Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Digital Receding-Horizon Learning Controller for Nonlinear Continuous-time Systems

Xinglong Zhang * **Wenzhang Li** * **Xin Xu** * **Wei Jiang** *

* *College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China.*

Abstract: The integration of reinforcement learning (RL) and model predictive control (MPC) is promising for solving nonlinear optimization problems in an efficient manner. In this paper, a digital receding horizon learning controller is proposed for continuous-time nonlinear systems with control constraints. The main idea is to develop a digital design for RL with actor-critic design (ACD) in the framework of MPC, to realize near-optimal control of continuous-time nonlinear systems. Different from classic RL for continuous-time systems, the actor adopted is learned in discrete-time steps, while the critic evaluates the learned control policy continuously in the time domain. Moreover, we use soft barrier functions to deal with control constraints and the robustness of the actor-critic network is proven. A simulation example is considered to show the effectiveness of the proposed approach.

Keywords: Reinforcement learning, receding horizon strategy, sampled-data control, continuous-time, nonlinear system

1. INTRODUCTION

In real-world applications, many systems are characterized by nonlinear dynamics, and the system state evolves in continuous time. The control for such nonlinear continuous-time systems is non-trivial, especially for the variable constrained case. In principle, continuous-time model predictive control (MPC) can be used to solve the prescribed problem in view of its capability in dealing with constraints explicitly and of the well-developed theoretical developments, see Qin and Badgwell (2003); Mayne et al. (2000). In real industrial applications, the control systems are usually performed in discrete-time with control action being piece-wise constant in each sampling interval, which is the motivation of the digital control algorithm. In this scenario, the sampled-data based MPC and its robust version with resorting to tube-based robust control framework have been developed in Magni and Scattolini (2004); Farina and Scattolini (2012). In the underlying optimization problem, the control signal is regarded as a piece-wise constant decision variable, while the value function is minimized continuously with time to go. Compared with continuous-time MPC, sampled-data based MPC is more computationally efficient. Nevertheless, the control performance might still be hampered by high dimension and nonlinearity of the considered system.

As an alternative to MPC, reinforcement learning (RL) and approximate dynamic programming (ADP) are widely studied for solving nonlinear optimization problems. Among the notable contributions, algorithms with actor-critic designs (ACDs) usually utilize an actor network and a critic network for control policy and value function approximations. In this setting, the computational complexity of online optimization can be reduced especially for nonlinear optimization problems. For the past decades, many algorithms have been proposed to solve infinite-horizon optimization problems for continuous-time nonlinear systems, see in Vamvoudakis and Lewis (2010); Liu et al. (2014) and linear systems, in Jiang and Jiang (2012). In this respect, some works have been extended to solving finite-horizon optimization problems, see for instance Cheng et al. (2007); Heydari and Balakrishnan (2012); Li et al. (2015); Zhao et al. (2013). Different from that with infinite-horizon, the ACD with finite horizon results in an open-loop control problem, where the value function approximate network is time-dependent. To realize closed-loop control, recent efforts have been put on the developments with ACDs according to the receding horizon strategy like MPC. Among them, in Xu et al. (2018), a learning-based predictive controller has been proposed for perturbed discrete-time systems, where the process is learned in an iterative batch mode way. An ADP based functional nonlinear MPC has been developed in Dong et al. (2018) for nonlinear discrete-time systems with control saturation.

Motivated by prescribed works, a digital receding horizon learning controller (RH-LC) is proposed in this paper for continuous-time nonlinear systems. The main idea is to utilize the digital control realization of ACD in the receding horizon framework to realize near-optimal control. Different from classic ACDs for continuous-time systems, the actor adopted is learned in discrete-time steps, in the sense that the output of the actor is computed at each discrete-time instant and held to be piece-wise constant in each sampling interval, while the critic evaluates the learned policy continuously in time domain. Note that, different from the works in Xu et al. (2018); Dong et al. (2018) for discrete-time systems, the proposed approach aims at digital control for continuous-time systems, thus the techniques adopted are different. Moreover, we use soft logarithmic barrier functions to deal with control constraints. The robustness of the actor-critic network are proven. A simulation study for the regulation of a Van der Pol oscillator system is performed, which shows the potentiality of the proposed approach in control performance improvement compared with the classic infinite-

[1] The work was supported by the National Natural Science Foundation of China under Grant 61751311, 61825305, and the National Key R&D Program of China 2018YFB1305105.

horizon Heuristic Dynamic Programming (HDP), and in terms of computational complexity reduction than the sampled-data based MPC.

The rest of the paper is organized as follows. Section II introduces the control problem and preliminary works. In Section III the value function reconstruction for the proposed approach is first introduced, while the main idea of the proposed RH-LC is described in Section IV. Section V presents the simulation results on a Van der Pol oscillator. Conclusions are drawn in Section VI, while proofs to the main results are given in the Appendix.

Notation: We use $\otimes$ to denote the Kronecker product and use I_m to represent identity matrix with dimension m. For a square matrix X, we use $\lambda_{min}(X)$ and $\lambda_{max}(X)$ to denote the minimal and maximal eigenvalues respectively. For a given set of variables $s_i \in \mathbb{R}^{q_i}, i = 1,2,\ldots,M$, we define the form $(s_1, s_2, \cdots, s_{\mathrm{M}}) = [s_1^\top \ s_2^\top \ \cdots \ s_{\mathrm{M}}^\top]^\top \in \mathbb{R}^q$, where $q = \sum_{i=1}^p q_i$. Finally, a ball with radius ρ_{ε_i} and centered at the origin in the $\mathbb{R}^{dim}$ space is defined as follows

$$\mathscr{B}_{\rho_{\varepsilon_i}}(\bar{x}) := \{x \in \mathbb{R}^{dim} : ||x|| \leq \rho_{\varepsilon_i}\}.$$

2. CONTROL PROBLEM AND PRELIMINARY SOLUTIONS

Consider the following nonlinear continuous-time system described by

$$\dot{x}(t) = F(x(t),u(t)) = f(x(t)) + g(x(t))u(t), \quad x(0) = x_0 \quad (1)$$

where t is the continuous-time index, $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathscr{U} \subseteq \mathbb{R}^m$ are the state and input variables respectively. $\mathscr{U} = \{u | \sum_{i=1}^p a_i^\top u \leq b_i\}$, is a compact convex set containing the origin in the interior. Functions $f(x)$ and $g(x)$ are lipschitz continuous and differentiable in the domain $(x,u) \in \mathbb{R}^n \times \mathscr{U}$. Along the same line with Magni and Scattolini (2004), we assume the system is linearizable at the origin, that is

$$\begin{aligned} F(x,u) &= \frac{\partial F}{\partial x}|_{x,u=0}\delta x + \frac{\partial F}{\partial u}|_{x,u=0}\delta u + \rho(\delta x, \delta u) \\ &= A_c\delta x + B_c\delta u + \rho(\delta x, \delta u) \end{aligned}$$

where $\|\rho(\delta x, \delta u)\| \to 0$, as $\|(\delta x, \delta u)\| \to 0$.

The control scope concerned in this paper is to steer the state-input pair (x,u) to the origin according to an infinite-horizon integral cost described as

$$J_\infty(x_0, u(\cdot), 0) = \int_{\tau=0}^{\infty} l(x(\tau), u(\tau)) d\tau$$

where the stage cost $l(x,u) = x^\top Qx + u^\top Ru$, $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are symmetric positive-definite matrices. The control action requires to be computed and operated piece-wise constant in each sampling interval. Specifically, fix a sampling interval T and define a generic discrete-time index k corresponding to the continuous-time index t_k, such that $t_k = kT$. The control action to be applied in each interval is

$$u(t) = u_{\lfloor t/T \rfloor}, \text{ for } t \in [t_k, t_{k+1})$$

where and in the rest of this draft, we use the notation with a subscript index, e.g., z_k, to denote the variable in discrete time.

Assumption 1. For any finite initial condition $x_0 \in \mathbb{R}^n$, there exists a policy $u(0:\infty) \in \mathscr{U}^\infty$ such that $J_\infty < \infty$.

For latter use, we also define the discrete-time version of (1) as

$$x_{k+1} = f_d(x_k) + g_d(x_k)u_k$$

Definition 1. The Lipschitz constants L_f and L_g are such that

$$\|f_d(z) - f_d(y)\| \leq L_f\|z-y\| \quad (2a)$$
$$\|g_d(z) - g_d(y)\| \leq L_g\|z-y\| \quad (2b)$$

for any z, y.

To solve the prescribed problem, a MPC problem can be stated similar to the approach in Farina and Scattolini (2012) at any discrete-time k, i.e.,

$$\min_{\overrightarrow{u}_{k:k+N-1}} \quad J(x_k, t_k) \quad (3)$$

subject to:

1) the system dynamics (1)
2) the control constraints
$$u_j \in \mathscr{U} \ \text{ for } \ j = k, \cdots, k+N-1,$$
3) the terminal constaint
$$x(t_f) \in \mathscr{X}_f$$

where

$$J = \sum_{j=0}^{N-1} L(x_{k+j}, u_{k+j}) + J_f(x(t_f))$$

and where $L(x_j, u_j) = \int_{\tau=t_j}^{t_{j+1}} \|x(\tau)\|_Q^2 d\tau + T\|u_j\|_R^2$, $\forall j = k, \cdots, k+N-1$, the terminal cost $J_f(x(t_f)) = \|x(t_f)\|_P^2$. The symmetric matrix P is the solution of the Lyapunov equation

$$F^\top PF - P = -\bar{Q}$$

where K is chosen such that $F = A + BK$ is Schur stable, and where $A = A_d(T)$, $B = B_d(T)$, $A_d(\tau) = e^{A_c\tau}$, $B_d(\tau) = \int_{\eta=0}^{\tau} e^{A_c(\tau-\eta)}B_c d\eta$. The matrix $\bar{Q} = \int_{\tau=0}^{T} A_d(\tau)^\top Q A_d(\tau) d\tau + TK^\top RK$. According to Magni and Scattolini (2004), there exists a positive scalar α and sampling interval T such that $\mathscr{X}_f = \{x | \|x\|_P^2 \leq \alpha\}$ and, for any $x_k \in \mathscr{X}_f$ with system evolution constraint $x_{k+1} = Ax_k + Bu_k$, given $u_k = Kx_k \in \mathscr{U}$, it holds that

$$\|x_{k+1}\|_P^2 - \|x_k\|_P^2 \leq -\|x_k\|_{\bar{Q}}^2 \quad (4)$$

Assume the constrained problem (3) is feasible and denote $\overrightarrow{u}_{k:k+N-1|k}$ as an optimal solution. The first control action is held to be piece-wise constant and applied to system (1) as

$$u(t) = u_{\lfloor t/T \rfloor | k} \ \text{ for } t \in [t_k, t_{k+1}).$$

Then at the subsequent time t_{k+1}, the optimization problem (3) is computed repeatedly according to the receding horizon strategy. Note that, (3) might be computationally expensive for systems of high dimension and non-linearity, which leads to the MPC algorithm being not applicable for real-time control implementations with fast control commitment. Motivated by the above reasons, in that follows we propose an RH-LC algorithm to solve the prescribed nonlinear constrained optimization problem with sampled-data control orientation. We design a digital version of ACD in the receding horizon framework to achieve near-optimal control policy. To realize digital-based control, the weight associated with the actor is updated in each discrete-time instant and held to be piece-wise constant in each sampling interval, while the weight associated with the critic is learned in continuous-time so as to evaluate the learned control policy continuously. In this way, the control design can be simplified and the computational load can be reduced. Also, to realize constraint satisfaction, the hard control and terminal-state constraints are transformed into soft ones with logarithmic barrier functions and integrated into the value function to be optimized.

3. VALUE FUNCTION RECONSTRUCTION WITH BARRIER FUNCTIONS

As described previously, we use continuous and differentiable barrier functions to transform the hard control and terminal state constraints into soft ones and integrate them into the value function to be optimized. To this objective, we introduce following definitions about barrier functions in terms of ellipsoidal and polyhedral constraints.

Definition 2. For any variable $z \in \mathbb{Z}$, where $\mathbb{Z} = \{z = (z_1 \cdots, z_p) \; | a_i^\top z \leq b_i, \forall i = 1, \cdots, p\}$ is a polyhedron, the barrier function is defined as

$$\bar{B}(z) = \begin{cases} -\sum_{i=1}^{p} \log\big(b_i - a_i^\top z\big) & z \in Int(\mathbb{Z}) \\ +\infty \text{ otherwise.} \end{cases}$$

Definition 3. For any variable $z \in \mathbb{Z}$, where $\mathbb{Z} = \{z | z^\top Z z \leq 1\}$ is an ellipsoid, and where Z is a symmetric positive-definite matrix with suitable dimensions, the barrier function is defined as

$$\bar{B}(z) = \begin{cases} -\log\big(1 - z^\top Z z\big) & z \in Int(\mathbb{Z}) \\ +\infty \text{ otherwise.} \end{cases}$$

Note however that in the above definitions $\bar{B}(0)$ is not guaranteed to be zero, leading to the optimal value function $J^o\big(0,0\big) \neq 0$. For this reason, the pair (x,u) might not be able to converge into the neighbor of origin. To circumvent this problem, we introduce the following Lemma about barrier functions according to Wills and Heath (2004); Zhang et al. (2019).

Lemma 1.

(1) Let $B_c(z) = \bar{B}(z) - \bar{B}(0) - \bigtriangledown \bar{B}(0)^\top z$ be a gradient re-centered barrier function of $\bar{B}(z)$, then $B_c(z)$ is differentiable and convex for all $z \in Int(\mathbb{Z})$, and $B_c(0) = 0$;

(2) Let the relaxed barrier function for polyhedral constraint be defined as

$$B(z) = \begin{cases} B_c(z) & \bar{\nu} \geq \kappa \\ \gamma(z, \bar{\nu}) & \bar{\nu} < \kappa \end{cases} \tag{5}$$

where the small positive scalar κ is the relaxing factor, $\nu_i = b_i - a_i^\top z$, $i = 1, \cdots, p$, $\bar{\nu} = \min\{\nu_1, \cdots, \nu_p\}$, the function $\gamma(z, \bar{\nu}) : (-\infty, \kappa)$ is strictly monotone and differentiable such that $B(z)$ is differentiable at any z that $\bar{\nu} = \kappa$, and $\bigtriangledown^2 \gamma(z, \bar{\nu})$ is smaller than $\sum_{i=1}^{p} \|a_i\|_{\kappa^{-2}}$, then there exists a positive-definite matrix $H \geq \frac{1}{2} \sum_{i=1}^{p} \|a_i\|^2_{\kappa^{-2}}$ such that $B(z) \leq z^\top H z \leq B_{max}(z)$, where $B_{max}(z) = \max_{z \in \mathbb{Z}} z^\top H z$.

Now we reconstruct the value function with soft barrier functions as

$$J(x_k, t_k) = \sum_{i=0}^{N-1} \bar{L}(x_{k+i}, u_{k+i}) + J_f(x(t_f)) + \mu B_f(x(t_f)) \tag{6}$$

where $\bar{L}(x_{k+i}, u_{k+i}) = L(x_{k+i}, u_{k+i}) + \mu T B(u_{k+i})$, and where μ is a positive scalar, $B_f(z) = B_c(z)$.

The terminal penalty matrix P is modified as

$$F^\top P F - P = -\bar{Q} - \mu T K^\top H K, \tag{7}$$

where H is computed according to Lemma 1.2 with a presumed value of κ in (5) for $\mathscr{U}$.

4. DIGITAL RECEDING HORIZON LEARNING CONTROLLER

In this section, the main idea and implementing details of the proposed digital RH-LC is described.

4.1 Digital HJB equation

Under the assumption that f and g are Lipschitz continuous, the infinitesimal version of (6) in continuous-time is given as

$$\frac{\partial J(x(t),t)}{\partial t} + \frac{\partial J(x(t),t)}{\partial x(t)}^\top [f(x(t)) + g(x(t))u(t)] = -l(x(t), u(t)) - \mu B(u(t))$$

where $J(x(t_f), t_f) = J_f(x(t_f)) + \mu B_f(x(t_f))$.

At any sampling time instant t_k, the continuous-time HJB equation along the prediction horizon is rewritten as

$$\frac{\partial J(x(t),t)}{\partial t} + \frac{\partial J(x(t),t)}{\partial x(t)}^\top [f(x(t)) + g(x(t))u(t|t_k)] + l(x(t), u(t|t_k)) + \mu B(u(t|t_k)) = 0 \tag{8}$$

where $t \in [t_k, t_f]$. For any predictive time interval $[t_j, t_{j+1}] \subseteq [t_k, t_f]$, integrate both-sides of (8) with time yielding

$$\int_{\tau=t_j}^{t_{j+1}} \frac{\partial J(x(\tau),\tau)}{\partial t} + \frac{\partial J(x(\tau),\tau)}{\partial x(\tau)}^\top [f(x(\tau)) + g(x(\tau))u(\tau|t_k)] + l(x(\tau), u(\tau|t_k)) + \mu B(u(\tau|t_k)) d\tau = 0 \tag{9}$$

Recall that the control input is piece-wise constant, i.e., $u(t|t_k) = u_{\lfloor t/T \rfloor | k}$, for $t \in [t_j, t_{j+1})$. Taking derivative of both-side of (9) with respect to u leads to the optimal control policy satisfying

$$2TRu_{t_j|t_k} + \mu T \frac{\partial B(u_{t_j|t_k})}{\partial u_{t_j|t_k}} = -\int_{\tau=t_j}^{t_{j+1}} g(x(\tau))^\top \frac{\partial J(x(\tau),\tau)}{\partial x(\tau)} d\tau \tag{10}$$

It is difficult to solve the above HJB equation (8) analytically with (10) due to non-linearity of the adopted system. For this reason, we utilize the actor-critic structure to solve the control problem in each prediction horizon.

4.2 Actor-critic network approximations

Define the value function with a neural network with infinite nodes, that is

$$\begin{aligned} J(x(t),t) &= \sum_{i=1}^{r} \phi_i(x(t),t) w_i + \sum_{i=r+1}^{\infty} \phi_i(x(t),t) w_i \\ &= W_c^\top h_c(x(t),t) + \varepsilon_{c,N}(x(t),t) \end{aligned} \tag{11}$$

where $W_c = [w_1 \; \cdots \; w_r]^\top \in \mathbb{R}^r$ is the weighting vector, $h_c = [\phi_1 \; \cdots \; \phi_r]^\top \in \mathbb{R}^r$ is the vector of basis functions, and $\varepsilon_{c,N}(x(t),t)$ is the residual error. For practical reasons, to obtain near-optimal control policies, we first define the value function as the output of neural network with finite nodes:

$$\hat{J}(x(t),t) = \hat{W}_c^\top h_c(x(t),t) \tag{12}$$

where $\hat{W}_c \in \mathbb{R}^{r \times 1}$. Provided sufficient number of nodes, one can achieve $\hat{W}_c \to W_c$. From (12), we write

$$\frac{\partial \hat{J}(x(t),t)}{\partial t} = \bigtriangledown_t h_c(x(t),t)^\top \hat{W}_c \tag{13a}$$

$$\frac{\partial \hat{J}(x(t),t)}{\partial x(t)} = \bigtriangledown_x h_c(x(t),t)^\top \hat{W}_c \tag{13b}$$

where the notations ∇_t and ∇_x are the gradient with respect to time and x respectively. With (13), the HJB equation can be rewritten as

$$\sigma_1^\top \hat{W}_c + l(x(t), \hat{u}(t|t_k)) + \mu B(\hat{u}(t|t_k)) = e(t)$$

where $\sigma_1 = \sigma_t + \sigma_x$, $\sigma_x = \nabla_x h_c[f(x(t)) + g(x(t))\hat{u}(t|t_k)]$, $\hat{u}(t|t_k) = \hat{u}_{\lfloor t/T \rfloor|k}$, $\sigma_t = \nabla_t h_c$, the latter will be defined in (15). Aslo, denoting $e_f = \hat{W}_c^\top h_c(x(t_f), t_f) - J(x(t_f), t_f)$, at any time belonging in the prediction horizon $t \in [t_k, t_f)$, the objective to be minimized is defined as $E_c = \frac{1}{2}(e(t)^\top e(t) + e_f^\top e_f)$. Hence, the update rule of $\hat{W}_c$ can be compute:

$$\begin{aligned}\dot{\hat{W}}_c = -\alpha_1 \Big\{ &\frac{\sigma_1}{\left(\sigma_1^\top \sigma_1 + 1\right)^2}\left(\sigma_1^\top \hat{W}_c + l(x(t), u(t|t_k)) + \right. \\ &\left. \mu B(u(t|t_k))\right) + \frac{\sigma_2}{\left(\sigma_2^\top \sigma_2 + 1\right)^2}\left(\sigma_2^\top \hat{W}_c - J(x(t_f), t_f)\right) \Big\}\end{aligned} \tag{14}$$

where $\sigma_2 = h_c(x(t_f), t_f)$.

According to (10) and in view of (11), the optimal control policy at any discrete-time j can be rewritten as

$$\begin{aligned}&2RTu_{j|k} + \mu T \frac{\partial B(u_{j|k})}{\partial u_{j|k}} = \\ &-\int_{\tau=t_j}^{t_{j+1}} g(x(\tau))^\top \sum_{i=1}^{\infty} \nabla_x \phi_i(x(\tau), \tau) w_i d\tau\end{aligned}$$

As the estimated value function is defined in (12), it is reasonable to define the desired control policy as

$$\begin{aligned}&2RTu^d_{j|k} + \mu T \frac{\partial B(u^d_{j|k})}{\partial u^d_{j|k}} = \\ &-\int_{\tau=t_j}^{t_{j+1}} g(x(\tau))^\top \nabla_x h_c(x(\tau), \tau)^\top \hat{W}_c(\tau) d\tau =: G(u^d_{j|k})\end{aligned}$$

Similar to that with the value function, we define

$$u^d_{j|k} = W_{a,j}^\top h_a(x_j, j) + \varepsilon_a(x_j, j)$$

where $W_{a,j} \in \mathbb{R}^{r \times m}$, $h_a \in \mathbb{R}^r$, $\varepsilon_a(x_j, j)$ is the residual error. To approximate the desired control policy, we define the actor as

$$\hat{u}_{j|k} = \hat{W}_{a,j}^\top h_a(x_j, j) \tag{15}$$

Different from classic ACDs, in order to minimize the distance from $\hat{u}_{j|k}$ to $u^d_{j|k}$, we optimize the difference of $G(\hat{u}_{j|k}) =: 2RT\hat{u}_{j|k} + \mu T \frac{\partial B(\hat{u}_{j|k})}{\partial \hat{u}_{j|k}}$ with $G(u^d_{j|k})$ in Euclidean basis. To be specific, we define $E_{a,j} = 1/2(G(\hat{u}_{j|k}) - G(u^d_{j|k}))^\top (G(\hat{u}_{j|k}) - G(u^d_{j|k}))$ as the objective function. The update rule of $\hat{W}_a$ in discrete-time is given as

$$\hat{W}_{a,j+1} = \hat{W}_{a,j} - \alpha_2 \frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}} \tag{16}$$

To prove the robustness of the actor-critic structure, we require the following assumptions.

Assumption 2.

(1) The approximate errors of the neural networks in (11) are such that

$$\|\varepsilon_{c,N}\| \le \bar{\varepsilon}_c, \|\varepsilon_a\| \le \bar{\varepsilon}_a \tag{17a}$$

(2) We also need the mild assumption about $h_c(x,t)$, i.e.,

$$\|h_c(x,t)\| \le \varepsilon_h \tag{17b}$$

Assumption 3. The signal σ_1 and σ_2 are persistently exciting over the interval $[t_k, t_f]$, i.e., there exist positive scalars $\underline{\gamma_1} \le \overline{\gamma_1}$, $\underline{\gamma_2} \le \overline{\gamma_2}$, such that $\underline{\gamma_1} \le \int_{\tau=t_k}^{t_f} \sigma_1^\top \sigma_1 d\tau \le \overline{\gamma_1}$, and $\underline{\gamma_2} \le \int_{\tau=t_k}^{t_f} \sigma_1^\top \sigma_2 d\tau \le \overline{\gamma_2}$.

Theorem 1. Define the Lyapunov function as

$$V_j = V_{c,j} + V_{a,j}$$

where $V_{c,j} = \tilde{W}_{c,j}^\top \alpha_1^{-1} \tilde{W}_{c,j}$, $V_{a,j} = \mathrm{tr}(\tilde{W}_{a,j}^\top \alpha_2^{-1} \tilde{W}_{a,j})$, and where $\tilde{W}_{c,j} = W_c - \hat{W}_{c,j}$, $\tilde{W}_{c,j} = W_{a,j} - \hat{W}_{a,j}$. Consider the weighting update rule defined in (14) and (16), if T and α_2 are such that

$$c(T) < 0,$$

where $c(T)$ will be defined in (A.4), then V_j keeps decreasing along the discrete-time instant till $\tilde{W}_{c,j}$ and $\tilde{W}_{a,j}$ reach

$$\begin{aligned}\|\tilde{W}_{c,j}\| &\ge \frac{\sqrt{\mathrm{error_t}}}{\sqrt{T\lambda_{min}(\bar{\sigma}_1)}}, \\ \|\xi_{a,j}\| &\ge \frac{\sqrt{\mathrm{error_t}}}{\sqrt{\lambda_{min}(-c_1(T))}}.\end{aligned}$$

where $\xi_{a,j} = \tilde{W}_{a,j}^\top h_a(x_j, j)$, $\bar{\sigma}_2 > 0$ is such that $\bar{\sigma}_2 + \bar{\sigma}_1 = \bar{\sigma}$, $c_1(T) > c(T)$, for $\bar{\sigma}_1 > 0$, $\bar{\sigma}$ and $\mathrm{error_t}$ will be introduced in (A.1) and (A.6).

5. LEARNING CONTROL SIMULATION FOR A VAN DER POL OSCILLATOR

The proposed RH-LC is used to regulate a Van der Pol oscillator. The continuous-time system model is represented as

$$\begin{cases}\dot{x}_1 = x_2 \\ \dot{x}_2 = 1 - x_1^2 x_2 - x_1 + u\end{cases} \tag{18}$$

where the physical variables x_1 and x_2 represent the position and speed respectively, while u is the control force. The control constraint is to be verified, i.e., $-5m^2/s \le u \le 5m^2/s$.

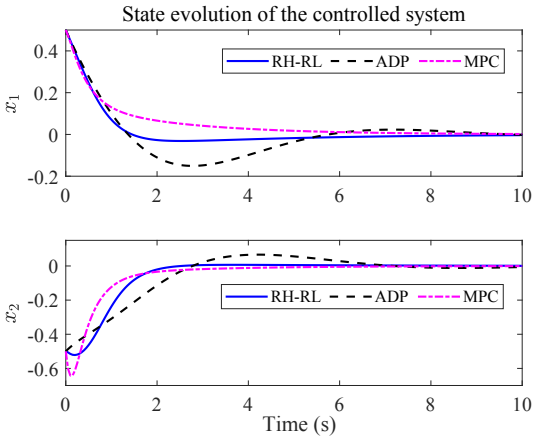

Fig. 1. State evolution of the controlled system.

To design the learning controller, first model (18) has been linearized at the origin, and the sampling interval has been chosen as $T = 0.05$s to compute the discrete-time linear model. The tuning parameters Q and R have been selected as $Q = I_2$, $R = 0.1$. The feedback gain matrix K has been obtained via solving a discrete-time LQ problem. With this choice, the parameter $\bar{Q}$ has been computed. μ has been selected as 0.001, and the relaxed scalar $\kappa = 0.1$. The terminal penalty has been obtained according to (7), i.e.,

$$P = \begin{bmatrix} 2.46 & -1.06 \\ -1.06 & 1.71 \end{bmatrix}$$

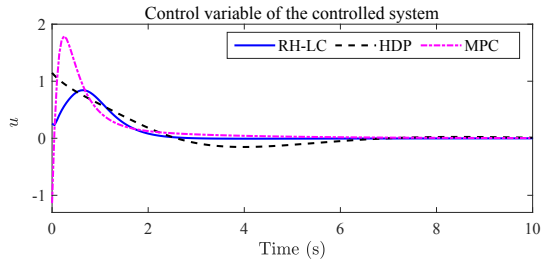

Fig. 2. The control variables of the controlled system.

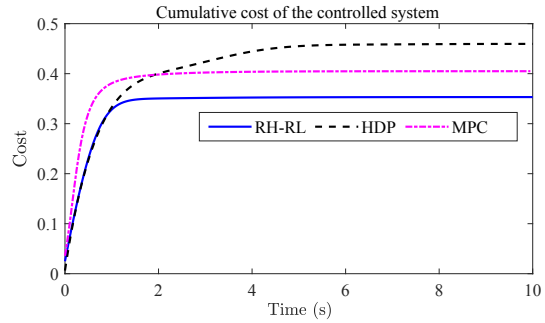

Fig. 3. The cumulative cost of the actor-critic with the RH-LC.

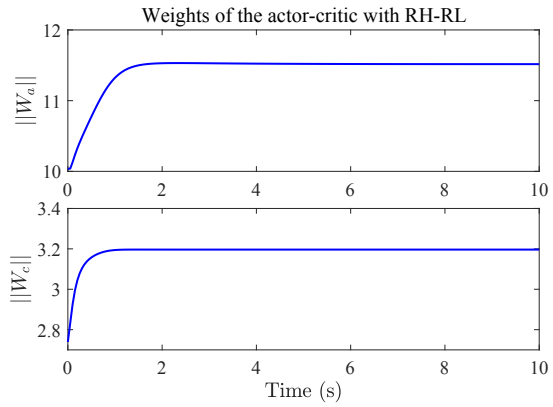

Fig. 4. The weights of the actor-critic with the RH-LC.

The parameter α has been obtained as 0.088. The learning rates are selected as $\alpha_1 = 100$, $\alpha_2 = 10$. Starting from initial condition $x_0 = (0.5, -0.5)$, the RH-LC has been implemented with $N_{sim} = 200$ steps and with random weight initialization. Also, the traditional HDP algorithm with infinite horizon in Vamvoudakis and Lewis (2010) and sampled-data based MPC in Magni and Scattolini (2004) have been adopted for comparisons. In the HDP and MPC, the control parameters, as well as the learning rates, are selected coincident with that in the proposed RH-LC. All the simulations have been performed in a Laptop with Intel Core i7-8550U CPU@1.8GHz installed with Win 10 operating system. All the three approaches have been tested in the Matlab 2019a environment, also the sampled-data based MPC has used an external CasADi toolbox described in Andersson et al. (2019). The simulation results have been presented in Fig. 1-4. It can be seen from Fig. 1-2 that the state and control of the RH-LC converge to the origin significantly faster than that with the HDP and slightly faster than that with the MPC. The cumulative cost $V = \int_{\tau=0}^{N_{sim}T} l(x(\tau), u(\tau))d\tau$ has been collected with time to go in Fig. 3, which shows that the regulating cost of the RH-LC is the lowest one among all the approaches. In Fig. 4, the convergence of the weights of the actor-critic with the RH-LC can be verified. Also, the average computational time has been collected in Table 1, which displays significant computational complexity reduction of the RH-LC than the MPC.

Table 1. Comparison in terms of computational time.

Algorithm	RH-LC	HDP	MPC
Average computational time (s)	0.079	0.041	0.227

6. CONCLUSIONS

In this paper, a digital RH-LC has been proposed for continuous-time nonlinear systems with control constraints. In each prediction horizon, the actor adopted has been learned in discrete-time and applied as piece-wise constant signal, while the critic has evaluated the value function in continuous-time. In the learning process, soft barrier functions are introduced for coping with control constraints. The robustness of the actor-critic network has been analyzed. The simulation tests on a Van der Pol oscillator show that, the RH-LC outperforms the HDP and the MPC, and exhibit an advantageous point in terms of computational complexity reduction compared to the data-based MPC. Future research will focus on the extension to robust control for systems with stochastic noise and with state and control possibility constraints.

Appendix A. PROOF OF THEOREM 1

For the selected Lyapunov function $V_j = V_{c,j} + V_{a,j}$, the difference of $V_{c,j}$ can be computed as

$$\Delta V_{c,j+1} = \int_{\tau=t_j}^{t_{j+1}} \tilde{W}_c^\top \alpha_1^{-1} \dot{\tilde{W}}_c d\tau$$

Note that,

$$\begin{aligned}
&\tilde{W}_c^\top \alpha_1^{-1} \dot{\tilde{W}}_c = \\
&= \tilde{W}_c^\top \{ \frac{\sigma_1}{\left(\sigma_1^T \sigma_1 + 1\right)^2} \left(\sigma_1^\top \hat{W}_c + l(x(t), u(t|t_k)) + \mu B(u(t|t_k)) \right) + \\
&\frac{\sigma_2}{\left(\sigma_2^\top \sigma_2 + 1\right)^2} \left(\sigma_2^\top \hat{W}_c - J(x(t_f), t_f) \right) \}
\end{aligned}$$

We recall the fact that $\sigma_1^\top W_c + l(x(t), \hat{u}(t|t_k)) + \mu B(\hat{u}(t|t_k)) = \varepsilon_H < \infty$. This requires $\hat{u}$ to be bounded, which can be fulfilled via performing saturation on the estimated value function. Hence, in view of (11)

$$\begin{aligned}
&\int_{\tau=t_j}^{t_{j+1}} \tilde{W}_c^\top \alpha_1^{-1} \dot{\tilde{W}}_c d\tau = \\
&= \int_{\tau=t_j}^{t_{j+1}} \tilde{W}_c^\top \{ \frac{\sigma_1}{\left(\sigma_1^\top \sigma_1 + 1\right)^2} \left(-\sigma_1^\top \tilde{W}_c + \varepsilon_H \right) + \\
&\frac{\sigma_2}{\left(\sigma_2^\top \sigma_2 + 1\right)^2} \left(-\sigma_2^\top \tilde{W}_c - \varepsilon_{c,N}(x(t_f), t_f) \right) \} d\tau \\
&= -\int_{\tau=t_j}^{t_{j+1}} (\tilde{W}_c^\top \bar{\sigma} \tilde{W}_c + \tilde{W}_c^\top b_\varepsilon) d\tau
\end{aligned} \tag{A.1}$$

where $\bar{\sigma} = \frac{\sigma_1 \sigma_1^\top}{\left(\sigma_1^\top \sigma_1 + 1\right)^2} + \frac{\sigma_2 \sigma_2^\top}{\left(\sigma_2^\top \sigma_2 + 1\right)^2}$, $b_\varepsilon = -\frac{\sigma_1}{\left(\sigma_1^\top \sigma_1 + 1\right)^2} \varepsilon_H + \frac{\sigma_2}{\left(\sigma_2^\top \sigma_2 + 1\right)^2} \varepsilon_{c,N}(x(t_f), t_f)$. The difference of the second term $V_a(j)$ is

$$\begin{aligned}&\Delta V_{a,j+1} = \\ &= \mathrm{tr}\left(\tilde{W}_{a,j+1}^\top \alpha_2^{-1}\tilde{W}_{a,j+1} - \tilde{W}_{a,j}^\top \alpha_2^{-1}\tilde{W}_{a,j}\right) \\ &= \mathrm{tr}\left(-2\tilde{W}_{a,j}^\top \frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}} + \alpha_2\left(\frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}}\right)^\top \frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}}\right)\end{aligned}$$

Consider that $B(u) = -\sum_{i=1}^{p} log(b_i - a_i^\top u) + b_i - \frac{a_i}{b_i}u$, then $\frac{\partial B(u)}{\partial u} = \sum_{i=1}^{p} \frac{a_i^\top}{b_i - a_i u} - \frac{a_i^\top}{b_i}$. Recalling the fact that $\frac{\partial a^\top X^\top b}{\partial X} = ba^\top$ where X is a matrix, a and b are two vectors, one has

$$\begin{aligned}&\frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}} = \\ &= \frac{< \partial G_{\hat{u}_{j|k}}, G_{\hat{u}_{j|k}} - G_{u_{j|k}^d} >}{\partial \hat{W}_{a,j}} \\ &= T^2(\bar{\mu}_1 h_a(x_j,j)\xi_{a,j}^\top R - \bar{\mu}_2 h_a(x_j,j)\xi_{a,j}^\top + \varepsilon_{a,t})\end{aligned}$$

where $\bar{\mu}_1 = 4 - 2\mu l + 2\mu$, $\bar{\mu}_2 = \mu^2 l\bar{l}$, $l = \sum_{i=1}^{p} a_i^\top a_i \hat{\delta}_i \delta_i$, $\bar{l} = \sum_{i=1}^{p} a_i^\top a_i \hat{\delta}_i^2$, $\delta = \frac{1}{(b_i - a_i^\top u)}$, $\hat{\delta}_i = \frac{1}{(b_i - a_i^\top \hat{u})^2}$, $\varepsilon_{a,t} = \bar{\mu}_1 h_a(x_j,j)\varepsilon_a(x_j,j)^\top R - \bar{\mu}_2 h_a(x_j,j)\varepsilon_a(x_j,j)^\top$.

Then, in view of the fact that $\mathrm{tr}(A^\top B) = \mathrm{tr}(AB^\top) = \mathrm{tr}(BA^\top) = \mathrm{tr}(B^\top A)$, it holds that

$$\mathrm{tr}\left(-2\tilde{W}_{a,j}^\top \frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}}\right) = \|\xi_{a,j}\|_{d_1(T)}^2 + \xi_{a,j}^\top d_1(T)\varepsilon_a(x_j,j) \quad \text{(A.2)}$$

where $d_1(T) = 2T^2(-\bar{\mu}_1 R + \bar{\mu}_2 I)$. Also, one can compute:

$$\begin{aligned}&\mathrm{tr}\left(\alpha_2\left(\frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}}\right)^\top \frac{\partial E_{a,j}}{\partial \hat{W}_{a,j}}\right) = \\ &\|\xi_{a,j}\|_{d_2(T)}^2 + 2\xi_{a,j}^\top d_2(T)\varepsilon_a(x_j,j) + \|\varepsilon_a(x_j,j)\|_{d_2(T)}^2\end{aligned} \quad \text{(A.3)}$$

where $d_2(T) = \alpha_2 T^4(\bar{\mu}_1^2 \bar{h}_a R^2 + \bar{\mu}_2^2 \bar{h}_a I - 2\bar{\mu}_1\bar{\mu}_2 R\bar{h}_a)$, $\bar{h}_a = h_a(x_j,j)^\top h_a(x_j,j)$. Hence, one promptly has

$$\begin{aligned}&\Delta V_{a,j+1} = \|\xi_{a,j}\|_{c(T)}^2 + \xi_{a,j}^\top \bar{c}(T)\varepsilon_a(x_j,j) + \|\varepsilon_a(x_j,j)\|_{d_2(T)}^2 \\ &\le \|\xi_a(x_j,j)\|_{c_1(T)}^2 + \|\varepsilon_a(x_j,j)\|_{d_2(T) - c_2(T)^{-1}\bar{c}(T)^2}^2\end{aligned} \quad \text{(A.4)}$$

where $c(T) = d_1(T) + d_2(T)$, $\bar{c}(T) = d_1(T) + 2d_2(T)$, $c_2(T) = c(T) - c_1(T)$.

Also note that, in view of the definition of $\bar{\sigma}_2$, one can prove

$$-\int_{\tau=t_j}^{t_{j+1}} (\tilde{W}_c^\top \bar{\sigma}_2 \tilde{W}_c + \tilde{W}_c^\top b_\varepsilon) d\tau \le \frac{\int_{\tau=t_j}^{t_{j+1}} b_\varepsilon^\top b_\varepsilon d\tau}{4\lambda_{min}(\bar{\sigma}_2)}$$

In view of Assumption 2-3, it holds that

$$\int_{\tau=t_j}^{t_{j+1}} b_\varepsilon^\top b_\varepsilon d\tau \le |\varepsilon_H|^2 \overline{\gamma_1} + \bar{\varepsilon}_c^2 \varepsilon_h^2 + 2\bar{\varepsilon}_c |\varepsilon_H| \overline{\gamma_2} := \bar{b}_\varepsilon < \infty, \quad \text{(A.5)}$$

Therefore ΔV_j satisfies

$$\Delta V_j \le -\int_{\tau=t_j}^{t_{j+1}} \tilde{W}_c^\top \bar{\sigma}_1 \tilde{W}_c d\tau + \|\xi_{a,j}\|_{c_1(T)}^2 + \mathrm{error_t} \quad \text{(A.6)}$$

where

$$\mathrm{error_t} = \frac{\bar{b}_\varepsilon}{4\lambda_{min}(\bar{\sigma}_2)} + \|\bar{\varepsilon}_a\|_{d_2(T) - c_2(T)^{-1}\bar{c}(T)^2}^2$$

In view of (A.6), one can conclude that $\Delta V_j < 0$ as long as the following conditions hold:

$$\|\tilde{W}_{c,j}\| \ge \frac{\sqrt{\mathrm{error_t}}}{\sqrt{T\lambda_{min}(\bar{\sigma}_1)}},$$

$$\|\xi_{a,j}\| \ge \frac{\sqrt{\mathrm{error_t}}}{\sqrt{\lambda_{min}(-c_1(T))}}.$$

□

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi:10.1007/s12532-018-0139-4.

Cheng, T., Lewis, F.L., and Abu-Khalaf, M. (2007). A neural network solution for fixed-final time optimal control of nonlinear systems. *Automatica*, 43(3), 482–490.

Dong, L., Yan, J., Yuan, X., He, H., and Sun, C. (2018). Functional nonlinear model predictive control based on adaptive dynamic programming. *IEEE Transactions on cybernetics*.

Farina, M. and Scattolini, R. (2012). Tube-based robust sampled-data mpc for linear continuous-time systems. *Automatica*, 48(7), 1473–1476.

Heydari, A. and Balakrishnan, S.N. (2012). Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics. *IEEE Transactions on Neural Networks and Learning Systems*, 24(1), 145–157.

Jiang, Y. and Jiang, Z.P. (2012). Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10), 2699–2704.

Li, C., Liu, D., and Li, H. (2015). Finite horizon optimal tracking control of partially unknown linear continuous-time systems using policy iteration. *IET Control Theory & Applications*, 9(12), 1791–1801.

Liu, D., Wang, D., Wang, F.Y., Li, H., and Yang, X. (2014). Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems. *IEEE Transactions on cybernetics*, 44(12), 2834–2847.

Magni, L. and Scattolini, R. (2004). Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49(6), 900–906.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, 11(7), 733–764.

Vamvoudakis, K.G. and Lewis, F.L. (2010). Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5), 878–888.

Wills, A.G. and Heath, W.P. (2004). Barrier function based model predictive control. *Automatica*, 40(8), 1415–1422.

Xu, X., Chen, H., Lian, C., and Li, D. (2018). Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances. *IEEE Transactions on neural networks and learning systems*, 29(99), 1–12.

Zhang, X., Liu, J., Xu, X., and Chen, H. (2019). Robust learning-based predictive control for constrained nonlinear systems. *arXiv preprint arXiv:1911.09827*.

Zhao, Q., Xu, H., Dierks, T., and Jagannathan, S. (2013). Finite-horizon neural network-based optimal control design for affine nonlinear continuous-time systems. In *The 2013 international joint conference on neural networks (IJCNN)*, 1–6. IEEE.