Robust Learning-Based Predictive Control for Discrete-Time Nonlinear Systems With Unknown Dynamics and State Constraints

Xinglong Zhang, Jiahang Liu, Xin Xu, *Senior Member, IEEE*, Shuyou Yu, and Hong Chen, *Senior Member, IEEE*

***Abstract*—Robust model predictive control (MPC) is a well-known control technique for model-based control with constraints and uncertainties. In classic robust tube-based MPC approaches, an open-loop control sequence is computed via periodically solving an online nominal MPC problem, which requires prior model information and frequent access to onboard computational resources. In this article, we propose an efficient robust MPC solution based on receding horizon reinforcement learning, called r-LPC, for unknown nonlinear systems with state constraints and disturbances. The proposed r-LPC utilizes a Koopman operator-based prediction model obtained offline from precollected input–output datasets. Unlike classic tube-based MPC, in each prediction time interval of r-LPC, we use an actor–critic structure to learn a near-optimal feedback control policy rather than a control sequence. The resulting closed-loop control policy can be learned offline and deployed online or learned online in an asynchronous way. In the latter case, online learning can be activated whenever necessary; for instance, the safety constraint is violated with the deployed policy. The closed-loop recursive feasibility, robustness, and asymptotic stability are proven under function approximation errors of the actor–critic networks. Simulation and experimental results on two nonlinear systems with unknown dynamics and disturbances have demonstrated that our approach has better or comparable performance when compared with tube-based MPC and linear quadratic regulator, and outperforms a recently developed actor–critic learning approach.**

***Index Terms*—Model predictive control (MPC), nonlinear systems, reinforcement learning (RL), robustness, state constraints.**

I. Introduction

Model predictive control (MPC) has received considerable attention due to its theoretical developments and wide-spreading applications in industrial plants, robots, etc., (see [1]–[3]). Most of the existing MPC approaches are model based, whose implementations rely on the knowledge of dynamics that are typically identified *a priori*. For control of systems with modeling errors caused by identification and possible exogenous disturbances, robust MPC such as min–max MPC in [4] and [5] or tube-based MPC in [6], [7], and the references therein can be used to ensure robustness and constraint satisfaction. In classic tube-based MPC (also termed as tube MPC) for linear systems, the resulting control is composed of a nominal control action computed by a standard MPC and an offline linear feedback control policy. However, in tube MPC for nonlinear systems (cf. [7]), two nonlinear optimization problems are typically solved, likely with an intensive computational load. Resorting to the recent developments in the machine learning community, in [8]–[10] and the references therein, deep neural networks have been used to approximate an explicit control policy of robust MPC. The control policy is learned offline and deployed online. As a consequence, the computational load can be dramatically reduced; however, new issues, such as insufficient generalization and transfer abilities, might be inherited through deep neural networks.

As a class of methods for solving optimal control problems, reinforcement learning (RL) and adaptive dynamic programming (ADP) have also received significant attention in the past decades (see [11]–[17] and the references therein). Among the classic RL and ADP approaches, infinite-horizon optimal control problems with continuous state space can be solved in a forward-in-time way via actor–critic learning. Along this direction, various notable algorithms have been studied in [18]–[20] for discrete-time nonlinear systems with prior dynamical knowledge, in [21] and [22] for unknown discrete-time nonlinear dynamics, and in [23] for unknown continuous-time nonlinear dynamics. The extensions to learning-based event-triggered control approaches can be found, for instance, in [24] for discrete-time nonlinear systems and in [25] for control of autonomous vehicles under denial-of-service attacks. To solve finite-horizon optimal control problems, a finite-horizon ADP solution was proposed in [26] for nonlinear discrete-time systems with input constraints, and the stability is

Manuscript received October 25, 2021; accepted January 12, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61825305, Grant 62003361, and Grant U21A20518; in part by the China Postdoctoral Science Foundation under Grant 47680; and in part by the National Key Research and Development Program of China under Grant 2018YFB1305105. This article was recommended by Associate Editor Q. Wei. *(Corresponding author: Xin Xu.)*

Xinglong Zhang, Jiahang Liu, and Xin Xu are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: zhangxinglong18@nudt.edu.cn; liujiahang1992@foxmail.com; xuxin_mail@263.net).

Shuyou Yu is with the State Key Laboratory of Automotive Simulation and Control and the Department of Control Science and Engineering, Jilin University, Changchun 130025, China (e-mail: yushuyou@126.com).

Hong Chen is with the College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: chenhong2019@tongji.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2022.3146284.

Digital Object Identifier 10.1109/TSMC.2022.3146284

guaranteed under an open-loop stability assumption. In [27], a finite-horizon near-optimal control algorithm was presented for discrete-time nonlinear affine systems with unknown dynamics by using an identifier-actor–critic structure.

Due to the common roots in optimal control, the relations between MPC and RL and their comparisons have been studied in [28] and [29], and some initial research works on the integration of MPC and RL have emerged recently, which can be called learning-based predictive control (LPC). In [30] and [31], safe robust MPC algorithms were proposed for learning-based control with an event-triggered mechanism. In [3], a safe learning controller was presented based on set-membership recursion. From a different perspective that this article focuses on, RL with an actor–critic structure has been used to solve LPC problems (see, e.g., [32] and [33]). The optimization problem in each prediction interval is solved in a forward-in-time manner via actor–critic learning. In specific, a batch-mode RL-based predictive controller for discrete-time systems with stochastic noises and control constraints was proposed in [32]. In [33], an ADP-based functional MPC was proposed for nonlinear discrete-time systems, where the control saturation is used to deal with the control constraint. The uniform ultimate boundedness of the closed-loop system is proven under certain conditions, relying on a stability assumption similar to [26].

In the framework of LPC, the closed-loop robustness for unstable (stabilizable) perturbed systems under actor–critic approximation errors is a critical issue, which has not been addressed in [26], [27], [32], and [33]. Also, the trial-and-error learning manner of RL could lead to a state constraint violation, which is expensive for safety-critical systems. To the best of our knowledge, no prior LPC with actor–critic network has addressed this point. These issues motivated our work.

In this article, we propose a robust learning-based predictive controller using an actor–critic structure, i.e., r-LPC, for a class of discrete-time nonlinear systems with unknown dynamics, state constraints, and disturbances. This work can be regarded as a new development of previous robust tube-based MPC by using an RL-based strategy to learn closed-loop control policies in an efficient way. The main features of the proposed r-LPC are summarized as follows. First, the proposed approach can learn an explicit closed-loop control policy on the state variable. As a consequence, the control policy can be learned offline and deployed online, or learned online in an asynchronous way whenever necessary, for instance, the safety constraint is violated, such that the online computational load can be reduced. The closed-loop recursive feasibility, robustness, and asymptotic stability of r-LPC are proven under modeling and function approximation errors. Simulation and experimental results on two nonlinear plants with unknown dynamics and disturbances have demonstrated that our approach can obtain better or comparable performance when compared with the previous tube-based MPC [34] and linear quadratic regulator (LQR). In addition, the proposed r-LPC approach also outperforms a recently developed actor–critic learning approach [35] in the adopted tests.

The remainder of this article is organized as follows. Section II introduces the control problem and preliminaries. In Section III, the theoretical property of r-LPC is given. Section IV shows the simulation and experimental results, while some conclusions are drawn in Section V. Some proofs and offline computational details are given in the Appendix.

Notation: We denote $\mathbb{N}_{l_1}^{l_2}$ as the set of integers $l_1, l_1 + 1, \dots, l_2$. Given the variable r, we use $\boldsymbol{r}_{k:k+N-1}$ to denote the sequence $r(k) \dots r(k+N-1)$, where $k, N \in \mathbb{N}_0^{\infty}$. For a vector $x \in \mathbb{R}^n$, we denote $\|x\|_Q^2$ as $x^\top Q x$ and $\|x\|$ as the Euclidean norm (as the Frobenius norm if x is a matrix). For a matrix $A \in \mathbb{R}^{m\times n}$, we denote $\sigma_{\min}(A)$ as the minimal singular value. Given two sets $\mathcal{Z}$ and $\mathcal{V}$, their Minkowski sum is $\mathcal{Z} \oplus \mathcal{V} = \{z + v | z \in \mathcal{Z}, v \in \mathcal{V}\}$, and we denote $\mathrm{Int}(\mathcal{Z})$ as the interior of $\mathcal{Z}$. For variables $z_i \in \mathbb{R}^{q_i}$ and $i \in \mathbb{N}_1^M$, we define $(z_1, z_2, \dots, z_{\mathrm{M}}) = [\, z_1^\top \; z_2^\top \; \cdots \; z_{\mathrm{M}}^\top \,]^\top \in \mathbb{R}^q$, where $q = \sum_{i=1}^M q_i$.

II. Problem Formulation and Preliminaries

A. Control Problem and Preliminary Solution

Consider a class of discrete-time nonlinear systems with additive disturbances described by

$$x(k+1) = f(x(k), u(k)) + w(k) \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u \in \mathcal{U} \subseteq \mathbb{R}^m$ are the state and control variables, $w \in \mathcal{W} \subset \mathbb{R}^n$ is a bounded additive disturbance, $\mathcal{X}$, $\mathcal{U}$, and $\mathcal{W}$ are convex sets containing the origin in their interiors, and mapping f can be unknown. It is assumed that f is C^1 and $f(0,0) = 0, f < +\infty$ in the domain $\mathcal{X} \times \mathcal{U}$, the state x is measurable.

Starting from any $x(0) \in \mathcal{X}$, the control goal of interest is to minimize a cost function $V_\infty(x(0)) = \sum_{k=0}^{+\infty} \|x(k)\|_Q^2 + \|u(k)\|_R^2$ subject to constraints $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$, where $Q = Q^\top \in \mathbb{R}^{n\times n}$ and $R = R^\top \in \mathbb{R}^{m\times m}$, $Q, R \succ 0$.

In the following, we review a recently developed tube-based Koopman MPC in [34] for solving the considered problem.

We first introduce the Koopman model based on the Koopman operator (cf. [34], [36], [37]). To this end, let $\phi(x) : \mathcal{X} \to \mathbb{C}$ be an observable of state x and $\mathcal{F}$ be a given space of observables. The Koopman operator describes dynamics $x(k+1) = f(x(k))$ using a linear dynamic evolution of the observable (cf. [34], [36], [37]), i.e.,

$$\phi(x(k)) = \mathcal{K}\phi(x(k-1)) = \mathcal{K}^k \phi(x(0)) \tag{2}$$

for every $\phi(x) \in \mathcal{F}$.

To apply the Koopman operator for systems with controls, i.e., (1), we define an extended state space $\mathcal{X} \times \ell(\mathcal{U})$, where $\ell(\mathcal{U})$ is the space of $\{u_w(i)\}_{i=0}^{\infty}$ with $u_w(i) = (u(i), w(i)) \in \mathcal{U} \times \mathcal{W}$. Letting $f_W(x, u_w) := f(x, u) + w$ and $\boldsymbol{u}_w(i)$ be the ith element of $\boldsymbol{u}_w := \{u_w(i)\}_{i=0}^{\infty}$, one can write the evolution of the extended state $\boldsymbol{s} = (x, \boldsymbol{u}_w)$ as

$$\boldsymbol{s}(k+1) = F(\boldsymbol{s}(k)) \tag{3}$$

where $F(\boldsymbol{s}) = (f_W(x, \boldsymbol{u}_w(0)), \Gamma \boldsymbol{u}_w)$, Γ is a left shift operator such that $\boldsymbol{u}_w(i+1) = \Gamma \boldsymbol{u}_w(i)$. Hence, one can apply the Koopman operator in (2) to (3), (see also [34], [36], [37]), i.e.,

$$\phi(\boldsymbol{s}(k)) = \mathcal{K}\phi(\boldsymbol{s}(k-1)) = \mathcal{K}^k \phi(\boldsymbol{s}(0)) \tag{4}$$

for every $\phi(\boldsymbol{s}) : \mathcal{X} \times \ell(\mathcal{U} \times \mathcal{W}) \to \mathbb{C}$ belonging in $\mathcal{F}_e$, which is a given space of $\phi(\boldsymbol{s})$.

Note that in (4), the Koopman operator $\mathcal{K}$ is of infinite dimension. For practical concerns of controller design, a finite-dimensional approximation of $\mathcal{K}$ is required. Especially, we choose a collection of observables as $\Phi(\boldsymbol{s}) = (\Psi(x), u_w)$ where $\Psi(x) = (\psi_1(x), \ldots, \psi_{\bar{n}}(x))$, $\bar{n} > n$, and ψ_i, $i \in \mathbb{N}_1^{\bar{n}}$ are constructed as basis functions. Let the approximation of Koopman operator be $\mathcal{K}_{\bar{N}} \in \mathbb{R}^{\bar{N}\times\bar{N}}$ associated with $\Phi(\boldsymbol{s})$, where $\bar{N} = \bar{n} + m + n$. From (4), one writes

$$\Phi(\boldsymbol{s}(k+1)) = \mathcal{K}_{\bar{N}}\Phi(\boldsymbol{s}(k)) + \varepsilon(k) \tag{5}$$

where ε is the error due to the approximation of $\mathcal{K}$.

As shown in [34], there exists an inverse mapping, denoted as Ψ^{-1}, such that $\Psi^{-1}(\Psi(x)) = x$, i.e., the evolution of x can be recovered with (2) through Ψ^{-1}. Let $C\Psi(x)$, $C \in \mathbb{R}^{n\times\bar{n}}$, be an approximation of Ψ^{-1}, such that $x = \Psi^{-1}(\Psi(x)) = C\Psi(x) + v$ (cf. [34]), where the approximation error $v \in \mathcal{V}$, $\mathcal{V}$ is a compact set containing the origin. As the mapping from $\Phi(s)$ to $\Psi(x)$ is of interest, we denote the first $\bar{n}$ rows of $\mathcal{K}_{\bar{N}}$ as $[\mathcal{K}_{\bar{N}}]_{1:\bar{n}} = [A\ B\ D]$, $A \in \mathbb{R}^{\bar{n}\times\bar{n}}$, $B \in \mathbb{R}^{\bar{n}\times m}$, and $D \in \mathbb{R}^{\bar{n}\times n}$. Let $z = \Psi(x)$, in view of (5), then one can write an equivalent form of (1) considering the modeling errors, i.e.,

$$\begin{cases} z(k+1) = Az(k) + Bu(k) + d(k), \ z(k) = \Psi(x(k)) \\ x(k) = Cz(k) + v(k) \end{cases} \tag{6}$$

where $d = Dw + \bar{w} \in \mathcal{D}$ is the exogenous input, $\bar{w} = \varepsilon_{[1:\bar{n}]}$, $\mathcal{D}$ is a compact set containing the origin. The model parameters A, B, C, and D and sets $\mathcal{D}$ and $\mathcal{V}$ are computed in a data-driven way, which are deferred in Appendix A-E.

The following assumptions about model (6) hold.

(A1) The matrix A is stabilizable and full rank.

(A2) The lifted function $\Psi(x)$ satisfies $\Psi(0) = 0$.

(A3) $\Psi(x)$ is Lipschitz continuous for all $x \in \mathcal{X}$.

To control (6), i.e., (1), a robust tube MPC can be used. The overall control law is given as

$$u = \hat{u}^* + Ke_z \tag{7}$$

where $e_z = z - \hat{z}$, matrix $K \in \mathbb{R}^{m\times\bar{n}}$ is such that $F = A + BK$ is Schur stable, and $\hat{u}^*$ is computed by a nominal MPC [deferred in (10)] using the following predictor:

$$\begin{cases} \hat{z}(k+1) = A\hat{z}(k) + B\hat{u}(k) \\ \hat{x}(k) = C\hat{z}(k). \end{cases} \tag{8}$$

By subtracting (6) with (7) and (8), the error e_z evolves in the following unforced system:

$$\begin{cases} e_z(k+1) = Fe_z(k) + d(k) \\ e_x(k) = Ce_z(k) + v(k) \end{cases} \tag{9}$$

where $e_x = x - \hat{x}$. Let the robust invariant set of e_z be $\mathcal{O}_z$, such that $F\mathcal{O}_z \oplus \mathcal{D} \subseteq \mathcal{O}_z$. The corresponding robust "output" invariant set of e_x is defined as $\mathcal{O}_x = C\mathcal{O}_z \oplus \mathcal{V}$.

Inline with [34], at any time k, a nominal MPC is solved online to compute $\hat{u}^*$ in (7), i.e.,

$$\min_{\hat{\boldsymbol{u}}_{k:k+N-1}} V = \sum_{i=0}^{N-1}\left(\left\|\hat{z}(k+i)\right\|_{\bar{Q}}^2 + \left\|\hat{u}(k+i)\right\|_R^2\right) + V_f\big(\hat{z}(k+N)\big) \tag{10}$$

where $\bar{Q} = \bar{Q}^\top \in \mathbb{R}^{\bar{n}\times\bar{n}}$, $\bar{Q} \succ 0$, $V_f(\hat{z}) = \|\hat{z}\|_P^2$, $P = P^\top \in \mathbb{R}^{\bar{n}\times\bar{n}}$, $P \succ 0$ is the solution to

$$F^\top PF - P = -\bar{Q} - K^\top RK. \tag{11}$$

Problem (10) is performed subject to constraints (8), $\hat{z}(k+i) \in \mathcal{Z}$, $\hat{u}(k+i) \in \hat{\mathcal{U}}$, $i \in \mathbb{N}_0^{n-1}$, and $\hat{z}(k+N) \in \mathcal{Z}_f$, where $\mathcal{Z} = \{\hat{z} | C\hat{z} \oplus \mathcal{O}_x \in \mathcal{X}\}$, $\hat{\mathcal{U}} \oplus K\mathcal{O}_z = \mathcal{U}$, $\mathcal{Z}_f$ is a positive invariant set of (8) under constraints $\hat{z} \in \mathcal{Z}$ and $\hat{u} \in \hat{\mathcal{U}}$. Like [34], it is assumed that the sets $\mathcal{Z}$ and $\hat{\mathcal{U}}$ are nonempty and contain the origin in the interior; otherwise, problem (10) is infeasible.

Remark 1: We now summarize the roles of the two control terms in the control law (7) as follows. The first term is computed by solving a nominal MPC [see (10)] to generate the center trajectory of the tube, where tightened constraints $\hat{z} \in \mathcal{Z}$ and $\hat{u} \in \hat{\mathcal{U}}$ are enforced for real constraint fulfillment. The auxiliary feedback term Ke_z is introduced to steer the real state z to $\hat{z}$.

Let $\hat{\boldsymbol{u}}_{k+N-1|k}^*$ be the optimal solution to (10) at time k, then the control

$$u(k) = \hat{u}^*(k|k) + Ke_z(k) \tag{12}$$

is applied at time k, and problem (10) is repeatedly solved according to the receding horizon principle.

B. Definitions About Barrier Functions and Feasible Control

We also introduce a type of barrier functions about constraints to be used in the actor–critic learning algorithm for state constraint satisfaction.

Definition 1 (Barrier Function): For any convex set $\mathcal{Z} = \{z \in \mathbb{R}^n | g_i(z) \le 1 \ \forall i \in \mathbb{N}_1^p\}$, a barrier function is defined as

$$\bar{\mathcal{B}}(z) = \begin{cases} -\sum_{i=1}^p \log(1 - g_i(z)) & z \in \mathrm{Int}(\mathcal{Z}) \\ +\infty & \text{otherwise.} \end{cases}$$

We introduce the relaxed barrier function in the following lemma according to [38] and [39].

Lemma 1 (Relaxed Barrier Function): Let $g_i(z) = a_i^\top z$ where $a_i \in \mathbb{R}^{\bar{n}}$, and define a relaxed barrier function as

$$\mathcal{B}(z) = \begin{cases} \bar{\mathcal{B}}(z) & \bar{\sigma} \ge \kappa \\ \gamma_b(z, \bar{\sigma}) & \bar{\sigma} < \kappa \end{cases} \tag{13}$$

where $\kappa > 0$ is a relaxing factor, $\bar{\sigma} = \min_{i\in\mathbb{N}_{i=1}^p} 1 - a_i^\top z$, the function $\gamma_b(z, \bar{\sigma})$:$(-\infty, \kappa)$ is strictly monotone and differentiable, and $\bigtriangledown^2\gamma_b(z, \bar{\sigma}) \le \bigtriangledown^2\mathcal{B}(z)|_{\bar{\sigma}=\kappa}$, then there exists a positive-definite matrix $H_z \ge (1/2)\bigtriangledown^2\mathcal{B}(z)|_{\bar{\sigma}=\kappa}$, such that $\mathcal{B}(z) \le z^\top H_z z \le \mathcal{B}_{\max}(z)$, where $\mathcal{B}_{\max}(z) = \max_{z\in\mathcal{Z}} z^\top H_z z$.

Proof: In view of Definition 1, for $\bar{\sigma} \ge \kappa$, one has $\bigtriangledown^2\gamma_b(z, \bar{\sigma}) \le \bigtriangledown^2\mathcal{B}(z) \le 2H_z$. As $\mathcal{B}(0) = 0$, it holds that $\mathcal{B}(z) \le z^\top H_z z$ is verified. ■

Definition 2 (Feasible Control Policy): The control policy $\hat{\boldsymbol{u}}_{k:k+N-1}$ is feasible in the prediction interval $[k, k+N-1]$ if $\hat{u}(\tau|k) \in \hat{\mathcal{U}}$, $\hat{z}(\tau|k) \in \mathcal{Z}$, and $\hat{z}(k+N|k) \in \mathcal{Z}_f \ \forall\tau \in \mathbb{N}_0^{N-1}$.

III. Design of Robust MPC Based on RL

In this section, we first give the control structure of the proposed robust MPC based on RL, i.e., r-LPC. Then, we present by order the finite-horizon RL algorithm and the actor–critic implementation, for learning an explicit nominal

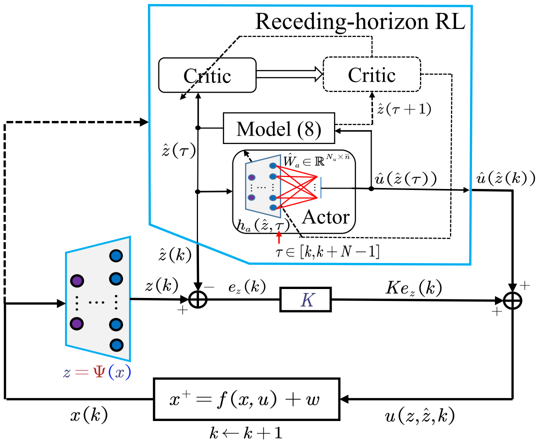
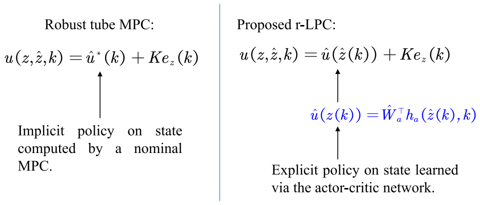


Fig. 1. Control structure of the proposed r-LPC algorithm.


Robust tube MPC:

$u(z,\hat{z},k) = \hat{u}^*(k) + Ke_z(k)$

Implicit policy on state computed by a nominal MPC.

Proposed r-LPC:

$u(z,\hat{z},k) = \hat{u}(\hat{z}(k)) + Ke_z(k)$

$\hat{u}(z(k)) = \hat{W}_a^\top h_a(\hat{z}(k),k)$

Explicit policy on state learned via the actor-critic network.

Fig. 2. Difference between r-LPC and classic tube MPC. In classic robust tube MPC (on the left-hand side), the controller is composed of a state feedback term and an implicit control policy $\hat{u}^*$ via numerically solving an online nominal MPC. Whilst, in the proposed r-LPC approach, the term $\hat{u}(\hat{z})$ is an explicit policy on the state, learned via an actor-critic network.

control policy of r-LPC in each prediction interval. Finally, the pseudocode of r-LPC is given at the end of this section.

A. Control Structure of r-LPC

In the proposed r-LPC approach, the control policy at any time k is of type

$$u(z, \hat{z}, k) = \hat{u}(\hat{z}(k)) + Ke_z(k) \tag{14}$$

where $\hat{u}(\hat{z})$ is an explicit network-based control policy on the state $\hat{z}$ learned by a receding-horizon RL algorithm using an actor–critic structure, while the second term is an offline nonlinear state-feedback policy (see Fig. 1).

Peculiarly, in each prediction interval $[k, k+N-1]$ of r-LPC, the actor and critic adopted in the finite-horizon RL (deferred in Section III-C), are designed with neural networks for approximating the near-optimal control policy and value function. The learned actor network results in a nominal control policy, i.e., $\hat{u}(\hat{z}(k+i|k))$, $i = 0, \ldots, N-1$. Then, the resulting control to be applied at time k is

$$u\big(z, \hat{z}, k\big) = \hat{u}\big(\hat{z}(k|k)\big) + Ke_z(k). \tag{15}$$

At the next time $k+1$, the control policy $\hat{u}(\hat{z})$ can be directly applied or improved via repeatedly solving the finite-horizon learning problem. Also, the difference of the proposed r-LPC with classic robust tube-based MPC is highlighted in Fig. 2.

In the following two sections, we focus on presenting the finite-horizon constrained RL with an actor–critic structure for learning the nominal control policy, i.e., $\hat{u}(\hat{z})$ in the prediction interval. First, the constrained Hamilton–Jacobi–Bellman (HJB) equation is formulated and a finite-horizon iterative RL is given to solve the HJB equation. Then, the actor–critic structure is used to implement the finite-horizon RL, which results in a network-based nominal control policy.

B. Constrained Finite-Horizon RL in the Prediction Interval

Note that it is nontrivial to deal with state constraints in the learning process. Hence, we first transform the hard state constraint $\hat{z} \in \mathcal{Z}$ and control constraint $\hat{u} \in \hat{\mathcal{U}}$ into soft ones with barrier functions defined in Section II-B. Specifically, in line with [38] and [40], the value function in (10) is reconstructed using barrier functions on the states and controls, i.e.,

$$\bar{V}\big(\hat{z}(k)\big) = \sum_{i=0}^{N-1} r\big(\hat{z}(k+i), \hat{u}(k+i)\big) + \bar{V}_f\big(\hat{z}(k+N)\big) \tag{16}$$

where $r(\hat{z}(\tau), \hat{u}(\tau)) = \|\hat{z}(\tau)\|_Q^2 + \|\hat{u}(\tau)\|_R^2 + \mu\mathcal{B}(\hat{z}(\tau)) + \mu\mathcal{B}(\hat{u}(\tau))$, $\bar{V}_f(\hat{z}(k+N)) = V_f(\hat{z}(k+N)) + \mu\mathcal{B}_f(\hat{z}(k+N))$, $\mu > 0$ is a weighting scalar that determines the influence of barrier function values on $\bar{V}(\hat{z}(k))$, and barrier functions $\mathcal{B}(\hat{z})$, $\mathcal{B}(\hat{u})$, and $\mathcal{B}_f(\hat{z})$ are designed according to Definition 1 and Lemma 1. In the case that $\mathcal{Z}_f$ is of an ellipsoidal form, i.e., $\mathcal{Z}_f = \{z \in \mathbb{R}^{\bar{n}} | g(z) \leq 1\}$ and $g(z)$ is a quadratic function, we set $\mathcal{B}_f(\hat{z}) = \bar{\mathcal{B}}(\hat{z})$.

In view of (16), the Lyapunov equation in (11) is modified as

$$F^\top PF - P = -\bar{Q} - K^\top RK - \mu H \tag{17}$$

where $H = H_{\hat{z}} + K^\top H_{\hat{u}} K$, $H_{\hat{z}}$, and $H_{\hat{u}}$ are computed according to Lemma 1 for constraints $\mathcal{Z}$ and $\hat{\mathcal{U}}$, respectively.

With the reconstructed barrier function-based cost function (16), the original constrained optimization problem (10) is transformed as an equality constrained one, which can be solved via the RL framework. To this end, at any time k, letting the remaining cost function at the prediction time $\tau \in [k, k+N-1]$ be $\bar{V}(\hat{z}(\tau)) = \sum_{i=0}^{N-\tau-1} r(\hat{z}(\tau+i), \hat{u}(\tau+i)) + \bar{V}_f(\hat{z}(k+N))$, one can write

$$\bar{V}\big(\hat{z}(\tau)\big) = r\big(\hat{z}(\tau), \hat{u}(\tau)\big) + \bar{V}\big(\hat{z}(\tau+1)\big) \tag{18}$$

where $\bar{V}(\hat{z}(k+N)) = \bar{V}_f(\hat{z}(k+N))$.

Let $\bar{V}^*(\hat{z}(\tau))$ be the optimal value function at $\tau \in [k, k+N-1]$. Then, the discrete-time HJB equation is given as

$$\bar{V}^*\big(\hat{z}(\tau)\big) = \min_{\hat{u}(\tau)} r\big(\hat{z}(\tau), \hat{u}(\tau)\big) + \bar{V}^*\big(\hat{z}(\tau+1)\big) \tag{19}$$

leading to the optimal control policy

$$\hat{u}^*(\hat{z}(\tau)) = \operatorname*{argmin}_{\hat{u}(\tau)} r\big(\hat{z}(\tau), \hat{u}(\tau)\big) + \bar{V}^*\big(\hat{z}(\tau+1)\big). \tag{20}$$

Let $\lambda(\hat{z}(\tau)) = \partial\bar{V}(\hat{z}(\tau))/\partial\hat{z}(\tau)$ be the costate and denote $\lambda_f(\hat{z}(k+N)) = \lambda(\hat{z}(k+N))$ as the terminal costate. Substituting $\hat{u}$ with $\hat{u}^*$ into (19) and in view of the optimality condition $\partial\bar{V}^*(\hat{z}(\tau))/\partial\hat{u}^*(\hat{z}(\tau)) = 0$, one has

$$\mu\frac{\partial\mathcal{B}\big(\hat{u}^*\big(\hat{z}(\tau)\big)\big)}{\partial\hat{u}^*\big(\hat{z}(\tau)\big)} + 2R\hat{u}^*\big(\hat{z}(\tau)\big) + B^\top\lambda^*\big(\hat{z}(\tau+1)\big) = 0. \tag{21}$$

From (19), one has

$$\lambda^*\big(\hat{z}(\tau)\big) = \mu\frac{\partial\mathcal{B}\big(\hat{z}(\tau)\big)}{\partial\hat{z}(\tau)} + 2\bar{Q}\hat{z}(\tau) + A^\top\lambda^*\big(\hat{z}(\tau+1)\big). \tag{22}$$

The computation of $\hat{u}^*$ with (19) and (20) results in a nonlinear problem since $\mathcal{B}(\cdot)$ is nonlinear. To solve this problem, the following finite-horizon iterative RL (i.e., value iteration) is introduced. Given an initial choice $\lambda^0(\hat{z}(\tau)) = 0$, the following two main steps are repetitively performed for all $\tau = k, \ldots, k+N-1$.

1) *Policy Improvement:* Provided $\lambda^i(\hat{z}(\tau+1))$

$$\hat{u}^{i+1}\big(\hat{z}(\tau)\big) = \arg_{\hat{u}(\tau)}\left\{\mu\frac{\partial\mathcal{B}\big(\hat{u}(\tau)\big)}{\partial\hat{u}(\tau)} + 2R\hat{u}(\tau) + B^\top\lambda^i\big(\hat{z}(\tau+1)\big) = 0\right\}. \tag{23}$$

2) *Value Function Update:* Given $\hat{u}^i(\hat{z}(\tau))$

$$\lambda^{i+1}\big(\hat{z}(\tau)\big) = \mu\frac{\partial\mathcal{B}\big(\hat{z}(\tau)\big)}{\partial\hat{z}(\tau)} + 2\bar{Q}\hat{z}(\tau) + A^\top\lambda^i\big(\hat{z}(\tau+1)\big). \tag{24}$$

Remark 2: The convergence proof of $\hat{u}^i(\tau)$ and $\lambda^i(\tau)$ $\forall\tau \in [k, k+N-1]$ to the optimal values under value iteration steps (23) and (24) can be easily verified as $i \to +\infty$, following the results in [32] and [41].

C. Learning Nominal Control Policy With Actor-Critic Structure

In the prediction time $\tau \in [k, k+N-1]$, instead of directly solving steps (23) and (24), a regularized actor–critic structure is used where the critic network is to approximate the optimal costate $\lambda^*(\hat{z}(\tau))$ and the actor network is in charge of learning the optimal control policy $\hat{u}^*(\hat{z}(\tau))$. The resulting nominal control policy is nonlinear, network based, and explicit, which is different from that in classic tube MPC.

We first design the critic network. To this end, let $\lambda^*(\hat{z}(\tau))$ be represented as

$$\lambda^*(\hat{z}(\tau)) = W_c^\top h_c(\hat{z}(\tau), \tau) + \bar{\epsilon}_c(\tau)$$

where $W_c \in \mathbb{R}^{N_c\times\bar{n}}$ is the weighting matrix, $h_c \in \mathbb{R}^{N_c}$ is a vector of basis functions, and $\bar{\epsilon}_c(\tau)$ is the network residual. Let the adopted critic network be defined as

$$\hat{\lambda}(\hat{z}(\tau)) = \hat{W}_c^\top h_c(\hat{z}(\tau), \tau) \tag{25}$$

for all $\tau \in [k, k+N]$, where $\hat{\lambda}(\hat{z}(k+N))$ is the approximated terminal costate, i.e., $\hat{\lambda}(\hat{z}(k+N)) = \hat{\lambda}_f(\hat{z}(k+N))$, $\hat{W}_c$ is the approximation of W_c via minimizing the distance of the optimal costate λ^* and $\hat{\lambda}$. Note however λ^* is not available, in line with [41] and (24), we define a target to be steered by $\hat{\lambda}$, i.e.,

$$\lambda_d(\hat{z}(\tau)) = \begin{cases} \mu\frac{\partial\mathcal{B}(\hat{z}(\tau))}{\partial\hat{z}(\tau)} + 2\bar{Q}\hat{z}(\tau) + A^\top\hat{\lambda}(\hat{z}(\tau+1)), & \tau \in [k, k+N) \\ \mu\frac{\partial\mathcal{B}_f(\hat{z}(\tau))}{\partial\hat{z}(\tau)} + 2P\hat{z}(\tau) := \lambda_{d,f}(\hat{z}(\tau)), & \tau = k+N \end{cases} \tag{26}$$

where $\hat{z}(k+N)$ can be randomly chosen from set $\mathcal{Z}_f$.

Remark 3: In the learning process, with the goal of steering λ_d, $\hat{\lambda}$ is recursively updated along with $\hat{W}_c$ based on the gradient descend mechanism, which however might lead to the state constraint nonsatisfaction since $\hat{W}_c$ is related to $\hat{\lambda}$ (rather than $\hat{z}$). To improve the learning efficiency, one can instead enforce the restriction on $\hat{\lambda}$. To proceed, let at time $\tau = k+j$, $\bar{\mathcal{Z}}^j$ be the feasible set of problem (10), which can be computed according to [42] since Assumption (A1) holds. With $\bar{\mathcal{Z}}^j$, one can compute an estimate of the set where $\hat{\lambda}(\tau)$ lies in, i.e., $\Lambda^j = 2\bar{Q}\bar{\mathcal{Z}}^j + A^\top\Lambda^{j+1}$, starting from $\Lambda^N = 2P\Lambda^N$, since $\partial\mathcal{B}(\hat{z})/\partial\hat{z}$ is close to 0 in view of (26).

Once Λ^j $\forall j \in \mathbb{N}_1^N$ are available, letting $\epsilon_c(\tau) = \lambda_d(\hat{z}(\tau)) - \hat{\lambda}(\hat{z}(\tau))$ $\forall\tau \in [k, k+N-1]$ and $\epsilon_{c,N} = \lambda_d(\hat{z}(k+N)) - \hat{\lambda}(\hat{z}(k+N))$, it is possible to define a regularized optimization cost of the critic network as

$$\delta_c(\tau) = \big\|\big(\epsilon_c(\tau), \epsilon_{c,N}\big)\big\|^2 + \bar{\mu}\Big(\mathcal{B}\Big(\hat{\lambda}\big(\hat{z}(\tau)\big)\Big) + \mathcal{B}_f\Big(\hat{\lambda}\big(\hat{z}(k+N)\big)\Big)\Big) \tag{27}$$

where $\mathcal{B}(\cdot)$ and $\mathcal{B}_f(\cdot)$ are regularization terms on $\hat{W}_c$, defined as barrier functions of constraints $\hat{\lambda}(\hat{z}(k+j)) \in \Lambda^j$ and $\hat{\lambda}(\hat{z}(k+N)) \in \Lambda^N$. At any time instant τ, the weight $\hat{W}_c$ is updated via minimizing (27) according to the following rule:

$$\hat{W}_c(\tau+1) = \hat{W}_c(\tau) - \gamma_c\frac{\partial\delta_c(\tau)}{\partial\hat{W}_c(\tau)} \tag{28}$$

where γ_c is the learning rate. Note that the term $\partial\delta_c(\tau)/\partial\hat{W}_c(\tau) = \partial\delta_c(\tau)/\partial\hat{\lambda}(\tau)\cdot\partial\hat{\lambda}(\tau)/\partial\hat{W}_c(\tau)$ can be computed according to the chain rule since $\epsilon_c(\tau)$ and $\epsilon_{c,N}$ are related to $\hat{\lambda}$, and $\hat{\lambda}$ is a function on $\hat{W}_c$ [see (25)]. Likewise, $\hat{u}^*(\hat{z}(\tau))$ can be represented using a network

$$\hat{u}^*\big(\hat{z}(\tau)\big) = W_a^\top h_a\big(\hat{z}(\tau), \tau\big) + \bar{\epsilon}_a(\tau)$$

where $W_a \in \mathbb{R}^{N_u\times m}$ is the weighting matrix, $h_a \in \mathbb{R}^{N_u}$ is a vector of basis functions, and $\bar{\epsilon}_a(\tau)$ is the network residual. To define the actor network, in view of (21), we define a desired control action $\hat{u}_d(\hat{z}(\tau))$, $\tau \in [k, k+N-1]$ satisfying

$$\begin{aligned} G\big(\hat{u}_d\big(\hat{z}(\tau)\big)\big) &:= \mu\frac{\partial\mathcal{B}\big(\hat{u}_d\big(\hat{z}(\tau)\big)\big)}{\partial\hat{u}_d\big(\hat{z}(\tau)\big)} + 2R\hat{u}_d\big(\hat{z}(\tau)\big) \\ &= -B^\top\hat{\lambda}\big(\hat{z}(\tau+1)\big). \end{aligned} \tag{29}$$

The approximation of $\hat{u}_d(\hat{z}(\tau))$ is then defined as

$$\hat{u}(\hat{z}(\tau)) = \hat{W}_a^\top h_a(\hat{z}(\tau), \tau), \tau \in [k, k+N-1] \tag{30}$$

where $\hat{W}_a$ is an approximation of W_a.

Note that the left-hand side of (29) is composed of a linear mapping and a nonlinear function on $\hat{u}_d$. Letting $\epsilon_a(\tau) = G(\hat{u}_d(\hat{z}(\tau))) - G(\hat{u}(\hat{z}(\tau)))$, $\|\epsilon_a(\tau)\|^2$ is to be minimized in the learning process, which is equivalent to driving $\mu\partial\mathcal{B}(\hat{u}(\hat{z}(\tau)))/\partial\hat{u}(\hat{z}(\tau)) + 2R\hat{u}(\hat{z}(\tau)) + B^\top\hat{\lambda}(\hat{z}(\tau+1)) \to 0$ [see (21)] in the Euclidean norm sense. To also enforce a regularization on $\hat{u}$, we choose to penalize ϵ_a via a regularized loss function as follows:

$$\delta_a(\tau) = \|\epsilon_a(\tau)\|^2 + \bar{\mu}\mathcal{B}\big(\hat{u}\big(\hat{z}(\tau)\big)\big) \tag{31}$$

where the second term imposes a regularization on $\hat{W}_a$ with barrier functions. At any time instant $\tau \in [k, k+N-1]$, $\hat{W}_a$

Algorithm 1 Pseudocode of r-LPC

```
Off-line designs:
1: choose $\Psi(x)$ such that Assumptions (A2) and (A3) are fulfilled;
2: calculate $A$, $B$, $C$, and $D$ with (64) (see Appendix A-E), and check that Assumption (A1) is satisfied;
3: compute $\mathcal{D}$ and $\mathcal{V}$ according to [34];
4: calculate $P$ and $K$ with (17) and compute $\mathcal{O}_z$, $\mathcal{O}_x$ with $K$;
5: compute $\mathcal{Z}$, $\hat{\mathcal{U}}$, and $\mathcal{Z}_f$ with $K$.
..................................................
Online procedures:
Input: Set the iteration threshold $\bar{i}$, initialize $\gamma_a$, $\gamma_c$, select $\hat{W}_a(\tau)$ and $\hat{W}_c(\tau)$ such that $\delta_a < \mathcal{B}_{\max}(\hat{u})$, and $\delta_c < \mathcal{B}_{\max}(\hat{\lambda})$, $i = 0$.
1: At any time steps $k = 1, 2, \cdots$ do
2: .....%%learning process in the prediction interval.....
3: repeat %%iteration loop in the prediction interval
4:     for $\tau = k, \cdots, k+N-1$ do %%finite-horizon forward-in-time learning
5:         compute $\hat{u}(\hat{z}(\tau))$, $\hat{z}(\tau+1)$ with (30), (8);
6:         compute $\hat{\lambda}(\hat{z}(\tau+1))$, $\lambda_d(\hat{z}(\tau))$ with (25), (26)
7:         compute $G(\hat{u}_d(\hat{z}(\tau)))$ with (29);
8:         generate $\hat{\lambda}(\hat{z}(k+N))$, $\lambda_d(\hat{z}(k+N))$ with (25), (26) using randomly selected terminal state $\hat{z}(k+N) \in \mathcal{Z}_f$;
9:         compute $\epsilon_c(\tau), \epsilon_{c,N}$ and update $\hat{W}_c(\tau+1)$ with (27), (28);
10:        compute $\epsilon_a(\tau)$ and update $\hat{W}_a(\tau+1)$ with (31), (32);
11:        if $\hat{z}(\tau) \notin \mathcal{Z} \vee \hat{u}(\tau) \notin \hat{\mathcal{U}} \vee \hat{z}(k+N) \notin \mathcal{Z}_f$ then
12:            Repeat steps 3 to 16 using re-initialized weight (38) and break;
13:        end if
14:    end for
15:    $i \leftarrow i+1$
16:    until $i = \bar{i}$
17: ......%%apply the learned control to the system......
18:    if $\bar{V}(k|k) \le \bar{V}(k|k-1)$ then
19:        generate $\hat{u}(\hat{z}(k|k))$ with (30);
20:    else
21:        compute $\hat{u}(\hat{z}(k|k))$ with (36);
22:    end if
23:    apply $u(z, \hat{z}, k) = \hat{u}(\hat{z}(k|k)) + Ke_z(k)$ to system (1)
24:    update $\hat{z}(k+1)$, $z(k+1)$, $\hat{x}(k+1)$, and $x(k+1)$;
25:    $k \leftarrow k+1$;
```

is updated via minimizing $\delta_a(\tau)$ under the following rule:

$$\hat{W}_a(\tau+1) = \hat{W}_a(\tau) - \gamma_a \frac{\partial \delta_a(\tau)}{\partial \hat{W}_a(\tau)} \tag{32}$$

where γ_a is the learning rate. Likewise, the term $\partial\delta_a(\tau)/\partial\hat{W}_a(\tau) = \partial\delta_a(\tau)/\partial\hat{u}(\tau) \cdot \partial\hat{u}(\tau)/\partial\hat{W}_a(\tau)$ can be computed according to the chain rule since ϵ_a is related to $\hat{u}$ and $\hat{u}$ is a function on $\hat{W}_a$ [see (30)].

In summary, the pseudocode of the proposed r-LPC approach is given in Algorithm 1.

IV. Main Theoretical Results

In this section, we first prove the convergence of the barrier-function regularized actor–critic learning algorithm under (28) and (32) in each prediction interval, which has not been proven by existing works, e.g., [26], [27], [32], and [33]. Then, we give a necessary condition to guarantee the recursive feasibility and robustness of r-LPC, which is the main focus of our work. Moreover, we show that the closed-loop asymptotic stability can be ensured under no external disturbance.

We first introduce the following assumptions.

(A4) $\|W_\star\| \le W_{\star,m}$, $\|h_\star\| \le h_{\star,m}$, $\|\bar{\epsilon}_\star\| \le \bar{\epsilon}_{\star,m}$, where $\star = a, c$ in turns.

Theorem 1 (Convergence of Actor–Critic): Under Assumption (A4) and the learning rules (28), (32), if learning rates γ_a and γ_c are such that $\bar{G}_{c1}, \bar{G}_{a1} > 0$ (whose definitions are deferred in Appendix A-A), then there exist $\eta_c, \eta_a \ge 0$ such that

$$\begin{aligned}\left\|\hat{\lambda}\big(\hat{z}(\tau+1)\big) - \lambda^*\big(\hat{z}(\tau+1)\big)\right\| &\le \eta_c\\ \left\|\hat{u}\big(\hat{z}(\tau)\big) - \hat{u}^*\big(\hat{z}(\tau)\big)\right\| &\le \eta_a.\end{aligned} \tag{33}$$

$\tau \in [k, k+N-1]$, as $N \to +\infty$. Also, if $\bar{\epsilon}_a, \bar{\epsilon}_c, \bar{\mu} \to 0$, and A is Schur stable, then

$$\hat{W}_c(\tau) \to W_c, \quad \hat{W}_a(\tau) \to W_a \tag{34}$$

and

$$\begin{aligned}\hat{\lambda}\big(\hat{z}(\tau+1)\big) &\to \lambda^*\big(\hat{z}(\tau+1)\big)\\ \hat{u}\big(\hat{z}(\tau)\big) &\to \hat{u}^*\big(\hat{z}(\tau)\big).\end{aligned} \tag{35}$$

$\tau \in [k, k+N-1]$, as $N \to +\infty$.

Proof: Refer to Appendix A-A. ■

Remark 4: Theorem 1 implies that a sufficient large choice of prediction horizon N is required to guarantee (33). To achieve this, an outer iterative loop with a length of $\bar{i}$ (see Algorithm 1) can be adopted since the weighting matrices $\hat{W}_c$ and $\hat{W}_a$ are time independent. This allows using a smaller choice of N via increasing $\bar{i}$, such that $\bar{i}N$ is sufficiently large.

To state the following theorem in a compact form, letting $\hat{u}(\hat{z}(\tau|k))$ be a control policy at any time k, we define a backup policy at time $k+1$ as:

$$\begin{aligned}\hat{\boldsymbol{u}}^s\big(\hat{z}(k+1)\big) = \hat{u}\big(\hat{z}(k+1|k)\big), \ldots, \hat{u}\big(\hat{z}(k+N-1|k)\big)\\ K\hat{z}(k+N|k).\end{aligned} \tag{36}$$

Also, we let $\boldsymbol{h}_a = [h_a(\hat{z}^s(k+1)), \ldots, h_a(\hat{z}^s(k+N))]$, where $\hat{z}^s$ are computed with control policy $\hat{\boldsymbol{u}}^s$.

Theorem 2 (Recursive Feasibility): Under Assumptions (A1)–(A4), if the learned control policy $\hat{u}(\hat{z}(\tau|k))$ is feasible (see Definition 2) at time k, then $\hat{\boldsymbol{u}}^s(\hat{z}(k+1))$ is feasible at time $k+1$. Also, if $\boldsymbol{h}_a$ satisfies the persistent excitation condition, i.e.,

$$\rho_1 I \le \boldsymbol{h}_a\boldsymbol{h}_a^\top \le \rho_2 I \tag{37}$$

where $\rho_1, \rho_2 > 0$, then the weight $\hat{W}_a$ satisfying

$$\hat{W}_a^\top = \hat{\boldsymbol{u}}^s\big(\hat{z}(k+1)\big)\boldsymbol{h}_a^\top\left(\boldsymbol{h}_a\boldsymbol{h}_a^\top\right)^\dagger \tag{38}$$

constructs a feasible control policy in the interval $[k, k+N-1]$.

Proof: Refer to Appendix A-B. ■

Remark 5: Note that $\hat{\boldsymbol{u}}(\hat{z}) = \hat{\boldsymbol{u}}^s(\hat{z})$ can be directly used whenever a learning failure occurs. For the sake of control performance improvement, one can use $\hat{W}_a$ (38) as a feasible initialization for improving the control policy.

Remark 6: We highlight that recent work in [43] proposed a Lyapunov-based model-free RL approach to control dynamical systems with safety constraints. However, the reliability of the safety guarantee was closely related to the data samples used in the training process. In our approach, the constraint satisfaction can be fulfilled by resorting to the receding horizon principle.

Theorem 3 (Closed-Loop Robustness): Under Assumptions (A1)–(A4), let at time k

$$\hat{u}(k) = \begin{cases} \hat{u}\big(\hat{z}(k|k)\big) & \text{if } \bar{V}(k|k) \le \bar{V}(k|k-1) \\ \hat{u}\big(\hat{z}(k|k-1)\big) & \text{otherwise} \end{cases} \tag{39}$$

then the closed-loop asymptotic convergence of $\hat{u}$, $\hat{z}$, and $\hat{x}$ to the origin and of u, z, and x to the robust tubes are verified, i.e., $\hat{u}(k), \hat{z}(k), \hat{x}(k) \to 0$, and $u(k) \to K\mathcal{O}_z$, $z(k) \to \mathcal{O}_z$, $x(k) \to \mathcal{O}_x$ as $k \to +\infty$.

Proof: Refer to Appendix A-C. ■

In the following, we prove the asymptotic stability under $w = 0$. To proceed, in view of [34], Ψ^{-1} is Lipschitz continuous under Assumption (A3). Then, one can derive

$$\|\bar{w}(z, u, 0)\| \le L_z\|z\| + L_u\|u\| \tag{40}$$

for all $z = \Psi(x)$ satisfying $x \in \mathcal{X}$ and $u \in \mathcal{U}$, where L_z and L_u are Lipschitz constants. From (40), one has $\|\bar{w}\| \le L_z\|z\| + L_u\|z\|_{K^\top K}$ in view of (15) and of $\hat{u}, \hat{z} \to 0$ asymptotically.

Theorem 4 (Asymptotic Stability): If $w = 0$ and there exists a scalar $\gamma > 0$ such that

$$\begin{bmatrix} F^\top PF - P + \gamma\bar{L} & F^\top P \\ PF & -\gamma I \end{bmatrix} \prec 0 \tag{41}$$

where $\bar{L} = (L_z I + L_u K^\top K)$, the closed-loop system (6) and (1) with (7) converge to the origin asymptotically, i.e., $x(k) \to 0$, $u(k) \to 0$, and $z(k) \to 0$ asymptotically.

Proof: Refer to Appendix A-D. ■

Remark 7: The interior-point numerical optimization method takes $O(N(\bar{n}+m)^3)$ operations for systems with block-diagonal structure and $O(N^3(\bar{n}+m)^3)$ operations if the block-diagonal structure is not exploited (see [44]). However, the online computational complexity of our method is roughly $O(N(\bar{n}+m)(\bar{n}+N_c))$ given $N_c \ge N_a$. Also, considering an offline training and an online deploying case, the computational load is only due to (14) and (30).

V. Simulation and Experimental Results

A. Simulation Results on Van Der Pol Oscillator

Consider a continuous-time Van der Pol oscillator [37]. Its model is given as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ -2x_2 + 10x_1^2x_2 + 0.8x_1 + u \end{bmatrix} + w \tag{42}$$

where x_1 and x_2 are the states, and u is the control, the disturbance $w = 0.4\sin(10\pi t)$. Let $x = (x_1, x_2)$, then the following constraints are restricted, i.e., $-(2.5, 2.5) \le x \le (2.5, 2.5)$, $-10 \le u \le 10$.

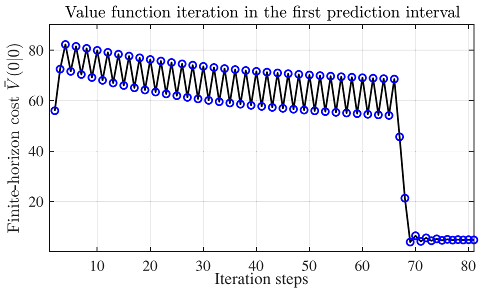


Fig. 3. Van der Pol oscillator: Variations of the value function $\bar{V}(0|0)$ in the first prediction interval.

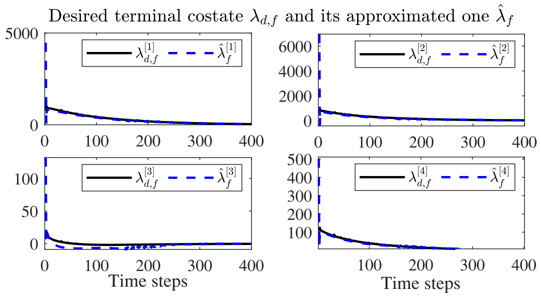


Fig. 4. Van der Pol oscillator: Desired terminal costate $\lambda_{d,f}$ and the approximated one $\hat{\lambda}_f$, where $\lambda_{d,f}^{[i]}$ ($\hat{\lambda}_f^{[i]}$) is the ith entry of $\lambda_{d,f}$ ($\hat{\lambda}_f$).

To implement the proposed r-LPC algorithm, model (42) was discretized with a sampling period $T = 0.01$ s to obtain the discrete-time counterpart, i.e., (1). Then, to derive the Koopman model, we selected a type of $\Psi(x)$ as $\Psi(x) = (x, \psi_1(x), \psi_2(x)) - (0, \psi_1(0), \psi_2(0))$ where $\psi_i(x) = \|x - c_i\|^2 \log(\|x - c_i\|)$, $i = 1, 2$, $\bar{n} = 4$, $c_1 = (0.381, -0.341)$, $c_2 = (0.267, -0.889)$ are the kernel centers randomly generated according to a uniform distribution. The model parameters and sets $\mathcal{V}$ and $\mathcal{D}$ are computed according to the steps described in Appendix A-E. The resulting constraint on the nominal state $\hat{x}$ was $-[2.38\ 2.05] \le \hat{x} \le [2.38\ 2.05]$.

In the proposed r-LPC, the penalty matrices $Q = I_2$ and $R = 0.1$. The penalty scalars related to barrier functions were $\mu = \bar{\mu} = 0.001$. The prediction horizon was $N = 10$. The functions h_a and h_c were chosen as $h_a = h_c = (\sigma(\hat{z}), \nu\tau, \nu\tau^2)$, where $\sigma(\hat{z}) = 1/(1 + \exp(-W_1\hat{z} + b_1))$, $\nu = 0.001$, where $W_1 \in \mathbb{R}^{8\times 4}$, $b_1 \in \mathbb{R}^8$, are weighting matrices.

The proposed r-LPC was first implemented with an initial condition $x(0) = (1.5, -1.5)$. The weighting matrices $\hat{W}_a$ and $\hat{W}_c$ were initialized as uniformly distributed (UD) random values. The simulations with r-LPC were performed in MATLAB 2019a environment running with Windows 10 operating system. In each prediction interval, multiple episodes were used to learn $\hat{W}_a$ and $\hat{W}_c$. The iterative variations of value function $\bar{V}$ in a prediction interval are displayed in Fig. 3, which shows $\bar{V}$ reduces gradually and converges to the neighbor of a constant value after about 70 iterative steps. In the following presented simulation results, the iteration in each prediction interval was terminated if the terminal region was reached and the episode was greater than 20. Also, as shown in Fig. 4, the approximated and desired terminal costates

TABLE I
VAN DER POL OSCILLATOR: COMPARISONS IN TERMS OF CUMULATIVE COST J

$x(0)$	$(0.5, -0.5)$	$(1, -1)$	$(1, -1.5)$	$(1.5 - 1.5)$	$(1.5, -2)$	$(1.8 - 1.8)$	$(2, -2)$
r-LPC (learned online)	45	70	**70**	**243**	260	**528**	**848**
r-LPC (learned off-line)	26	**66**	78	280	268	701	889
r-KMPC [34]	25	68	71	270	**247**	599	861
LQR	**22**	79	87	287	275	576	870

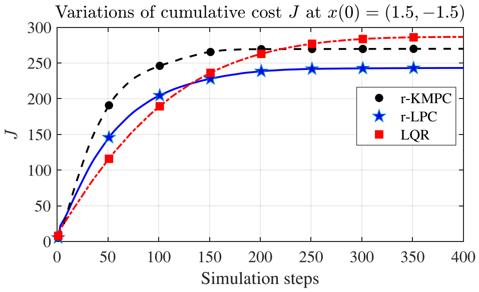


Fig. 5. Van der Pol oscillator: Comparison of the cumulative cost J of the three approaches at $x(0) = (1.5, -1.5)$.

converge asymptotically to the origin, and the approximated terminal costate by the critic network is close to the desired one.

Comparisons With r-KMPC [34] and an LQR: The comparative studies of r-LPC with r-KMPC [34] and an LQR using a local linearized model were also considered. For a fair comparison, parameters Q and R in the r-KMPC and LQR were selected the same as r-LPC. The variations of cumulative cost $J = \sum_{j=1}^{N_{\text{sim}}} \|x(j)\|_Q^2 + \|u(j)\|_R^2$, $N_{\text{sim}} = 400$, of all the three controllers are presented in Fig. 5. The results show that the control performance of r-LPC is slightly better than r-KMPC and LQR in terms of regulation cost under $x(0) = (1.5, -1.5)$. To further verify the effectiveness of r-LPC, we also conducted simulation tests with multiple different initial conditions. The resulting cumulative costs obtained with r-LPC (using online as well as offline learned policy), r-KMPC, and LQR are listed in Table I. The results reveal that r-LPC can perform slightly better in the case that initial conditions were far from the origin. Also, compared with LQR, r-LPC and r-KMPC can deal with constrained control problems. As a piece of evidence to the constraint satisfaction of r-LPC, the result of a safety test is presented in Fig. 6, which shows that r-LPC can recover a feasible control policy from constraint violation in the adopted prediction interval.

B. Simulation and Experimental Results on an Inverted Pendulum

Consider also the problem of regulating an Inverted Pendulum. The continuous-time nonlinear system model is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{3}{2l}(g\sin x_1 + u\cos x_1) \\ x_4 \\ u \end{bmatrix} + w \tag{43}$$

where x_1, x_2, x_3, and x_4 are the angle displacement and velocity of the rod, position and velocity of the car, $l = 0.5$ m, $g = 9.8$ m/s^2, $\|w\|_\infty \leq 0.5$. Denoting $x = (x_1, x_2, x_3, x_4)$, the state and control are limited as $-(0.25$ rad, 2 rad/s, 1 m, 2 m/s$) \leq x \leq (0.25$ rad, 2 rad/s, 1 m, 2 m/s$)$, $|u| \leq 20$ m/s^2. For a visual display of the inverted pendulum platform, see Fig. 7.

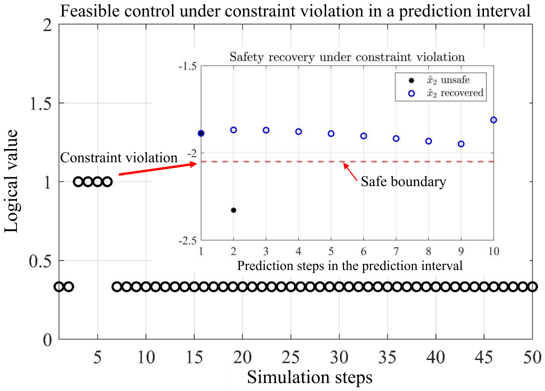


Fig. 6. Van der Pol oscillator: Feasible control recovery when a constraint violation occurs. The logical value being 1 means a constraint violation is detected when the proposed feasible control policy is applied to guarantee recursive feasibility of r-LPC.

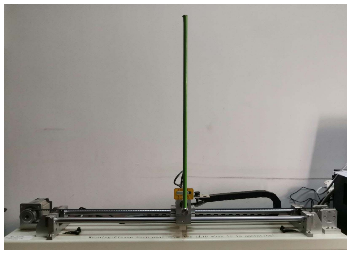


Fig. 7. Experimental platform of an inverted pendulum.

To derive the Koopman model, input–output datasets pairs of (43) with $M = 2 \cdot 10^4$ were collected using a random control policy with a uniform distribution. Gaussian kernel functions were selected to construct Ψ, i.e.,

$$\Psi(x) = (x, \psi_1(x), \psi_2(x), \psi_3(x)) - (0, \psi_1(0), \psi_2(0), \psi_3(0))$$

where $\psi_i(x) = e^{-\|x-c_i\|^2}$, $i = 1, 2, 3$, $\bar{n} = 7$, and kernel centers $c_1 = (0.59, -0.73, 0.14, 0.04)$, $c_2 = (0.67, 0.26, 0.2, 0.72)$, and $c_3 = (0.3, 0.99, 0.58, -0.37)$ are generated randomly. The parameters of the linear predictor were computed similar to the previous section (see [34]). The penalty matrices Q and R were selected as $Q = I_2$ and $R = 0.02$. $\mu = \bar{\mu} = 0.001$. The functions h_a and h_c were chosen as $h_a = h_c = (\sigma(\hat{z}), \nu\tau, \nu\tau^2)$, $\sigma(\hat{z}) = 1/(1 + \exp(-W_2\hat{z} + b_2))$, $W_2 \in \mathbb{R}^{12\times 7}$, $b_2 \in \mathbb{R}^{12}$, where $\nu = 0.001$. The variations of the terminal costate and its approximated value are displayed in Fig. 8, which shows that

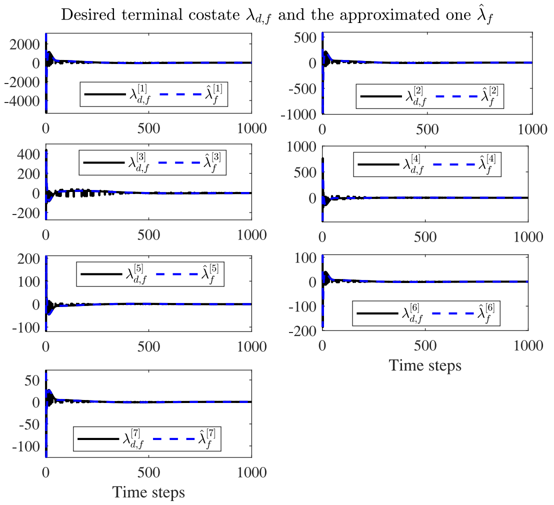


Fig. 8. Inverted pendulum: Desired terminal costate $\lambda_{d,f}$ and the approximated one $\hat{\lambda}_f$, where $\lambda_{d,f}^{[i]}$ ($\hat{\lambda}_f^{[i]}$) is the ith entry of $\lambda_{d,f}$ ($\hat{\lambda}_f$).

TABLE II
COMPARISONS WITH R-KMPC IN TERMS OF CUMULATIVE COST J

Prediction horizon N	8	12	16	20	32
r-LPC (learned online)	217	218	217	216	217
r-LPC (learned off-line)	149	148	148	147	148
r-KMPC [34]	151	151	151	151	151

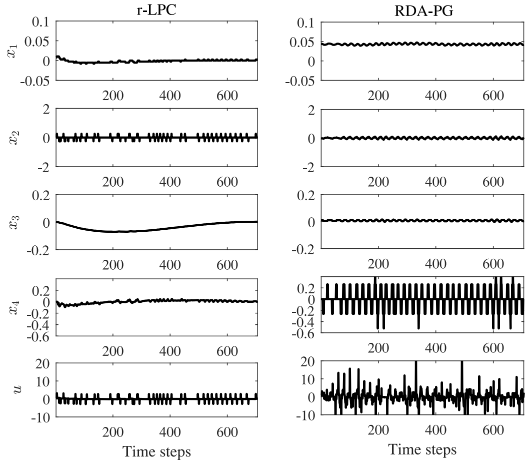


Fig. 9. Inverted pendulum: Control performance comparison between r-LPC and RDA-DG in the experimental tests under no exogenous disturbance.

the approximated and real terminal costates converge rapidly to the origin, and the approximated terminal costate is close to the desired one.

Comparison With r-KMPC in the Simulation Tests: The r-LPC and r-KMPC were implemented with an initial condition $x(0) = (0.1, 0.1, 0.2, 0.2)$ and a disturbance $w = 0.5\sin(10\pi t)$. The r-LPC was run ten times to improve the control performance. The final cumulative costs of r-LPC and r-KMPC, i.e., $J = \sum_{j=1}^{N_{\text{sim}}} \|x(j)\|_Q^2 + \|u(j)\|_R^2$, $N_{\text{sim}} = 600$, are collected and displayed in Table II, which show that the costs obtained with offline learned policies are comparable to that of r-KMPC with different prediction horizons.

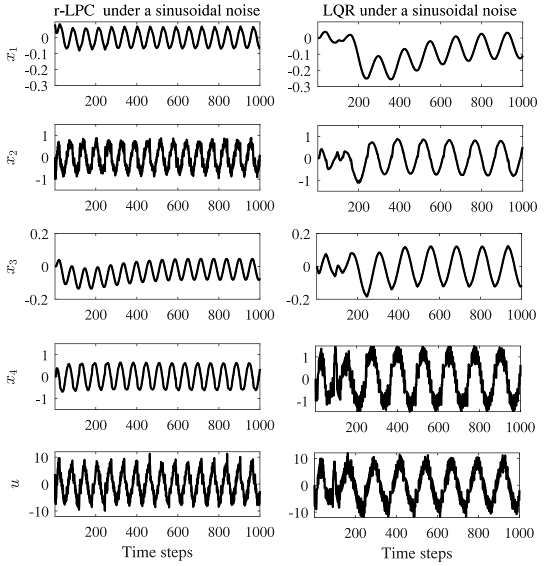


Fig. 10. Inverted pendulum: Control performance comparison between r-LPC and LQR in the experimental tests under an additive sinusoidal noise, i.e., $w = 0.1\sin(20\pi t)$.

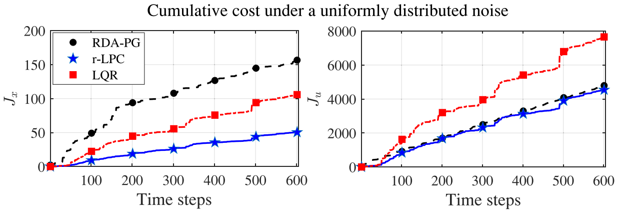


Fig. 11. Inverted pendulum: Comparison in terms of cumulative cost of the r-LPC, RDA-PG, and LQR in the experimental tests under UD noises.

Comparison With an RDA-PG [35] and an LQR in the Experimental Tests: In peculiar, a recently developed regularized policy gradient method, i.e., RDA-PG in [35] and LQR were used for comparisons in the experimental tests. The control policies used in the experimental studies of r-LPC and RDA-PG were learned offline. All the experimental tests were performed with the sampling interval $T = 0.01$ s. The tests were validated in three model conditions, i.e., the inverted pendulum was run under a nominal condition, a sinusoidal (S) noise $w = 0.1\sin(20\pi t)$, and a UD noise with an amplitude of 0.003 and a frequency of 10 Hz, respectively. The comparative results of r-LPC and RDA-PG in the nominal case are initially displayed in Fig. 9, which shows that our approach performs better than RDA-PG. The comparative results of r-LPC and LQR in Fig. 10 under the sinusoidal noise illustrate the strong point of our approach to LQR. Moreover, the comparative results of the three controllers under the UD noise in terms of the cumulative costs $J_x = \sum_{k=1}^{N_{\text{sim}}} \|x(k)\|^2$ and $J_u = \sum_{k=1}^{N_{\text{sim}}} \|u(k)\|^2$ are displayed in Fig. 11, which shows that our approach results in the lowest value of J_x and J_u for the regulation of state and control, respectively. For comprehensive comparison, the mean cumulative costs, $\bar{J}_x = J_x/N_{\text{sim}}$ and $\bar{J}_u = J_u/N_{\text{sim}}$, in the prescribed three model scenarios are collected and displayed in Table III, which shows the effectiveness of r-LPC in control of systems under disturbances when compared with RDA-PG and LQR.

TABLE III
INVERTED PENDULUM: COMPARISONS OF MEAN CUMULATIVE COSTS IN EXPERIMENTAL TESTS

Model		Nominal	S noise	UD noise
r-LPC	$\bar{J}_x$	0.01	**0.44**	**0.08**
	$\bar{J}_u$	0.79	**22.3**	**7.51**
RDA-PG [35]	$\bar{J}_x$	0.03	10.4	0.26
	$\bar{J}_u$	12	31.4	7.98
LQR	$\bar{J}_x$	**0.008**	0.99	0.17
	$\bar{J}_u$	**0.6**	33.5	12.6

VI. CONCLUSION

This article proposes a robust LPC scheme, i.e., r-LPC, for nonlinear discrete-time dynamical systems with unknown dynamics, state constraints, and uncertainties. In r-LPC, instead of numerically computing a control sequence by a nominal MPC, an actor–critic learning algorithm is proposed to learn an explicit time-dependent control policy in a receding-horizon manner. The resulting output of r-LPC is composed of an RL-based control policy and an offline nonlinear state-feedback policy. As the prominent feature, r-LPC can guarantee feasibility in each prediction interval if state constraint violations occur by actor–critic learning, which allows applying our approach to control safety-critical systems. The closed-loop robustness with r-LPC under approximation errors is proven, and asymptotic stability under no exogenous disturbance is obtained.

The proposed r-LPC algorithm has advantages over previous robust MPC and model-based optimal control methods. For instance, compared with tube MPC (see [7], [34]), r-LPC results in an explicit control policy, allowing for control implementation with limited computational resources; while compared to LQR, the advantage lies in the resultant nonlinear control policy, the constraint fulfilment, and the online learning ability due to the actor–critic learning. Future works will focus on the extension of r-LPC to continuous-time nonlinear systems.

APPENDIX

A. Proof of Theorem 1

1) Consider the Lyapunov function

$$L(\tau) = L_a(\tau) + L_c(\tau)$$

where $L_\star = (1/2)\mathrm{tr}(\tilde{W}_\star^\top \tilde{W}_\star)$, $\tilde{W}_\star = W_\star - \hat{W}_\star$, $\star = a, c$ in turns. In view of (28) and (32), the difference of L writes

$$\Delta L(\tau+1) = \Delta L_a(\tau) + \Delta L_c(\tau) \tag{44}$$

where

$$\Delta L_\star(\tau) = \underbrace{\mathrm{tr}\Big(\gamma_\star \nabla\delta_\star(\tau)^\top \tilde{W}_\star(\tau)\Big)}_{\nabla L_{\star,1}(\tau)} + \underbrace{0.5\mathrm{tr}\Big({\gamma_\star}^2 \nabla\delta_\star(\tau)^\top \nabla\delta_\star(\tau)\Big)}_{\nabla L_{\star,2}(\tau)}. \tag{45}$$

$\nabla\delta_\star = \partial\delta_\star/\partial\hat{W}_\star$, $\star = a, c$ in turns. To compute ΔL_c, in view of (27), one writes

$$\nabla\delta_c(\tau) = \underbrace{\frac{\partial\mathrm{tr}\Big(\|\epsilon_c(\tau)\|^2 + \big\|\epsilon_{c,N}\big\|^2\Big)}{\partial\hat{W}_c}}_{\nabla\delta_{c,1}(\tau)} + \underbrace{\frac{\partial\mathrm{tr}\Big(\bar{\mu}\Big(\mathcal{B}\Big(\hat{\lambda}(\tau)\Big) + \mathcal{B}_f\Big(\hat{\lambda}\Big)\Big)\Big)}{\partial\hat{W}_c}}_{\nabla\delta_{c,2}(\tau)}. \tag{46}$$

For the sake of simplicity, in the rest of the Appendix, we use q to denote a generic variable q with time index τ, i.e., $q(\tau)$, q^+ to represent $q(\tau+1)$, $q_N = q(k+N)$. In view of the definition of λ_d and λ^*, it holds that

$$\begin{aligned}\epsilon_c &= \lambda_d - \lambda^* + \lambda^* - \hat{\lambda}\\ &= \xi_c + \Delta\bar{\epsilon}_c\end{aligned} \tag{47}$$

where $\xi_c = -A^\top \tilde{W}_c^\top h_c^+ + \tilde{W}_c^\top h_c$, $\Delta\bar{\epsilon}_c = \bar{\epsilon}_c - A^\top \bar{\epsilon}_c^+$. The term $\nabla\delta_{c,1}$ in (46) can be computed as

$$\begin{aligned}\nabla\delta_{c,1} = -2\Big(&h_c(\xi_c + \Delta\bar{\epsilon}_c)^\top - h_c^+(\xi_c + \Delta\bar{\epsilon}_c)^\top A^\top\\ &+ h_{cN}(\xi_{cN} + \bar{\epsilon}_{cN})^\top\Big)\end{aligned} \tag{48}$$

where $\xi_{cN} = \tilde{W}_c^\top h_{cN}$, $h_{cN} = h_c(k+N)$.

Consider the constraint on $\hat{\lambda}$ of type $\Lambda^i = \{\hat{\lambda} | \hat{\lambda}^\top Z_i \hat{\lambda} \le 1\}$. Then, one can compute the term $\nabla\delta_{c,2}$ in (46) as

$$\nabla\delta_{c,2} = 2\bar{\mu}\Big(\kappa_\lambda^{-1} h_c h_c^\top \hat{W}_c Z_i + \kappa_{\lambda N}^{-1} h_{cN} h_{cN}^\top \hat{W}_c Z_N\Big) := 2\bar{\mu}\omega \tag{49}$$

where $\kappa_\lambda = 1 - \hat{\lambda}^\top Z_i \hat{\lambda}$, $\kappa_{\lambda N} = 1 - \hat{\lambda}^\top Z_N \hat{\lambda}$.

In view of (48) and (49), one can write the first term in (45) as

$$\begin{aligned}\nabla L_{c,1} &= -2\gamma_c \mathrm{tr}\Big(\xi_c(\xi_c + \Delta\bar{\epsilon}_c)^\top + \xi_{cN}(\xi_{cN} + \bar{\epsilon}_{cN})^\top - \bar{\mu}\tilde{W}_c^\top \omega\Big)\\ &= -\mathrm{tr}\Big(\big\|\bar{\xi}_c\big\|_M^2 + \bar{\xi}_c^\top M \rho_1 - 2\gamma_c \bar{\mu} \tilde{W}_c^\top \omega\Big)\end{aligned} \tag{50}$$

where $\bar{\xi}_c = (\xi_c, \xi_{cN})$, $\rho_1 = (\Delta\bar{\epsilon}_c, \bar{\epsilon}_{cN})$, and $M = 2\gamma_c I_2$. Also, one can compute

$$\nabla L_{c,2} = \mathrm{tr}\Big(-\big\|\bar{\xi}_c\big\|_{G_1}^2 + \bar{\xi}_c^\top (G_2\rho_1 + G_3\rho_2) + \bar{g}_{\bar{\epsilon}_c}\Big) \tag{51}$$

where $\rho_2 = (\hat{W}_c^\top h_c, \hat{W}_c^\top h_{cN})$, $\bar{g}_{\bar{\epsilon}_c} = 2\gamma_c^2(\varphi_c^\top \varphi_c + 2\bar{\mu}\varphi_c^\top \omega + \bar{\mu}^2 \omega^\top \omega))$, and $\varphi_c = -h_c \Delta\bar{\epsilon}_c^\top + h_c^+ \Delta\bar{\epsilon}_c^\top A^\top - h_{cN}\bar{\epsilon}_{cN}^\top$

$$G_1 = -\begin{bmatrix} g_1 & g_{12}^\top \\ g_{12} & g_2 \end{bmatrix}$$

$g_1 = 2\gamma_c^2(\bar{h}_c + \bar{h}_c^+ A^\top A - 2A^\top h_c^\top h_c^+)$, $g_2 = 2\gamma_c^2 \bar{h}_{cN}$, $g_{12} = 2\gamma_c^2(h_c^\top h_{cN} - A(h_c^+)^\top h_{cN})$, $\bar{q} = q^\top q$ for $q = h_c, h_{cN}, h_c^+$

$$G_2 = \begin{bmatrix} g_3 & g_4 \\ g_5 & g_6 \end{bmatrix}$$

and where $g_3 = 4\gamma_c^2(\bar{h}_c - h_c^\top h_c^+ A - (h_c^+)^\top h_c A^\top + \bar{h}_c^+ A A^\top)$ $g_4 = 4\gamma_c^2(-(h_c^+)^\top h_{cN} A + h_c^\top h_{cN})$, $g_5 = 4\gamma_c^2(h_{cN}^\top h_c - h_{cN}^\top h_c^+ A)$, $g_6 = 4\gamma_c^2 h_c^\top h_{cN}$

$$G_3 = \begin{bmatrix} g_7 & g_8 \\ g_9 & g_{10} \end{bmatrix}$$

and where $g_7 = 4\bar{\mu}\gamma_c^2\kappa_\lambda^{-1}(-\bar{h}_c + (h_c^+)^\top h_c A)Z_i$, $g_8 = 4\bar{\mu}\gamma_c^2\kappa_{\lambda N}^{-1}(-h_c^\top h_{cN} + (h_c^+)^\top h_{cN})A)Z_N$, $g_9 = -4\bar{\mu}\gamma_c^2\kappa_\lambda^{-1}h_{cN}^\top h_c Z_i$, and $g_{10} = -4\bar{\mu}\gamma_c^2\kappa_{\lambda N}^{-1}\bar{h}_{cN}Z_N$.

Taking (50) and (51) into consideration, ΔL_c results

$$\Delta L_c = \mathrm{tr}\Big(-\big\|\bar{\xi}_c\big\|_{G_{c1}}^2 + \bar{\xi}_c^\top(G_{c2}\rho_1 + G_{c3}\rho_2) + g_{\bar{\epsilon}_c}\Big) \quad (52)$$

where $G_{c1} = G_1 + M$, $G_{c2} = G_2 - M$, $G_{c3} = G_3$, and $g_{\bar{\epsilon}_c} = \bar{g}_{\bar{\epsilon}_c} + 2\gamma_c\bar{\mu}\tilde{W}_c^\top\omega$.

Applying Young's inequality and the Cauthy–Schwartz inequality, it holds that

$$\mathrm{tr}\Big(\bar{\xi}_c^\top(G_{c2}\rho_1 + G_{c3}\rho_2)\Big) \leq \frac{\alpha_c}{2}\Big(\|G_{c2}\|^2 + \|G_{c3}\|^2\Big)\|\bar{\xi}_c\|^2 + \frac{1}{2\alpha_c}\Big(\|\rho_1\|^2 + \|\rho_2\|^2\Big)$$

where $\alpha_c > 0$. As $\hat{W}_c$ is bounded provided δ_c is finite, we assume $\|\hat{W}_c\| \leq W_{c,m}$, then $\|\tilde{W}_c\| \leq 2W_{c,m}$ in view of Assumption (A4). One can promptly have $\|\omega\| \leq \bar{\omega}$, $\|\varphi_c\| \leq \bar{\varphi}_c$, and $\|g_{\bar{\epsilon}_c}\| \leq g_{c,m}$, where the derivation of $\bar{\omega}$, $\bar{\varphi}_c$, and $g_{c,m}$ is trivial and neglected for simplicity. Hence, (52) leads to

$$\Delta L_c(\tau) \leq -\big\|\bar{\xi}_c(\tau)\big\|_{\bar{G}_{c1}}^2 + \mathrm{res_c} \quad (53)$$

where $\bar{G}_{c1} = G_{c1} + [\alpha_c/2](\|G_{c2}\|^2 + \|G_{c3}\|^2)$, $\mathrm{res_c} = g_{c,m} + [1/2\alpha_c](2\bar{\epsilon}_{c,m}^2 + 2W_{c,m}^2 h_{c,m}^2)$.

As a second step, we compute the ΔL_a in (44). Note that in view of (31), one has

$$\nabla\delta_a = \frac{\partial\mathrm{tr}\big(\|\epsilon_a\|^2 + \bar{\mu}\mathcal{B}(\hat{u})\big)}{\partial\hat{W}_a}.$$

Consider the input constraint of type $\hat{\mathcal{U}} = \{\hat{u}|\sum_{i=1}^p a_{u,i}^\top\hat{u} \leq 1, \forall i \in \mathbb{R}_1^p\}$. One can write $[(\partial B(\hat{u}))/(\partial\hat{u})] = \sum_{i=1}^p a_{u,i}^\top\kappa_{\hat{u},i}^{-1}$, where $\kappa_{\hat{u},i} = 1 - a_{u,i}^\top\hat{u}$. Hence

$$\frac{\partial\mathrm{tr}\big(\|\epsilon_a\|^2\big)}{\partial\hat{W}_a} = -2h_a\Big(\varphi_a + \bar{R}_2\tilde{W}_a^\top h_a\Big)^\top\bar{R}_1 \quad (54)$$

where $\varphi_a = (\tilde{a}_{u,i}+I)\bar{\epsilon}_a + B^\top\tilde{W}_c^\top h_c^+ + B^\top\bar{\epsilon}_c^+$, $\bar{R}_1 = 2R + \kappa_{\hat{u},i}^{-2}\tilde{a}_{u,i}$, $\bar{R}_2 = 2R + \sum_{i=1}^p\tilde{a}_{u,i}\kappa_{u,i}^{-1}\kappa_{\hat{u},i}^{-1}$, $\kappa_{u,i} = 1 - a_{u,i}^\top\hat{u}_d$, $\tilde{a}_{u,i} = a_{u,i}a_{u,i}^\top$. Recalling the fact that $[(\partial a^\top X^\top b)/\partial X] = ba^\top$ where X is a matrix, a and b are two vectors, one has

$$\frac{\partial\mathrm{tr}\big(\mathcal{B}\big(\hat{u}(\tau)\big)\big)}{\partial\hat{W}_a(\tau)} = h_a l_u \quad (55)$$

where $l_u = \sum_{i=1}^p a_{u,i}^\top\hat{\kappa}_{u,i}^{-1}$. Hence, denoting $\xi_a = \tilde{W}_a^\top h_a$, in view of (54) and (55), it follows that:

$$\Delta L_a = \mathrm{tr}\Big(-\|\xi_a\|_{G_{a1}}^2 + G_{a2}\xi_a + g_{\bar{\epsilon}_a}\Big)$$

where $G_{a1} = (2\gamma_a I - 2\gamma_a^2\bar{h}_a\bar{R}_2\bar{R}_1)\bar{R}_1\bar{R}_2$, $G_{a2} = (4\gamma_a^2\bar{R}_2\bar{R}_1\bar{R}_1\varphi_a\bar{h}_a - 2\gamma_a\bar{R}_1\varphi_a + \bar{\mu}\gamma_a lu^\top + 2\gamma_a^2\bar{\mu}\bar{R}_2\bar{R}_1 lu^\top\bar{h}_a)^\top$, $\bar{h}_a = h_a^\top h_a$, $g_{\bar{\epsilon}_a} = 2\gamma_a^2\bar{R}_1\varphi_a\bar{h}_a\varphi_a^\top\bar{R}_1 - 2\gamma_a^2\bar{\mu}lu^\top\bar{h}_a\phi_a\bar{R}_1 + 1/2\gamma_a^2\bar{\mu}^2 lu^\top\bar{h}_a lu$.

Likewise, in view of Assumption (A4), one can promptly have $\|\varphi_a\| \leq \bar{\phi}_a$, $\|l_u\| \leq \bar{l}_u$, $\|g_{\bar{\epsilon}_a}\| \leq g_{a,m}$, and $\|G_{a2}\| \leq G_{am}$, where the derivation of $\bar{\phi}_a$, $\bar{l}_u$, $g_{a,m}$, and G_{am} is trivial and is neglected for simplicity. Applying Young's inequality and the Cauthy–Schwartz inequality leads to

$$\Delta L_a \leq -\|\xi_a\|_{\bar{G}_{a1}}^2 + \mathrm{res_a} \quad (56)$$

where $\bar{G}_{a1} = G_{a1} - (\alpha_a\gamma_a/2)$, $\alpha_a > 0$ is a tuning parameter, $\mathrm{res_a} = \frac{2}{\alpha_a\gamma_a}G_{am}^2 + g_{a,m}$. With (53) and (56), one can write

$$\Delta L \leq -\big\|\bar{\xi}_c\big\|_{\bar{G}_{c1}}^2 - \|\xi_a\|_{\bar{G}_{a1}}^2 + \mathrm{res} \quad (57)$$

where $\mathrm{res} = \mathrm{res_a} + \mathrm{res_c}$, leading to the variables $\bar{\xi}_c(\tau)$ and $\xi_a(\tau)$ converging to the corresponding sets as $\tau \to N$ and $N \to +\infty$, i.e.,

$$\big\|\bar{\xi}_c(\tau)\big\| \leq \frac{\mathrm{res}}{\sigma_{\min}\big(\bar{G}_{c1}\big)}, \quad \|\xi_a(\tau)\| \leq \frac{\mathrm{res}}{\sigma_{\min}\big(\bar{G}_{a1}\big)}. \quad (58)$$

In view of the definition of $\bar{\xi}_c$ [cf. (47)] and ξ_a, one has

$$\bar{\xi}_c = \epsilon_c^l - \tilde{\epsilon}_c, \quad \xi_a = \epsilon_a - \bar{\epsilon}_a$$

where $\epsilon_c^l = (\epsilon_c, \epsilon_{cN})$ and $\tilde{\epsilon}_c = (\Delta\bar{\epsilon}_c, \bar{\epsilon}_{cN})$. Hence, (58) results

$$\Big\|\epsilon_c^l(\tau)\Big\| \leq \frac{\mathrm{res}}{\sigma_{\min}\big(\bar{G}_{c1}\big)} + \pi_A\epsilon_{c,m} := \eta_c \quad (59a)$$

$$\|\epsilon_a(\tau)\| \leq \frac{\mathrm{res}}{\sigma_{\min}\big(\bar{G}_{a1}\big)} + \bar{\epsilon}_{a,m} := \eta_a \quad (59b)$$

where the inequality in (59a) comes from that $\|\tilde{\epsilon}_c\| = \sqrt{\Delta\bar{\epsilon}_c^\top\Delta\bar{\epsilon}_c + \bar{\epsilon}_{cN}^\top\bar{\epsilon}_{cN}} \leq \pi_A\epsilon_{c,m}$ and $\pi_A = \sqrt{2 + 2\|A\| + \|A^2\|}$.

2) Provided that $\bar{\epsilon}_c = 0$, $\bar{\epsilon}_{cN} = 0$, and $\bar{\mu} = 0$, one has $\Delta L_c(\tau) \leq -\|\bar{\xi}_c(\tau)\|_{\bar{G}_{c1}}^2$ which implies that $\bar{\xi}_c(\tau) \to 0$ exponentially. That is to say, $\hat{\lambda}(\tau)$ converges to $\lambda_d(\tau)$ at an exponential rate. In view of this, one can rewrite (26) as

$$\lambda_d^{i+1}\big(\hat{z}(\tau)\big) = \mu\frac{\partial\mathcal{B}\big(\hat{z}(\tau)\big)}{\partial\hat{z}(\tau)} + 2\bar{Q}\hat{z}(\tau) + A^\top\lambda_d^i\big(\hat{z}(\tau+1)\big) \quad (60)$$

where the superscript i is the iterative step corresponding to $\hat{W}_c(\tau+i)$, i.e., $\lambda_d^i(\hat{z}(\tau)) = \hat{W}_c(\tau+i)^\top h_c(\tau)$. In view of (22) and denoting $\tilde{\lambda}$ as the subtraction of λ^* and λ_d, one promptly has $\tilde{\lambda}^{i+1}(\tau) = A^\top\tilde{\lambda}^i(\tau+1)$. By induction, one has

$$\tilde{\lambda}^i(\tau) = (A^\top)^i\tilde{\lambda}^0(\tau+i) \to 0$$

as $i \to +\infty$, due to the fact $\|A^\top\| \to 0$ for $i \to +\infty$ in view of A being Schur stable. This also implies $\hat{\lambda}(\tau) \to \lambda^*(\tau)$ for $\tau \in [k, k+N-1]$ as $N \to +\infty$. With this result, the term $\hat{W}_c^\top h_c^+$ in (54) is a vanishing term, then one has

$$\Delta L_a \leq -\|\xi_a\|_{\bar{G}_{a1}}^2$$

leading to the result that ξ_a converges to ξ_a^*. Hence, $\hat{W}_c(\tau) \to W_c$ and $\hat{W}_a(\tau) \to W_a$ as $\tau \to N$ and $N \to +\infty$.

B. Proof of Theorem 2

Assume that at any time instant k, a feasible control sequence can be generated with (30) at time k. We denote the near-optimal control policy as $\hat{u}(\hat{z}(k|k)), \ldots, \hat{u}(\hat{z}(k+N-1|k))$ associated with a near-optimal cost given as $\bar{V}(k)$ such that $x(k+i|k) \in \mathcal{X}$, $u(k+i|k) \in \mathcal{U}$ $\forall i \in [0, N-1]$, $\hat{z}(k+N|k) \in \mathcal{Z}_f$. At the next time instant $k+1$, (36) is a feasible choice such that $x(k+i|k+1) \in \mathcal{X}$, $u(k+i|k+1) \in \mathcal{U}$ $\forall i \in$

$[1, N]$, $\hat{z}(k+N+1|k+1) \in \mathcal{Z}_f$. This is due to the standard recursive feasibility argument of MPC (see [6]). Hence, with (36), the recursive feasibility of the finite-horizon RL follows. Moreover, provided the feasible policy $\hat{\boldsymbol{u}}^s(\hat{z}(k+1))$ [see (36)] at time $k+1$, the corresponding weight satisfies $\hat{W}_a^\top \boldsymbol{h}_a = \hat{\boldsymbol{u}}^s(\hat{z}(k+1))$ leading to (38) in view of (37). ■

C. Proof of Theorem 3

At time instant $k+1$, the learned cost $\bar{V}(k+1|k+1)$ might not be smaller than $\bar{V}(k+1|k)$. In this case and the case that feasibility is not guaranteed, the control policy (39) can be applied. Hence, denoting $\bar{V}^s(k+1) = \min\{\bar{V}(k+1|k+1), \bar{V}(k+1|k)\}$, one has

$$\begin{aligned}\bar{V}^s(k+1) &- \bar{V}^s(k)\\ &\leq -\Big(\big\|\hat{z}(k)\big\|_{\bar{Q}}^2 + \big\|\hat{u}(k)\big\|_R^2 + \mu\mathcal{B}\big(\hat{z}(k)\big) + \mu\mathcal{B}\big(\hat{u}(k)\big)\Big)\\ &\quad - \mu\mathcal{B}_f\big(\hat{z}(k+N)\big) + \mu\mathcal{B}_f\big(\hat{z}(k+N+1)\big)\\ &\quad + \big\|\hat{z}(k+N)\big\|_{F^\top PF - P + \bar{Q} + K^\top RK + \mu H}^2. \end{aligned} \tag{61}$$

Recalling that

$$\begin{aligned}&\mathcal{B}_f\big(\hat{z}(k+N+1)\big) - \mathcal{B}_f\big(\hat{z}(k+N)\big)\\ &\quad = \frac{1}{1 - \hat{z}(k+N)^\top F^\top ZF\hat{z}(k+N)} - \frac{1}{1 - \hat{z}(k+N)^\top Z\hat{z}(k+N)}\\ &< 0 \end{aligned} \tag{62}$$

in view of (17), from (61), the monotonic property of the value function is obtained, i.e., $\bar{V}(k+1) - \bar{V}(k) \leq -(\|\hat{z}(k)\|_{\bar{Q}}^2 + \|\hat{u}(k)\|_R^2 + \mu\mathcal{B}(\hat{z}(k)) + \mu\mathcal{B}(\hat{u}(k)))$, which leads to $\bar{V}(k+1) - \bar{V}(k) \to 0$ as, $k \to \infty$. Hence, $\hat{z}(k)$, $\hat{u}(k) \to 0$ asymptotically. Consequently, $\hat{x}(k) \to 0$ asympotically since $\hat{x} = C\hat{z}$. Recall that the real state remains inside the tube, i.e., $\hat{z} \oplus \mathcal{Z} \in \mathcal{X}$, then the robustness is obtained, i.e., $z(k) \to \mathcal{O}_z$, consequently, $x(k) \to \mathcal{O}_x$ asymptotically.

D. Proof of Theorem 4

We present a different proof to [34]. Recalling that $\hat{u}$, $\hat{z}$, $\hat{x} \to 0$ (cf. Theorem 3) and $w = 0$, one can write (6) as

$$\begin{cases} z(k+1) = Fz(k) + \bar{w}(k), \ \|\bar{w}(k)\| \leq \|z(k)\|_{\bar{L}} \\ x(k) = Cz(k) + v(k). \end{cases} \tag{63}$$

Consider the Lyapunov function $V(k) = z(k)^\top Pz(k)$. To guarantee asymptotic stability, its difference $\Delta V(k) = z(k+1)^\top Pz(k+1) - z(k)^\top Pz(k) = z(k)^\top (F^\top PF - P)z(k) + 2z(k)^\top F^\top Pw(k) + w(k)^\top w(k) < 0$, leading to the linear matrix inequality (41) by applying the S-procedure in [45] with the condition $\|\bar{w}(k)\| \leq \|z(k)\|_{\bar{L}}$. Indeed, with (41), one has that $z(k) \to 0$ asymptotically. Consequently, it follows that $u(k) \to 0$ asymptotically. In view of $\Psi(0) = 0$ one has $v = 0$ as $x = 0$. Hence, $x = Cz + v \to 0$ asymptotically.

E. Derivation of A, B, C, and D and Sets $\mathcal{D}$ and $\mathcal{V}$

Inline with [34] and [37], the matrices A, B, C, and D can be computed using input and output datasets. Let us assume to have L datasets of (w_i, u_i, x_i, x_i^+), where x_i^+ is the successor state of x_i by applying (w_i, u_i) to (1), and w_i can be computed using nonlinear estimation methods if it is not measurable (see [37]). It is assumed that $\{(u_i, x_i)\}_{i=1}^L$ are drawn independently according to a nonnegative probability distribution. Then, A, B, C, and D can be obtained via minimizing a type of cost function defined as

$$\begin{aligned}\text{Loss} = &\sum_{i=1}^{L}\big\|A\Psi(x_i) + Bu_i + Dw_i - \Psi\big(x_i^+\big)\big\|^2\\ &+ \|C\Psi(x_i) - x_i\|^2 + \alpha\|[A \ B \ D]\|^2 + \beta\|C\|^2 \end{aligned} \tag{64}$$

where α and β are positive scalars. With (64) being solved, sets $\mathcal{D}$ and $\mathcal{V}$ can be estimated according to the empirical risk evaluation in [10]. For computational details on the derivation of $\mathcal{D}$ and $\mathcal{V}$, refer to [34].

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 11, no. 7, pp. 733–764, 2003.

[3] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 7130–7135.

[4] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1600–1606, Sep. 2003.

[5] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. London, U.K.: Springer, 1999, pp. 207–226.

[6] D. Q. Mayne, M. M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[7] P. Falugi and D. Q. Mayne, "Getting robustness against unstructured uncertainty: A tube-based MPC approach," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1290–1295, May 2014.

[8] S. Lucia and B. Karg, "A deep learning-based approach to robust nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 511–516, 2018.

[9] J. A. Paulson and A. Mesbah, "Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction," *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 719–724, Jul. 2020.

[10] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 543–548, Jul. 2018.

[11] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, "Adaptive dynamic programming for control: A survey and recent advances," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 142–160, Jan. 2021.

[12] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[13] D. Wang, D. Liu, Q. Zhang, and D. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1544–1555, Nov. 2016.

[14] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, vol. 17. Hoboken, NJ, USA: Wiley, 2013.

[15] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[16] Y. Liu, L. Tang, S. Tong, C. L. P. Chen, and D. Li, "Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015.

[17] B. Luo, D. Liu, T. Huang, and J. Liu, "Output tracking control based on adaptive dynamic programming with multistep policy evaluation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2155–2165, Oct. 2019.

[18] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[19] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[20] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[21] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[22] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[23] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[24] T. Li, D. Yang, X. Xie, and H. Zhang, "Event-triggered control of nonlinear discrete-time system with unknown dynamics based on HDP (λ)," *IEEE Trans. Cybern.*, early access, Feb. 2, 2021, doi: 10.1109/TCYB.2020.3044595.

[25] K. Zhang, R. Su, H. Zhang, and Y. Tian, "Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5502–5511, Dec. 2021.

[26] H. Xu, Q. Zhao, and S. Jagannathan, "Finite-horizon near-optimal output feedback neural network control of quantized nonlinear discrete-time systems with input constraint," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1776–1788, Aug. 2015.

[27] Q. Zhao, H. Xu, and S. Jagannathan, "Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 486–499, Mar. 2015.

[28] D. Görges, "Relations between model predictive control and reinforcement learning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4920–4928, 2017.

[29] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: A comparison on a power system problem," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 517–529, Apr. 2009.

[30] M. Zanon and S. Gros, "Safe reinforcement learning using robust MPC," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, Aug. 2021.

[31] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proc. IEEE Conf. Decis. Control (CDC)*, 2018, pp. 6059–6066.

[32] X. Xu, H. Chen, C. Lian, and D. Li, "Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6202–6213, Dec. 2018.

[33] L. Dong, J. Yan, X. Yuan, H. He, and C. Sun, "Functional nonlinear model predictive control based on adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4206–4218, Dec. 2019.

[34] X. Zhang, W. Pan, R. Scattolini, S. Yu, and X. Xu, "Robust tube-based model predictive control with Koopman operators," *Automatica*, vol. 137, Mar. 2022, Art. no. 110114.

[35] L. Li, D. Li, T. Song, and X. Xu, "Actor-critic learning control with regularization and feature selection in policy gradient estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1217–1227, Mar. 2021.

[36] A. Mauroy and Y. Susuki, "Introduction to the Koopman operator in systems and control," in *Proc. SICE Conf.*, 2018, pp. 59–63.

[37] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, Jul. 2018.

[38] A. G. Wills and W. P. Heath, "Barrier function based model predictive control," *Automatica*, vol. 40, no. 8, pp. 1415–1422, 2004.

[39] J. Nocedal and S. Wright, *Numerical Optimization*. Cham, Switzerland: Springer, 2006.

[40] C. Feller and C. Ebenbauer, "Relaxed logarithmic barrier function based model predictive control of linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1223–1238, Mar. 2017.

[41] A. Heydari and S. N. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 145–157, Jan. 2013.

[42] F. Scibilia, S. Olaru, and M. Hovd, "On feasible sets for MPC and their approximations," *Automatica*, vol. 47, no. 1, pp. 133–139, 2011.

[43] M. Han, Y. Tian, L. Zhang, J. Wang, and W. Pan, "Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee," *Automatica*, vol. 129, Jul. 2021, Art. no. 109689.

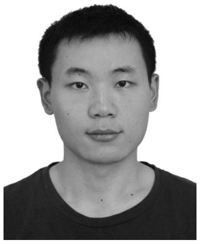
[44] W. Yang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

[45] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory." Philadelphia, PA, USA: Soc. Ind. Appl. Math., 1994.

Xinglong Zhang was born in Anhui, China, in 1990. He received the B.E. degree in mechanical engineering from Zhejiang University, Hangzhou, China, in 2011, the M.S. degree in mechanical engineering from the PLA University of Science and Technology, Nanjing, China, in 2014, and the Ph.D. degree in system and control from the Politecnico di Milano, Milan, Italy, 2018.

He is currently an Associate Professor with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. His research interests include learning-based model predictive control, adaptive dynamic programming, and their applications in automotive systems.

Dr. Zhang is a member of IEEE CIS Technical Committee on Adaptive Dynamic Programming and Reinforcement Learning.



Jiahang Liu received the bachelor's degree in automation from the College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2015, and the master's degree in control science and engineering from the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China, in 2018, where he is currently pursuing the Ph.D. degree in control science and engineering with the College of Intelligence Science and Technology.

Xin Xu (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Department of Automatic Control, National University of Defense Technology (NUDT), Changsha, China, in 1996, and the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, NUDT, in 2002.

He has been a Visiting Professor with Hong Kong Polytechnic University, Hong Kong; the University of Alberta, Edmonton, AB, Canada; the University of Guelph, Guelph, ON, Canada; and the University of Strathclyde, Glasgow, U.K. He is currently a Professor with the College of Intelligence Science and Technology, NUDT. He has coauthored more than 160 articles in international journals and conferences, and four books. His research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics, and autonomous vehicles.

Dr. Xu received the second class National Natural Science Award of China in 2012. He is currently an Associate Editor of the *Information Sciences*, the *Intelligent Automation and Soft Computing*, and the *Acta Automatica Sinica*, and served as a Guest Editor of the *International Journal of Adaptive Control and Signal Processing* and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. He is a member of the IEEE Computational Intelligence Society Technical Committee on Approximate Dynamic Programming and Reinforcement Learning and the IEEE Robotics and Automation Technical Committee on Robot Learning.

Shuyou Yu received the B.S. and M.S. degrees in control science and engineering from Jilin University, Changchun, China, in 1997 and 2005, respectively, and the Ph.D. degree in system dynamics and control engineering from the University of Stuttgart, Stuttgart, Germany, in 2011.

From 2010 to 2011, he was a Research and a Teaching Assistant with the Institute for Systems Theory and Automatic Control, University of Stuttgart. In 2012, he joined the Faculty of the Department of Control Science and Engineering, Jilin University, where he is currently a Full Professor. His current research interests include model predictive control, robust control, and applications in mechatronic systems.

Hong Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in process control from Zhejiang University, Hangzhou, China, in 1983 and 1986, respectively, and the Ph.D. degree in system dynamics and control engineering from the University of Stuttgart, Stuttgart, Germany, in 1997.

In 1986, she joined the Jilin University of Technology, Changchun, China. From 1993 to 1997, she was a Wissenschaftlicher Mitarbeiter with the Institut fuer Systemdynamik und Regelungstechnik, University of Stuttgart. Since 1999, she has been a Professor with Jilin University and hereafter a Tang Aoqing Professor. From 2015 to 2019, she was the Director of the State Key Laboratory of Automotive Simulation and Control. In 2019, she joined the Tongji University, Shanghai, China, as a Distinguished Professor. Her current research interests include model predictive control, nonlinear control, and applications in mechatronic systems focusing on automotive systems.