic Control*, vol. 68, no. 9, pp. 5287–5300, 2022.

[9] X. Yin and J. Liu, "Event-triggered state estimation of linear systems using moving horizon estimation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 901–909, 2020.

[10] Y. Ji, X. Yin, and S. Lafortune, "Local mean payoff supervisory control for discrete event systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2282–2297, 2022.

[11] S. Gros and M. Zanon, "Data-driven economic NMPC using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 636–648, 2019.

[12] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2021.

[13] D. Sun, A. Jamshidnejad, and B. De Schutter, "A novel framework combining mpc and deep reinforcement learning with application to freeway traffic control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6756–6769, 2024.

[14] Z. Zhu, G. Dong, Y. Lou, L. Sun, J. Yu, L. Wu, and J. Wei, "MPC-guided deep reinforcement learning for optimal charging of Lithium-Ion battery with uncertainty," *IEEE Trans. on Transport. Elect.*, 2024.

[15] Y. Lu, Z. Li, Y. Zhou, N. Li, and Y. Mo, "MPC-inspired reinforcement learning for verifiable model-free control," in *6th Annual Learning for Dynamics & Control Conference*, pp. 399–413, 2024.

[16] H. N. Esfahani, A. B. Kordabad, and S. Gros, "Reinforcement learning based on MPC/MHE for unmodeled and partially observable dynamics," in *2021 American Control Conference*, pp. 2121–2126, 2021.

[17] S. Gros and M. Zanon, "Bias correction in reinforcement learning via the deterministic policy gradient method for mpc-based policies," in *2021 American Control Conference*, pp. 2543–2548, 2021.

[18] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *IEEE transactions on automatic control*, vol. 57, no. 7, pp. 1615–1626, 2011.

[19] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[20] D. Bertsekas, *A course in reinforcement learning, 2nd Edition*. Athena Scientific, 2025.

[21] W. Zhao, T. He, and C. Liu, "Probabilistic safeguard for reinforcement learning using safety index guided gaussian process models," in *Learning for Dynamics and Control Conference*, pp. 783–796, 2023.

[22] T. Beckers and S. Hirche, "Prediction with approximated gaussian process dynamical models," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6460–6473, 2021.

[23] Y. Qin, Y. Yan, H. Ji, and Y. Wang, "Recursive correlative statistical analysis method with sliding windows for incipient fault detection," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4185–4194, 2021.

[24] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[25] K. Ciosek, Q. Vuong, R. Loftin, and K. Hofmann, "Better exploration with optimistic actor critic," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] F. Airaldi, B. De Schutter, and A. Dabiri, "Learning safety in model-based reinforcement learning using mpc and gaussian processes," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5759–5764, 2023.

[27] K. P. Wabersich and M. N. Zeilinger, "Cautious bayesian MPC: Regret analysis and bounds on the number of unsafe learning episodes," *IEEE Trans. on Automatic Control*, vol. 68, no. 8, pp. 4896–4903, 2022.

[28] H. N. Esfahani, A. B. Kordabad, and S. Gros, "Approximate robust NMPC using reinforcement learning," in *2021 European Control Conference*, pp. 132–137, 2021.