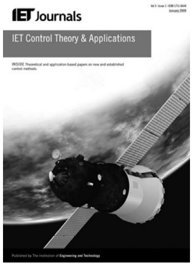
www.ietdl.org

Published in IET Control Theory and Applications
Received on 19th February 2014
Revised on 12th May 2014
Accepted on 6th June 2014
doi: 10.1049/iet-cta.2014.0186



ISSN 1751-8644

Online approximate optimal control for affine non-linear systems with unknown internal dynamics using adaptive dynamic programming

Xiong Yang, Derong Liu, Qinglai Wei

State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, People's Republic of China
E-mail: derong.liu@ia.ac.cn

Abstract: In this study, a novel online adaptive dynamic programming (ADP)-based algorithm is developed for solving the optimal control problem of affine non-linear continuous-time systems with unknown internal dynamics. The present algorithm employs an observer–critic architecture to approximate the Hamilton–Jacobi–Bellman equation. Two neural networks (NNs) are used in this architecture: an NN state observer is constructed to estimate the unknown system dynamics and a critic NN is designed to derive the optimal control instead of typical action–critic dual networks employed in traditional ADP algorithms. Based on the developed architecture, the observer NN and the critic NN are tuned simultaneously. Meanwhile, unlike existing tuning laws for the critic, the newly developed critic update rule not only ensures convergence of the critic to the optimal control but also guarantees stability of the closed-loop system. No initial stabilising control is required, and by using recorded and instantaneous data simultaneously for the adaptation of the critic, the restrictive persistence of excitation condition is relaxed. In addition, Lyapunov direct method is utilised to demonstrate the uniform ultimate boundedness of the weights of the observer NN and the critic NN. Finally, an example is provided to verify the effectiveness of the present approach.

1 Introduction

Optimal control problems for non-linear dynamical systems have been intensively studied during the past several decades [1–5]. A core challenge of deriving the solution for the non-linear optimal control problem is that it often falls to solve the Hamilton–Jacobi–Bellman (HJB) equation. It is well-known that the HJB equation is actually a partial differential equation, which is generally difficult to solve by analytical methods. To overcome the difficulty, Bellman introduced dynamic programming (DP) theory. The DP theory has been successfully applied to solve optimal control problems for many years. Nevertheless, a shortcoming of DP is that the computation grows exponentially with increase in the dimensionality of non-linear systems. Bellman coined this phenomenon 'the curse of dimensionality' [6].

For the sake of applying DP, Werbos proposes adaptive DP (ADP) algorithms [7, 8]. A distinct feature of the ADP method is that it employs neural networks (NNs) to derive approximate solutions of the HJB equation forward-in-time. There are several kinds of synonyms used for ADP, including 'adaptive dynamic programming' [9–13], 'approximate dynamic programming' [14–16], 'adaptive critic designs' [17] and 'neural dynamic programming' [18]. However, most of ADP approaches are either implemented offline by utilising iterative schemes or they require a priori knowledge of system dynamics. As for the real-world systems, these requirements are intractable to satisfy. Consequently, it gives rise to great challenges for implementing these algorithms in real-time control process. To address this issue, reinforcement learning (RL) methods are developed. RL is a class of approaches used in machine learning to revise the actions of an agent based on responses from its environment [19]. A general structure utilised to implement RL algorithm is the actor–critic architecture, where the actor performs actions by interacting with its environment, and the critic evaluates actions and offers feedback information to the actor, leading to the improvement in performance of the subsequent actor [20]. Compared with typical ADP methods, there is no prescribed behaviour or training model proposed to RL schemes. The feature of the RL method is often applied to adaptive optimal controller designs [21–25].

Recently, in order to overcome the iterative offline approach for real-time applications, several online RL-based algorithms were developed [26–28]. In [26], Vrabie and Lewis presented an online algorithm based on RL to solve the HJB equation of optimal control of non-linear continuous-time (CT) systems with unknown internal dynamics. By utilising the algorithm, the actor and the critic were sequently tuned and the solution of the HJB equation was successively approximated. It should be mentioned that, the system state needs to be reset at each iteration step and this brings about difficulties for stability analysis. After that, Vamvoudakis and Lewis [27] proposed a novel algorithm based on RL to synchronously tune the critic and the actor. However, the exact knowledge of CT non-linear systems is required in [27]. More recently, Bhasin *et al.* [28] developed a projection algorithm to obtain the optimal control of uncertain non-linear CT systems. Based on the algorithm, the actor, the critic and the identifier were all simultaneously tuned. Nevertheless, the use of the projection algorithm demands the selection of a predefined convex set so as to make the target NN weights remain in the set which is a challenge. Furthermore, it is worth pointing out that, the algorithms proposed in [26–28] all require an initial stabilising control. This requirement is restrictive and difficult to satisfy in practice. The reason is as follows: from a mathematical point of view, the initial stabilising policy is actually a suboptimal control. The suboptimal control is intractable to obtain since it is generally impossible to obtain analytical solutions of partial differential equations.

Lately, Dierks and Jagannathan [29] relaxed the requirement of initial stabilising control by using a single online approximator-based framework. After that, Zhang *et al.* [30] extended the work of [29] to derive the Nash equilibrium for CT non-zero-sum differential games. Meanwhile, based on the work of [29], Nodland *et al.* [31] developed an optimal adaptive output feedback control for an unmanned helicopter. Nevertheless, in order to guarantee exponential convergence of the NN weights to the actual optimal values, the persistence of excitation (PE) condition is required in [26–31]. It should be emphasised that, the PE condition is intractable to verify because of the presence of hidden-layers often involved. In addition, the PE signal is often derived by adding exploration noise. The inappropriate exploration noise might give rise to instability of the closed-loop system during implementing the algorithm, for there is no general structure to provide this kind of noise. To the best of our knowledge, there are rather few investigations on optimal control without employing the PE condition, especially the algorithms developed to derive optimal control without using both the PE condition and the initial stabilising control. This motivates our research.

In this paper, a novel online ADP-based algorithm is developed for solving the optimal control problem of affine non-linear CT systems with unknown internal dynamics. The present algorithm employs an observer–critic architecture to approximate the HJB equation. Two NNs are used in the architecture: an NN state observer is constructed to estimate unknown system dynamics and a critic NN is designed to derive the optimal control instead of typical action–critic dual networks employed in ADP algorithms. Based on the developed architecture, the observer NN and the critic NN are tuned simultaneously. Meanwhile, unlike existing tuning laws for the critic, the newly developed critic update rule not only ensures convergence of the critic to the optimal control but also guarantees stability of the closed-loop system. No initial stabilising control is required, and by using recorded and instantaneous data simultaneously for the adaptation of the critic, the restrictive PE condition is removed. Moreover, the weights of the observer NN and the critic NN are guaranteed to be uniformly ultimately bounded (UUB) through Lyapunov's direct method.

The main contributions of this paper are listed as follows:

1. To the best of authors' knowledge, it is the first time that an observer–critic architecture is developed to derive optimal control of partially uncertain non-linear CT systems without the requirement of both the PE condition and the initial stabilising control. Based on the constructed architecture, the observer NN and the critic NN can be tuned simultaneously.
2. In comparison with [26–31], a significant difference between these literature and the present paper is that, in our case, the restrictive PE condition is relaxed by using recorded and instantaneous data simultaneously to tune the critic. In addition, a clear advantage of the developed method in this paper as compared with [26–30] lies in that the requirement of the priori knowledge of system states is removed.
3. Unlike [31] using a linear-in-parameter (LP) NN observer, the developed observer utilises a non-LP (NLPs) NN. NLP NN is considered to be more powerful and accurate than LP NN employed to estimate the unknown non-linear system dynamics [32].

Furthermore, it should be mentioned that the present algorithm in this paper does not need value iteration and policy iteration. In other words, the developed algorithm does not share common feature with traditional RL methods. Hence, we consider the present algorithm to be a novel ADP approach.

This paper is organised as follows. Section 2 presents the problem statement and preliminaries. Section 3 constructs an NN state observer. Section 4 develops an online optimal neuro-controller. Section 5 conducts stability analysis and the performance of the closed-loop system. Section 6 provides simulation results to show the effectiveness of the proposed control scheme. Finally, Section 7 gives several concluding remarks.

Notations: $\mathbb{R}$ represents the set of the real numbers. $\mathbb{R}^m$ and $\mathbb{R}^{m\times n}$ represent the sets of the real m-vectors and the real $m\times n$ matrices, respectively. I_n represents the $n\times n$ identity matrix. T is the transposition symbol. Ω is a compact set of $\mathbb{R}^n$, $\mathrm{C}^m(\Omega)=\{f^{(m)}\in \mathrm{C}^1|f:\Omega\to\mathbb{R}^m\}$. When ξ is a vector, $\|\xi\|$ denotes the Euclidean norm of ξ. When A is a matrix, $\|A\|$ denotes the 2-norm of A.

2 Problem statement and preliminaries

Consider a non-linear CT system described by equations of the form

$$\begin{aligned}\dot{x}(t)&=f(x(t))+g(x(t))u(x(t))\\ y(t)&=Cx(t)\end{aligned}\tag{1}$$

where $x(t)\in\mathbb{R}^n$ is the state vector, $u(t)\in\mathbb{R}^m$ is the control input vector, $y(t)\in\mathbb{R}^l$ is the output vector, $f(x)\in\mathbb{R}^n$ is an unknown non-linear function and $g(x)\in\mathbb{R}^{n\times m}$ is a matrix of non-linear functions. It is assumed that $f(x)+g(x)u$ is Lipschitz continuous on a compact set $\Omega\subset\mathbb{R}^n$ containing the origin, such that the solution $x(t)$ of system (1) is unique for $\forall x_0\in\Omega$ and u, and $f(0)=0$. The states of system (1) are not available, only the system output $y(t)$ can be measured. For the sake of later analysis, the following assumptions are required.

Assumption 1: The control matrix $g(x)$ is known and bounded over the compact set Ω; that is, there exist positive constants g_m and g_M $(g_m<g_M)$ such that $g_m\le\|g(x)\|\le g_M$, for $\forall x\in\Omega$.

Assumption 2: System (1) is observable and system states are bounded in L_∞ [33, 34]. In addition, $C\in\mathbb{R}^{l\times n}$ $(l\le n)$ is a full row rank matrix; that is, $\operatorname{rank}(C)=l$.

In this paper, the value function for system (1) is given by

$$V(x(t)) = \int_t^\infty r(y(s), u(s))\mathrm{d}s \quad (s \geq t) \tag{2}$$

where $r(y, u) = y^\mathsf{T}Qy + u^\mathsf{T}Ru$, and Q and R are constant symmetric positive definite matrices.

Objective of control: The control goal in this paper is to obtain an online adaptive control not only stabilises system (1) but also minimises the value function (2), while ensuring that all the signals involved in the closed-loop system are UUB.

3 NN state observer

Since system states are unavailable and only the system output is known, we cannot directly derive the optimal control. To overcome the difficulty, an NN state observer is employed. According to [35], $\mathcal{F}(x) \in C^n(\Omega)(\mathcal{F}(x)$ is a non-linear function) can be represented by a two-layer feedforward NN as

$$\mathcal{F}(x) = W_\mathrm{o}^\mathsf{T}\sigma(V_\mathrm{o}^\mathsf{T}x) + \varepsilon_1(x) \tag{3}$$

where $\sigma(\cdot) \in \mathbb{R}^{N_1}$ is the activation function, $\varepsilon_1(x) \in \mathbb{R}^n$ is the NN function reconstruction error, $V_\mathrm{o} \in \mathbb{R}^{n\times N_1}$ and $W_\mathrm{o} \in \mathbb{R}^{N_1\times n}$ are the weights for the input layer to the hidden layer and the hidden layer to the output layer, respectively. The number of the hidden layer nodes is denoted by N_1. Activation functions for $\sigma(\cdot)$ are generally bounded, measurable, non-decreasing functions from the real numbers onto $[-1, 1]$ which include, for instance, hyperbolic tangent function $\sigma(x) = (e^x - e^{-x})/(e^x + e^{-x})$ and so on. Without loss of generality, in the state observer NN, we choose $\sigma(x) = \tanh(x)$.

From system (1), we have

$$\begin{aligned} \dot{x}(t) &= Ax + \mathcal{F}(x) + g(x)u \\ y(t) &= Cx(t) \end{aligned} \tag{4}$$

where $\mathcal{F}(x) = f(x) - Ax$, $A \in \mathbb{R}^{n\times n}$ is a Hurwitz matrix, and (C, A) is observable. By using (3), (4) can be rewritten as

$$\begin{aligned} \dot{x}(t) &= Ax + W_\mathrm{o}^\mathsf{T}\sigma(V_\mathrm{o}^\mathsf{T}x) + g(x)u + \varepsilon_1(x) \\ y(t) &= Cx(t) \end{aligned} \tag{5}$$

The NN state observer for system (1) is given by

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \hat{W}_\mathrm{o}^\mathsf{T}\sigma(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x}) + g(\hat{x})u + B(y - \hat{y}) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{6}$$

where $\hat{x}(t) \in \mathbb{R}^n$ and $\hat{y}(t) \in \mathbb{R}^l$ are the state and the output of the observer, respectively, $\hat{W}_\mathrm{o} \in \mathbb{R}^{N_1\times n}$ and $\hat{V}_\mathrm{o} \in \mathbb{R}^{n\times N_1}$ are estimated weights, and the observer gain $B \in \mathbb{R}^{n\times l}$ is chosen such that the matrix $A - BC$ is Hurwitz. In fact, such a matrix B does exist since (C, A) is observable.

Define the state and output estimation errors as $\tilde{x}(t) = x(t) - \hat{x}(t)$ and $\tilde{y}(t) = y(t) - \hat{y}(t)$, respectively. From (5) and (6), we can derive the observer error dynamics as

$$\begin{aligned} \dot{\tilde{x}}(t) &= A_c\tilde{x}(t) + \tilde{W}_\mathrm{o}^\mathsf{T}\sigma(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x}) + \delta(x) \\ \tilde{y}(t) &= C\tilde{x}(t) \end{aligned} \tag{7}$$

where $A_c = A - BC$, $\tilde{W}_\mathrm{o} = W_\mathrm{o} - \hat{W}_\mathrm{o}$ and $\delta(x) = W_\mathrm{o}^\mathsf{T}[\sigma(V_\mathrm{o}{}^\mathsf{T}x) - \sigma(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x})] + [g(x) - g(\hat{x})]u + \varepsilon_1(x)$.

Before presenting the stability analysis of the observer error $\tilde{x}(t)$, we provide some mild assumptions and facts. It should be mentioned that these assumptions are common techniques, which have been used in [36–38].

Assumption 3: The ideal observer NN weights W_o and V_o are bounded over Ω by known positive constants W_M and V_M, respectively. That is

$$\|W_\mathrm{o}\| \leq W_M, \quad \|V_\mathrm{o}\| \leq V_M$$

Assumption 4: The NN function reconstruction error $\varepsilon_1(x)$ is bounded over Ω as $\|\varepsilon_1(x)\| \leq \varepsilon_M$, where $\varepsilon_M > 0$.

Fact 1: The NN activation function is bounded over Ω; that is, there exists $\sigma_M > 0$ such that $\|\sigma(x)\| \leq \sigma_M$, for $\forall x \in \Omega$.

Fact 2: Since A_c is a Hurwitz matrix, there exists a positive-definite symmetric matrix $P \in \mathbb{R}^{n\times n}$ satisfying the Lyapunov equation

$$A_c^\mathsf{T}P + PA_c = -\theta I_n$$

where $\theta > 0$ is a design parameter.

Theorem 1: Let Assumptions 1–4 hold. If NN estimated weights $\hat{W}_\mathrm{o}$ and $\hat{V}_\mathrm{o}$ are updated as

$$\dot{\hat{W}}_\mathrm{o} = -l_1\sigma(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x})\tilde{y}^\mathsf{T}CA_c^{-1} - \kappa_1\|\tilde{y}\|\hat{W}_\mathrm{o} \tag{8}$$

$$\dot{\hat{V}}_\mathrm{o} = -l_2\mathrm{sgn}(\hat{x})\tilde{y}^\mathsf{T}CA_c^{-1}\hat{W}_\mathrm{o}^\mathsf{T}(I_{N_1} - \Phi(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x})) - \kappa_2\|\tilde{y}\|\hat{V}_\mathrm{o} \tag{9}$$

where $l_i > 0 (i = 1, 2)$ are design constants, $\kappa_i (i = 1, 2)$ satisfy

$$\kappa_1 > l_1\|CA_c^{-1}\|^2/4, \quad \kappa_2 > l_2 \tag{10}$$

$\Phi(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x}) = \mathrm{diag}\{\sigma_k^2(\hat{V}_{\mathrm{o}k}^\mathsf{T}\hat{x})\}(k = 1, \ldots, N_1)$, $\mathrm{sgn}(\hat{x}) = [\mathrm{sgn}(\hat{x}_1), \ldots, \mathrm{sgn}(\hat{x}_n)]^\mathsf{T}$ and $\mathrm{sgn}(\hat{x}_\iota)(\iota = 1, \ldots, n)$ are the sign function with respect to $\hat{x}_\iota$ [39]. Then, the NN state observer given in (6) can ensure that the observer error $\tilde{x}(t)$ converges to the compact set

$$\Omega_{\tilde{x}} = \left\{\tilde{x} : \|\tilde{x}\| \leq \frac{2\mathcal{B}}{\theta\|C\|\lambda_{\min}[(C^+)^\mathsf{T}C^+]}\right\} \tag{11}$$

where $\mathcal{B} > 0$ is a constant to be determined later [see (17)], C^+ is the Moore–Penrose pseudoinverse of the matrix C, and $\lambda_{\min}[(C^+)^\mathsf{T}C^+]$ is the minimum eigenvalue of the matrix $(C^+)^\mathsf{T}C^+$. In addition, the NN weight estimation errors $\tilde{W}_\mathrm{o}$ and $\tilde{V}_\mathrm{o} = V_\mathrm{o} - \hat{V}_\mathrm{o}$ are all guaranteed to be UUB.

Proof: Consider the Lyapunov function candidate

$$J(t) = J_1(t) + J_2(t) \tag{12}$$

where

$$\begin{aligned} J_1(t) &= \frac{1}{2}\tilde{x}^\mathsf{T}P\tilde{x} \\ J_2(t) &= \frac{1}{2}\mathrm{tr}(\tilde{W}_\mathrm{o}^\mathsf{T}l_1^{-1}\tilde{W}_\mathrm{o}) + \frac{1}{2}\mathrm{tr}(\tilde{V}_\mathrm{o}^\mathsf{T}l_2^{-1}\tilde{V}_\mathrm{o}) \end{aligned}$$

doi: 10.1049/iet-cta.2014.0186

Taking the time derivative of $J_1(t)$ and by using (7) and Facts 1 and 2, we have

$$\begin{aligned}\dot{J}_1(t) &= -\frac{\theta}{2}\tilde{x}^{\mathsf T}\tilde{x} + \tilde{x}^{\mathsf T}P(\tilde{W}_{\mathrm o}^{\mathsf T}\sigma(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x}) + \delta(x))\\ &= -\frac{\theta}{2}\tilde{y}^{\mathsf T}[(C^{+})^{\mathsf T}C^{+}]\tilde{y} + \tilde{y}^{\mathsf T}(C^{+})^{\mathsf T}P\\ &\quad\times(\tilde{W}_{\mathrm o}^{\mathsf T}\sigma(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x}) + \delta(x))\\ &\le -\frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]\|\tilde{y}\|^2\\ &\quad + \|\tilde{y}\|\|(C^{+})^{\mathsf T}P\|(\|\tilde{W}_{\mathrm o}\|\sigma_M + \delta_M)\end{aligned}\tag{13}$$

where δ_M is the upper bound of $\delta(x)$, that is, $\|\delta(x)\| \le \delta_M$. Actually, noticing that $u(x)$ is a continuous function defined on Ω, one can conclude that there exists $u_M > 0$ such that $\|u\| \le u_M$. Then, by Assumptions 1–4 and Fact 1, one can conclude that $\delta(x)$ in (7) is an upper bounded function.

Taking the time derivative of $J_2(t)$ and using weight update rules (8) and (9), we obtain

$$\begin{aligned}\dot{J}_2(t) &= \operatorname{tr}\left\{\tilde{W}_{\mathrm o}^{\mathsf T}\sigma(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x})\tilde{y}^{\mathsf T}CA_c^{-1} + \frac{\kappa_1}{l_1}\|\tilde{y}\|\tilde{W}_{\mathrm o}^{\mathsf T}(W_{\mathrm o} - \tilde{W}_{\mathrm o})\right\}\\ &\quad + \operatorname{tr}\left\{\tilde{V}_{\mathrm o}^{\mathsf T}\operatorname{sgn}(\hat{x})\tilde{y}^{\mathsf T}CA_c^{-1}(W_{\mathrm o} - \tilde{W}_{\mathrm o})^{\mathsf T}\right.\\ &\quad\left.\times(I_{N_1} - \Phi(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x})) + \frac{\kappa_2}{l_2}\|\tilde{y}\|\tilde{V}_{\mathrm o}^{\mathsf T}(V_{\mathrm o} - \tilde{V}_{\mathrm o})\right\}\end{aligned}\tag{14}$$

Note that $\operatorname{tr}(XY) = \operatorname{tr}(YX) = YX$, for $\forall X \in \mathbb{R}^{n\times 1}, Y \in \mathbb{R}^{1\times n}$ and $\operatorname{tr}[\tilde{Z}^{\mathsf T}(Z - \tilde{Z})] \le \|\tilde{Z}\|\|Z\| - \|\tilde{Z}\|^2$, for $\forall Z, \tilde{Z} \in \mathbb{R}^{m\times n}$. Then, (14) can be rewritten as

$$\begin{aligned}\dot{J}_2(t) &= \tilde{y}^{\mathsf T}CA_c^{-1}\tilde{W}_{\mathrm o}^{\mathsf T}\sigma(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x}) + \frac{\kappa_1}{l_1}\|\tilde{y}\|\operatorname{tr}(\tilde{W}_{\mathrm o}^{\mathsf T}(W_{\mathrm o} - \tilde{W}_{\mathrm o}))\\ &\quad + \tilde{y}^{\mathsf T}CA_c^{-1}(W_{\mathrm o} - \tilde{W}_{\mathrm o})^{\mathsf T}(I_{N_1} - \Phi(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x}))\tilde{V}_{\mathrm o}^{\mathsf T}\operatorname{sgn}(\hat{x})\\ &\quad + \frac{\kappa_2}{l_2}\|\tilde{y}\|\operatorname{tr}(\tilde{V}_{\mathrm o}^{\mathsf T}(V_{\mathrm o} - \tilde{V}_{\mathrm o}))\\ &\le \alpha\sigma_M\|\tilde{y}\|\|\tilde{W}_{\mathrm o}\| + \frac{\kappa_1}{l_1}\|\tilde{y}\|(W_M\|\tilde{W}_{\mathrm o}\| - \|\tilde{W}_{\mathrm o}\|^2)\\ &\quad + \alpha\|I_{N_1} - \Phi(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x})\|\|\tilde{y}\|(W_M + \|\tilde{W}_{\mathrm o}\|)\|\tilde{V}_{\mathrm o}\|\\ &\quad + \frac{\kappa_2}{l_2}\|\tilde{y}\|(V_M\|\tilde{V}_{\mathrm o}\| - \|\tilde{V}_{\mathrm o}\|^2)\end{aligned}\tag{15}$$

where $\alpha = \|CA_c^{-1}\|$. Combining (13) with (15) and noticing $\|I_{N_1} - \Phi(\hat{V}_{\mathrm o}^{\mathsf T}\hat{x})\| \le 1$, we obtain

$$\begin{aligned}\dot{J}(t) &\le -\frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]\|\tilde{y}\|^2 + \left\{\delta_M\|(C^{+})^{\mathsf T}P\|\right.\\ &\quad + \left((\|(C^{+})^{\mathsf T}P\| + \alpha)\sigma_M + \frac{\kappa_1}{l_1}W_M\right)\|\tilde{W}_{\mathrm o}\|\\ &\quad + \left(\alpha W_M + \frac{\kappa_2}{l_2}V_M\right)\|\tilde{V}_{\mathrm o}\| - \left(\frac{\kappa_1}{l_1} - \frac{\alpha^2}{4}\right)\|\tilde{W}_{\mathrm o}\|^2\\ &\quad\left. - \left(\frac{\kappa_2}{l_2} - 1\right)\|\tilde{V}_{\mathrm o}\|^2 - \left(\frac{\alpha}{2}\|\tilde{W}_{\mathrm o}\| - \|\tilde{V}_{\mathrm o}\|\right)^2\right\}\|\tilde{y}\|\\ &= -\frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]\|\tilde{y}\|^2 + \left\{\delta_M\|(C^{+})^{\mathsf T}P\|\right.\\ &\quad + \left(\frac{\kappa_1}{l_1} - \frac{\alpha^2}{4}\right)\beta_1^2 + \left(\frac{\kappa_2}{l_2} - 1\right)\beta_2^2\\ &\quad - \left(\frac{\kappa_1}{l_1} - \frac{\alpha^2}{4}\right)\|\tilde{W}_{\mathrm o} + \beta_1\|^2 - \left(\frac{\kappa_2}{l_2} - 1\right)\|\tilde{V}_{\mathrm o} + \beta_2\|^2\\ &\quad\left. - \left(\frac{\alpha}{2}\|\tilde{W}_{\mathrm o}\| - \|\tilde{V}_{\mathrm o}\|\right)^2\right\}\|\tilde{y}\|\end{aligned}\tag{16}$$

where

$$\beta_1 = \frac{2l_1(\alpha + \|(C^{+})^{\mathsf T}P\|)\sigma_M + 2\kappa_1 W_M}{\alpha^2 l_1 - 4\kappa_1}$$

$$\beta_2 = \frac{\alpha l_2 W_M + \kappa_2 V_M}{2(l_2 - \kappa_2)}$$

Combining (10) and (16), we derive

$$\begin{aligned}\dot{J}(t) &\le -\frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]\|\tilde{y}\|^2 + \left\{\delta_M\|(C^{+})^{\mathsf T}P\|\right.\\ &\quad\left. + \left(\frac{\kappa_1}{l_1} - \frac{\alpha^2}{4}\right)\beta_1^2 + \left(\frac{\kappa_2}{l_2} - 1\right)\beta_2^2\right\}\|\tilde{y}\|\\ &= -\left(\frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf T}C]\|\tilde{y}\| - \mathcal{B}\right)\|\tilde{y}\|\end{aligned}$$

where

$$\mathcal{B} = \delta_M\|(C^{+})^{\mathsf T}P\| + \left(\frac{\kappa_1}{l_1} - \frac{\alpha^2}{4}\right)\beta_1^2 + \left(\frac{\kappa_2}{l_2} - 1\right)\beta_2^2\tag{17}$$

Consequently, $\dot{J}(t)$ is negative as long as

$$\|\tilde{y}\| > \frac{2\mathcal{B}}{\theta\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]}\tag{18}$$

where $\mathcal{B}$ is defined in (17). Note that $\|\tilde{y}\| \le \|C\|\|\tilde{x}\|$. Then, (18) implies

$$\|\tilde{x}\| > \frac{2\mathcal{B}}{\theta\|C\|\lambda_{\min}[(C^{+})^{\mathsf T}C^{+}]}$$

That is, the observer error $\tilde{x}(t)$ converges to $\Omega_{\tilde{x}}$ defined as in (11). Meanwhile, according to the standard Lyapunov extension theorem [40], this verifies the uniform ultimate boundedness of the observer NN weight estimation errors $\tilde{W}_{\mathrm o}$ and $\tilde{V}_{\mathrm o}$. □

Remark 1: The first terms of (8) and (9) are both derived through the standard back-propagation algorithm, and the last terms of them are both employed to ensure the boundedness of parameter estimations. The size of $\Omega_{\tilde{x}}$ defined as in (11) can be kept sufficiently small by properly choosing parameters, for example, θ, κ_i, $l_i (i = 1, 2)$, such that higher accuracy of identification is guaranteed. Although (8) and (9) share similar feature as in [36], a significant difference between [36] and the present work is that, in our case, we do not use Taylor series in the process of identification. Owing to errors from using the Taylor series, our method is considered to be more accurate in estimating unknown system dynamics.

doi: 10.1049/iet-cta.2014.0186

Remark 2: By the knowledge of Linear Matrix [41, 42], we obtain $\mathrm{rank}(C) = \mathrm{rank}(C^+)$ and $\mathrm{rank}(C^+) = \mathrm{rank}[(C^+)^\mathsf{T} C^+]$. Hence, by using Assumption 2, we have $\mathrm{rank}[(C^+)^\mathsf{T} C^+] = \mathrm{rank}(C) = l$. Noticing that $(C^+)^\mathsf{T} C^+ \in \mathbb{R}^{l\times l}$ and $(C^+)^\mathsf{T} C^+$ is a symmetric semidefinite matrix, we can conclude that $(C^+)^\mathsf{T} C^+$ is positive definite. Therefore, $\lambda_{\min}[(C^+)^\mathsf{T} C^+] > 0$. This shows that the compact set $\Omega_{\tilde{x}}$ makes sense.

4 Online optimal neuro-controller design

This section is divided into two parts: In the first part, the HJB equation for system (1) is developed. Then, in the second part, an online NN-based optimal control scheme is presented.

4.1 HJB equation

In what follows we replace system (1) with (6), since system (1) can be approximated well by (6) outside of the compact set $\Omega_{\tilde{x}}$. Meanwhile, because of the unavailability of $x(t)$, we replace the actual system state $x(t)$ with the estimated state $\hat{x}(t)$. In this circumstance, system (1) can be represented as

$$\dot{\hat{x}}(t) = h(\hat{x}) + g(\hat{x})u \tag{19}$$

where $h(\hat{x}) = A\hat{x} + \hat{W}_\mathrm{o}^\mathsf{T}\sigma(\hat{V}_\mathrm{o}^\mathsf{T}\hat{x}) + B(y - C\hat{x})$. The value function (2) is rewritten as

$$V(\hat{x}(t)) = \int_t^\infty r(\hat{x}(s), u(s))\mathrm{d}s \tag{20}$$

where $r(\hat{x}, u) = Q_c(\hat{x}) + u^\mathsf{T} Ru$ with $Q_c(\hat{x}) = \hat{x}^\mathsf{T} C^\mathsf{T} QC\hat{x}$.

Let $\mathscr{A}(\Omega)$ be the set of admissible control [43]. If the control $u(\hat{x}) \in \mathscr{A}(\Omega)$ and the value function $V(\hat{x}) \in \mathrm{C}^1(\Omega)$, then we have

$$V_{\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u) + Q_c(\hat{x}) + u^\mathsf{T} Ru = 0$$

where $V_{\hat{x}} \in \mathbb{R}^n$ represents the partial derivative of $V(\hat{x})$ with respect to $\hat{x}$.

Define the Hamiltonian for the control $u(\hat{x})$ and the value function $V(\hat{x})$ as

$$H(\hat{x}, V_{\hat{x}}, u) = V_{\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u) + Q_c(\hat{x}) + u^\mathsf{T} Ru$$

Then, the optimal value $V^*(\hat{x})$ is obtained by solving the HJB equation

$$\min_{u(\hat{x})\in\mathscr{A}(\Omega)} H(\hat{x}, V_{\hat{x}}^*, u) = 0 \tag{21}$$

Consequently, the closed-form expression for optimal control can be derived as

$$u^*(x) = -\frac{1}{2}R^{-1}g^\mathsf{T}(\hat{x})V_{\hat{x}}^* \tag{22}$$

Substituting (22) into (21), we obtain the HJB equation as

$$(V_{\hat{x}}^*)^\mathsf{T} h(\hat{x}) + Q_c(\hat{x}) - \frac{1}{4}(V_x^*)^\mathsf{T} g(\hat{x})R^{-1}g^\mathsf{T}(\hat{x})V_{\hat{x}}^* = 0 \tag{23}$$

In this sense, one shall find that (23) is actually a partial differential equation with respect to $V_{\hat{x}}^*$, which is difficult to solve accurately by analytical methods. In order to confront the challenge, an online NN-based optimal control scheme is developed in the subsequent section. Before presenting the optimal control scheme, we provide the following required assumption.

Assumption 5: $L_1(\hat{x})$ is a continuously differentiable Lyapunov function candidate for system (19) and satisfies that $\dot{L}_1(\hat{x}) = L_{1\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u^*) < 0$ with $L_{1\hat{x}}$ the partial derivative of $L_1(\hat{x})$ with respect to $\hat{x}$. Meanwhile, there exists a positive definite matrix $\Lambda(\hat{x}) \in \mathbb{R}^{n\times n}$ defined on Ω such that

$$L_{1\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u^*) = -L_{1\hat{x}}^\mathsf{T}\Lambda(\hat{x})L_{1\hat{x}} \tag{24}$$

Remark 3: It should be emphasised that $h(\hat{x}) + g(\hat{x})u^*$ is often assumed to be bounded by a positive constant [27, 28], that is, there exists a constant $\rho > 0$ such that $\|h(\hat{x}) + g(\hat{x})u^*\| \le \rho$. To relax the condition, in this paper, $h(\hat{x}) + g(\hat{x})u^*$ is assumed to be bounded by a function with respect to x. Since $L_{1\hat{x}}$ is the function with respect to $\hat{x}$, without loss of generality, we assume that $\|h(\hat{x}) + g(\hat{x})u^*\| \le \varrho\|L_{1\hat{x}}\|(\varrho > 0)$. In this sense, one can derive that $\|L_{1\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u^*)\| \le \varrho\|L_{1\hat{x}}\|^2$. Noticing $L_{1\hat{x}}^\mathsf{T}(h(\hat{x}) + g(\hat{x})u^*) < 0$, one shall find that (24) defined as in Assumption 5 is reasonable. In addition, it is worth pointing out that $L_1(\hat{x})$ can be derived through proper selecting functions, such as polynomials.

4.2 Online neuro-optimal control scheme

In this subsection, an online optimal control scheme is constructed by using a unique critic NN. Owing to the universal approximation property of feedforward NNs [35], $V(\hat{x})$ in (20) can be represented as

$$V(\hat{x}) = W_c^\mathsf{T}\sigma(\vartheta_c^\mathsf{T}\hat{x}) + \varepsilon_2(\hat{x})$$

where $\vartheta_c \in \mathbb{R}^{n\times N}$ and $W_c \in \mathbb{R}^N$ denote the weights for the input layer to the hidden layer and the hidden layer to the output layer, respectively, and N is the number of the neurons. The activation function $\sigma(\vartheta_c^\mathsf{T}\hat{x})$ is written as $\sigma(\hat{x})$ for brevity, for ϑ_c is often initialised randomly and kept constant. $\sigma(\hat{x}) = [\sigma_1(\hat{x}), \sigma_2(\hat{x}), \ldots, \sigma_N(\hat{x})]^\mathsf{T} \in \mathbb{R}^N$ with $\sigma_i(\hat{x}) \in \mathrm{C}^1(\Omega)$, $\sigma_i(0) = 0$, and the set $\{\sigma_i(\hat{x})\}_1^N$ is chosen to be linearly independent, and $\varepsilon_2(\hat{x})$ is the NN function reconstruction error.

The derivative of $V(\hat{x})$ with respect to $\hat{x}$ is developed by

$$V_{\hat{x}} = \nabla\sigma^\mathsf{T}(\hat{x})W_c + \nabla\varepsilon_2 \tag{25}$$

where $\nabla\sigma(\hat{x}) = \partial\sigma(\hat{x})/\partial\hat{x}$ and $\nabla\sigma(0) = 0$.

By utilising (25), (22) can be represented as

$$u^*(\hat{x}) = -\frac{1}{2}R^{-1}g^\mathsf{T}(\hat{x})\nabla\sigma^\mathsf{T} W_c + \varepsilon_{u^*} \tag{26}$$

where $\varepsilon_{u^*} = -\frac{1}{2}R^{-1}g^\mathsf{T}(\hat{x})\nabla\varepsilon_2$. By the same token, (23) can be rewritten as

$$W_c^\mathsf{T}\nabla\sigma h(\hat{x}) + Q_c(\hat{x}) + \varepsilon_{\mathrm{HJB}} - \frac{1}{4}W_c^\mathsf{T}\nabla\sigma g(\hat{x})R^{-1}g^\mathsf{T}(\hat{x})\nabla\sigma^\mathsf{T} W_c = 0 \tag{27}$$

where $\varepsilon_{\mathrm{HJB}}$ is the residual error converging to zero when the number of NN nodes is large enough [25]; that is, there exists $\varepsilon_a > 0$ such that $\|\varepsilon_{\mathrm{HJB}}\| \le \varepsilon_a$.

In view of the unavailability of the ideal critic NN weight W_c, (26) cannot be implemented in real control process. Hence, we employ $\hat{V}(\hat{x})$ to approximate the value function in (20) as

$$\hat{V}(\hat{x}) = \hat{W}_c^{\mathsf{T}}\sigma(\hat{x}) \tag{28}$$

where $\hat{W}_c$ is the estimated weight of W_c. The weight estimation error for the critic NN is defined as

$$\tilde{W}_c = W_c - \hat{W}_c \tag{29}$$

By utilising (28), the estimates of (22) is given by

$$\hat{u}(\hat{x}) = -\frac{1}{2}R^{-1}g^{\mathsf{T}}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c \tag{30}$$

The approximated Hamiltonian is derived as

$$H(\hat{x}, \hat{W}_c) = \hat{W}_c^{\mathsf{T}}\nabla\sigma h(\hat{x}) + Q_c(\hat{x}) - \frac{1}{4}\hat{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c \triangleq e \tag{31}$$

where $\mathfrak{A}(\hat{x}) = g(\hat{x})R^{-1}g^{\mathsf{T}}(\hat{x})$.

Combining (26), (27) and (31), we have

$$e = -\tilde{W}_c^{\mathsf{T}}\nabla\sigma\left(\mathfrak{C}(\hat{x}) + \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\varepsilon_2\right) - \frac{1}{4}\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\tilde{W}_c - \varepsilon_{\mathrm{HJB}} \tag{32}$$

where $\mathfrak{C}(\hat{x}) = h(\hat{x}) + g(\hat{x})u^*$.

To derive the minimum value of e, it is desired to choose $\hat{W}_c$ to minimise the squared residual error $E = \frac{1}{2}e^{\mathsf{T}}e$. By using the gradient descent algorithm, the weight tuning law for the critic NN is often given by [24, 27, 28]

$$\dot{\hat{W}}_c = -\frac{\eta}{(1+\phi^{\mathsf{T}}\phi)^2}\frac{\partial E}{\partial\hat{W}_c} = -\eta\frac{\phi}{(1+\phi^{\mathsf{T}}\phi)^2}e \tag{33}$$

where $\phi = \nabla\sigma[h(\hat{x}) + g(\hat{x})\hat{u}]$, $\eta > 0$ is a design constant, and the term $(1+\phi^{\mathsf{T}}\phi)^2$ is employed for normalisation.

However, there exist two issues about the tuning rule (33):

1. Tuning the critic NN weights to minimise $E = \frac{1}{2}e^{\mathsf{T}}e$ alone cannot guarantee the stability of system (19) during the learning process of NNs.
2. The PE condition of $\phi/(1+\phi^{\mathsf{T}}\phi)$ is required to guarantee the weights of the critic NN exponential converge to the actual optimal values [24, 27–31]. Nevertheless, the PE condition is intractable to verify because of the presence of hidden-layers involving in the term $\phi/(1+\phi^{\mathsf{T}}\phi)$. In addition, the exploration noise is often added to obtain the PE signal, which might cause instability of the closed-loop system during implementing the algorithm.

To address above two issues, a novel weight update law for the critic NN is developed as

$$\begin{aligned}\dot{\hat{W}}_c = {} & -\eta\bar{\phi}\left(Y(\hat{x}) - \frac{1}{4}\hat{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right) \\ & - \eta\sum_{j=1}^{N}\bar{\phi}_{(j)}\left(Y(\hat{x}_{t_j}) - \frac{1}{4}\hat{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{A}(\hat{x}_{t_j})\nabla\sigma_{(j)}^{\mathsf{T}}\hat{W}_c\right) \\ & + \frac{\eta}{2}\Pi(\hat{x}, \hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}}\end{aligned} \tag{34}$$

where $\mathfrak{A}(\hat{x})$ is given in (31), $Y(\hat{x}) = \hat{W}_c^{\mathsf{T}}\nabla\sigma h(\hat{x}) + Q_c(\hat{x})$, $\bar{\phi} = \phi/m_s^2$ and $m_s = 1 + \phi^{\mathsf{T}}\phi$, $j \in \{1, \ldots, N\}$ denote the index of a stored data point $\hat{x}(t_j)$ (written as $\hat{x}_{t_j}$), $\bar{\phi}_{(j)} = \bar{\phi}(\hat{x}_{t_j})$, $m_{s_j} = 1 + \phi^{\mathsf{T}}(\hat{x}_{t_j})\phi(\hat{x}_{t_j})$, $\nabla\sigma_{(j)} = \nabla\sigma(\hat{x}_{t_j})$, $L_{1\hat{x}}$ is defined as in Assumption 5 and $\Pi(\hat{x}, \hat{u})$ is defined as

$$\Pi(\hat{x}, \hat{u}) = \begin{cases} 0, & \text{if } L_{1\hat{x}}^{\mathsf{T}}\left(h(\hat{x}) - \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right) < 0 \\ 1, & \text{otherwise}\end{cases} \tag{35}$$

Remark 4: Several notes about the weight tuning rule for the critic NN (34) are listed as follows:

(1) The first term in (34) shares the same feature with (33), which aims to minimise the objective function $E = \frac{1}{2}e^{\mathsf{T}}e$.
(2) The second term in (34) is utilised to relax the PE condition. If there is no second term in (34) and let $\hat{x} = 0$, then one can derive $\dot{\hat{W}}_c = 0$. In this sense, the approximated value function $\hat{V}(\hat{x})$ will no longer be updated. However, the optimal control might not be obtained at the finite time t_f which makes $\hat{x}(t_f) = 0$. To avoid this case from happening, the PE condition is usually employed [24, 27–31]. Interestingly, the second term in (34) can also avoid this pitfall as long as the set $\{\bar{\phi}_{(j)}\}_1^N$ is selected to be linearly independent. Now, we show this fact by contradiction as follows:

Suppose that when $\hat{x} = 0$, there exists $\dot{\hat{W}}_c = 0$. From (34), we obtain

$$\sum_{j=1}^{N}\bar{\phi}_{(j)}e_j = 0$$

where

$$e_j = Y(\hat{x}_{t_j}) - \frac{1}{4}\hat{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{A}(\hat{x}_{t_j})\nabla\sigma_{(j)}^{\mathsf{T}}\hat{W}_c$$

Since $\{\bar{\phi}_{(j)}\}_1^N$ is linearly independent, we can obtain $e_j = 0$ $(j = 1, \ldots, N)$. However, this case will not happen until the system state stays at the equilibrium point, for the points $\hat{x}_{t_j}$ $j \in \{1, \ldots, N\}$ are randomly selected. In other words, there at least exists a $j_0 \in \{1, \ldots, N\}$ such that $e_{j_0} \neq 0$ during the learning process of NNs. So, there is the contradiction. Hence, the second term in (34) guarantees that $\dot{\hat{W}}_c \neq 0$ during the learning process of NNs.
(3) The last term in (34) is employed to ensure the stability of the closed-loop system while the critic NN learns the optimal value. We denote the derivative of the Lyapunov function candidate for system (19) with the control input (30) as

$$\Theta = L_{1\hat{x}}^{\mathsf{T}}\left(h(\hat{x}) - \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right)$$

If the closed-loop system is unstable, then there exists $\Theta > 0$. In order to keep the closed-loop system stable (i.e. $\Theta <$

0), by using the gradient descent method, we have

$$\begin{aligned}-\eta\frac{\partial\Theta}{\partial\hat{W}_c} &= -\eta\frac{\partial\left[L_{1\hat{x}}^{\mathsf{T}}\left(h(\hat{x})-\frac{1}{2}\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right)\right]}{\partial\hat{W}_c}\\ &= \frac{\eta}{2}\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}} \end{aligned}\qquad(36)$$

Equation (36) shows the reason why we employ the last term of (34). In fact, observing the definition of $\Pi(\hat{x},\hat{u})$ given in (35), we find that if system (19) is stable (i.e. $\Theta<0$), then $\Pi(\hat{x},\hat{u})=0$ and the last term in (34) does not work. If system (19) is unstable, then $\Pi(\hat{x},\hat{u})=1$ and the last term in (34) is activated. Owing to the existence of the last term in (34), it makes no requirement of the initial stabilising control for system (19). The property shall be shown in the subsequent numerical simulation.

By Remark 4, we know that the set $\{\bar{\phi}_{(j)}\}_1^N$ should be linearly independent. Nevertheless, it is not an easy task to directly test this condition. Hence, we introduce a lemma as follows.

Lemma 1: If the set $\{\sigma(\hat{x}_{t_j})\}_1^N$ is linearly independent and $\hat{u}(\hat{x})$ stabilises system (19), then the following set

$$\{\nabla\sigma_{(j)}[h(\hat{x}_{t_j})+g(\hat{x}_{t_j})\hat{u}]\}_1^N$$

is also linearly independent.

Proof: Since the proof is similar to [43], we omit it here. □

Notice that $\bar{\phi}_{(j)}=\phi(\hat{x}_{t_j})/(1+\phi^{\mathsf{T}}(\hat{x}_{t_j})\phi(\hat{x}_{t_j}))^2$, where $\phi(\hat{x}_{t_j})=\nabla\sigma_{(j)}[h(\hat{x}_{t_j})+g(\hat{x}_{t_j})\hat{u}]$. By Lemma 1, we shall find that if $\{\sigma(\hat{x}_{t_j})\}_1^N$ is linearly independent, then $\{\bar{\phi}_{(j)}\}_1^N$ is also linearly independent. In other words, for ensuring the linear independence of $\{\bar{\phi}_{(j)}\}_1^N$, the following condition should be satisfied.

Condition 1: Let $\mathfrak{D}=[\sigma(\hat{x}_{t_1}),\ldots,\sigma(\hat{x}_{t_N})]\in\mathbb{R}^{N\times N}$ be the recorded data matrix. There exists sufficient large number of recorded data such that $\mathfrak{D}$ is non-singular, that is, $\det\mathfrak{D}\neq 0$.

Remark 5: Condition 1 can be satisfied by selecting and recording data during the learning process of NNs over a finite time interval. Compared with the PE condition, a clear advantage of Condition 1 is that it can be easily checked online [44]. In addition, Condition 1 makes full use of history data, which can improve the speed of the convergence of parameters. This fact will be shown in the numerical simulation.

By the definition of ϕ in (33) and using (26), we have

$$\phi=\nabla\sigma\left(\mathfrak{C}(\hat{x})+\frac{1}{2}\mathfrak{A}(\hat{x})\nabla\varepsilon_2\right)+\frac{1}{2}\nabla\sigma\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\tilde{W}_c\qquad(37)$$

where $\mathfrak{C}(\hat{x})$ is given in (32). From (29), (32), (34) and (37), we can derive

$$\begin{aligned}\dot{\tilde{W}}_c &= -\frac{\eta}{m_s^2}\left(\nabla\sigma\mathfrak{L}(\hat{x})+\frac{1}{2}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c\right)\\ &\quad\times\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x})+\frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c+\varepsilon_{\mathrm{HJB}}\right)\\ &\quad-\sum_{j=1}^{N}\frac{\eta}{m_{sj}^2}\left(\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j})+\frac{1}{2}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c\right)\\ &\quad\times\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j})+\frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c+\varepsilon_{\mathrm{HJB}}\right)\\ &\quad-\frac{\eta}{2}\Pi(\hat{x},\hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}}\end{aligned}\qquad(38)$$

where $\mathfrak{L}(\hat{x})=\mathfrak{C}(\hat{x})+\frac{1}{2}\mathfrak{A}(\hat{x})\nabla\varepsilon_2$, $\bar{\mathfrak{A}}(\hat{x})=\nabla\sigma\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}$ and $\bar{\mathfrak{A}}(\hat{x}_{t_j})=\nabla\sigma_{(j)}\mathfrak{A}(\hat{x}_{t_j})\nabla\sigma_{(j)}^{\mathsf{T}}$.

A general schematic programming of the proposed control algorithm is shown in Fig. 1.

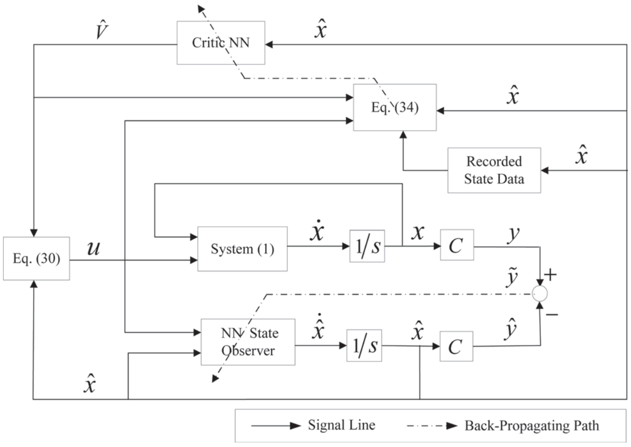


Fig. 1 *Developed control scheme for CT non-linear systems*

5 Stability analysis and the performance of the closed-loop system

In this section, we present our main results based on Lyapunov's direct method. Prior to demonstrating the main theorem, we provide another required assumption as follows:

Assumption 6: The derivative of the NN activation function $\sigma(\hat{x})$ with respect to $\hat{x}$ is bounded on Ω, that is, there exists $b_\sigma>0$ such that $\|\nabla\sigma(\hat{x})\|<b_\sigma$, $\forall x\in\Omega$. The derivative of the NN reconstruction error $\varepsilon_2(\hat{x})$ with respect to $\hat{x}$ is bounded on Ω, that is, there exists $\varepsilon_b>0$ such that $\|\nabla\varepsilon_2(\hat{x})\|<\varepsilon_b$, $\forall x\in\Omega$.

With Assumptions 1–6 and Facts 1 and 2, our main theorem is developed as follows:

Theorem 2: Given the input-affine dynamics described by (1) with associated HJB equation (23), let Assumptions 1–6 hold and take the control input for system (1) as in (30). Moreover, let weight update laws for the observer NN be (8) and (9), and let weight tuning rule for the critic NN be (34). Then, the state observer error $\tilde{x}(t)$, the NN weight estimation errors $\tilde{W}_{\mathrm{o}}$, $\tilde{V}_{\mathrm{o}}$ and $\tilde{W}_c$ are all UUB.

Proof: Consider the Lyapunov function candidate

$$L(t)=L_1(t)+L_2(t)+\frac{1}{2}\tilde{W}_c^{\mathsf{T}}\eta^{-1}\tilde{W}_c\qquad(39)$$

where $L_1(t)$ is defined as in Assumption 5, $L_2(t)=J(t)$ with $J(t)$ given in (12).

Taking the time derivative of (39) and by using Theorem 1, we derive

$$\begin{aligned}\dot{L}(t) &= \dot{L}_1(t) + \dot{L}_2(t) + \tilde{W}_c^{\mathsf{T}}\eta^{-1}\dot{\tilde{W}}_c\\ &\le L_{1\hat{x}}^{\mathsf{T}}\left(h(\hat{x}) - \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right)\\ &\quad - \frac{\theta}{2}\lambda_{\min}[(C^{+})^{\mathsf{T}}C^{+}]\|C\tilde{x}\|^2\\ &\quad + \mathcal{B}\|C\tilde{x}\| + \tilde{W}_c^{\mathsf{T}}\eta^{-1}\dot{\tilde{W}}_c \end{aligned}\tag{40}$$

where $\mathcal{B}$ is given in (17). By utilising (38), we derive the last term of (40) as

$$\tilde{W}_c^{\mathsf{T}}\eta^{-1}\dot{\tilde{W}}_c = \mathfrak{N}_1 + \mathfrak{N}_2 - \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\Pi(\hat{x},\hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}} \tag{41}$$

where

$$\begin{aligned}\mathfrak{N}_1 &= -\frac{1}{m_s^2}\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}) + \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c\right)\\ &\quad\times\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}) + \frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c + \varepsilon_{\mathrm{HJB}}\right)\\ \mathfrak{N}_2 &= -\sum_{j=1}^{N}\frac{1}{m_{s_j}^2}\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j}) + \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c\right)\\ &\quad\times\left(\tilde{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j}) + \frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c + \varepsilon_{\mathrm{HJB}}\right)\end{aligned}$$

Now, we consider the first term $\mathfrak{N}_1$ of (41). From $\mathfrak{N}_1$, we have

$$\begin{aligned}\mathfrak{N}_1 = -\frac{1}{m_s^2}\Bigg\{&(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}))^2 + \frac{1}{8}\left(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c\right)^2\\ &+ \frac{3}{4}(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}))(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c)\\ &+ \tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x})\varepsilon_{\mathrm{HJB}} + \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c\varepsilon_{\mathrm{HJB}}\Bigg\}\end{aligned}\tag{42}$$

Note that, for $\forall\ a, b \in \mathbb{R}$ and $\epsilon \neq 0$,

$$ab = \frac{1}{2}\left\{\left(\epsilon a + \frac{b}{\epsilon}\right)^2 - \left(\epsilon^2 a^2 + \frac{b^2}{\epsilon^2}\right)\right\} \tag{43}$$

Applying (43) to the last three terms of (42), we obtain

$$\begin{aligned}\mathfrak{N}_1 &= -\frac{1}{m_s^2}\Bigg\{\frac{1}{2}\left(3\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}) + \frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c\right)^2\\ &\quad + \frac{1}{2}(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}) + \varepsilon_{\mathrm{HJB}})^2 + \frac{1}{16}(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c)^2\\ &\quad + \frac{1}{2}\left(\frac{1}{4}\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c + 2\varepsilon_{\mathrm{HJB}}\right)^2\\ &\quad - 4(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}))^2 - \frac{5}{2}\varepsilon_{\mathrm{HJB}}^2\Bigg\}\\ &\le -\frac{1}{m_s^2}\left\{\frac{1}{16}(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c)^2 - 4(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}))^2 - \frac{5}{2}\varepsilon_{\mathrm{HJB}}^2\right\}\end{aligned}\tag{44}$$

Similarly, we can conclude

$$\begin{aligned}\mathfrak{N}_2 \le -\sum_{j=1}^{N}\frac{1}{m_{s_j}^2}\Bigg\{&\frac{1}{16}(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c)^2 - 4(\tilde{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j}))^2\\ &- \frac{5}{2}\varepsilon_{\mathrm{HJB}}^2\Bigg\}\end{aligned}\tag{45}$$

Substituting (44) and (45) into (41), and noticing that $1 \le m_s^2 \le 4$, $1 \le m_{s_j}^2 \le 4$, we obtain (see (46))

where $\mu_{\inf}(\mathcal{Y})$ denotes the lower bound of $\mathcal{Y}$ ($\mathcal{Y} = \bar{\mathfrak{A}}(\hat{x}_{t_j}), \bar{\mathfrak{A}}(\hat{x})$), and $\vartheta_{\sup}(\mathcal{Z})$ represents the upper bound of $\mathcal{Z}$ ($\mathcal{Z} = \mathfrak{L}(\hat{x}_{t_j}), \mathfrak{L}(\hat{x})$), and N is the number of neuron nodes in the hidden-layer.

Combining (40) and (46), we derive

$$\begin{aligned}\dot{L}(t) \le L_{1\hat{x}}^{\mathsf{T}}&\left(h(\hat{x}) - \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\sigma^{\mathsf{T}}\hat{W}_c\right)\\ &- \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\Pi(\hat{x},\hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}}\\ &- \frac{\mathfrak{T}_1}{64}\|\tilde{W}_c\|^4 + 4\mathfrak{T}_2\|\tilde{W}_c\|^2\\ &- \frac{\gamma}{2}(\|C\tilde{x}\| - \mathcal{B}/\gamma)^2 + \frac{\mathcal{B}^2}{2\gamma}\\ &+ \frac{5}{2}(N+1)\varepsilon_a^2\end{aligned}\tag{47}$$

$$\begin{aligned}\tilde{W}_c^{\mathsf{T}}\eta^{-1}\dot{\tilde{W}}_c &\le -\frac{1}{16}\left\{\sum_{j=1}^{N}\frac{1}{m_{s_j}^2}(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x}_{t_j})\tilde{W}_c)^2 + \frac{1}{m_s^2}(\tilde{W}_c^{\mathsf{T}}\bar{\mathfrak{A}}(\hat{x})\tilde{W}_c)^2\right\} + 4\left\{\sum_{j=1}^{N}\frac{1}{m_{s_j}^2}(\tilde{W}_c^{\mathsf{T}}\nabla\sigma_{(j)}\mathfrak{L}(\hat{x}_{t_j}))^2 + \frac{1}{m_s^2}(\tilde{W}_c^{\mathsf{T}}\nabla\sigma\mathfrak{L}(\hat{x}))^2\right\}\\ &\quad + \frac{5}{2}\left(\frac{1}{m_s^2} + \sum_{j=1}^{N}\frac{1}{m_{s_j}^2}\right)\varepsilon_{\mathrm{HJB}}^2 - \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\Pi(\hat{x},\hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}}\\ &\le -\frac{1}{64}\left\{\sum_{j=1}^{N}\mu_{\inf}^2(\bar{\mathfrak{A}}(\hat{x}_{t_j})) + \mu_{\inf}^2(\bar{\mathfrak{A}}(\hat{x}))\right\}\|\tilde{W}_c\|^4 + 4b_\sigma^2\left\{\sum_{j=1}^{N}\vartheta_{\sup}^2(\mathfrak{L}(\hat{x}_{t_j})) + \vartheta_{\sup}^2(\mathfrak{L}(\hat{x}))\right\}\|\tilde{W}_c\|^2\\ &\quad + \frac{5}{2}(N+1)\varepsilon_a^2 - \frac{1}{2}\tilde{W}_c^{\mathsf{T}}\Pi(\hat{x},\hat{u})\nabla\sigma\mathfrak{A}(\hat{x})L_{1\hat{x}}\end{aligned}\tag{46}$$

doi: 10.1049/iet-cta.2014.0186

where

$$\mathfrak{T}_1 = \mu_{\inf}^2(\bar{\mathfrak{A}}(\hat{x})) + \sum_{j=1}^{N} \mu_{\inf}^2(\bar{\mathfrak{A}}(\hat{x}_{t_j}))$$

$$\mathfrak{T}_2 = b_\sigma^2 \vartheta_{\sup}^2(\mathfrak{L}(\hat{x})) + b_\sigma^2 \sum_{j=1}^{N} \vartheta_{\sup}^2(\mathfrak{L}(\hat{x}_{t_j}))$$

$$\gamma = \theta\lambda_{\min}\big[(C^{+})^{\mathsf{T}} C^{+}\big]$$

In view of the definition of $\Pi(\hat{x}, \hat{u})$ in (35), we divide (47) into the following two cases for discussion.

Case 1: $\Pi(\hat{x}, \hat{u}) = 0$. By the definition of $\Pi(\hat{x}, \hat{u})$ in (35), we can derive that the first term in (47) is negative under this circumstance. Observing that $L_{1\hat{x}}^{\mathsf{T}}\dot{\hat{x}} < 0$, by using the Archimedean property of $\mathbb{R}$ [39], we can conclude that there exists a constant $\tau > 0$ such that $L_{1\hat{x}}^{\mathsf{T}}\dot{\hat{x}} < -\|L_{1\hat{x}}\|\tau \le 0$. Then, (47) becomes

$$\begin{aligned} \dot{L}(t) \le {} & -\tau\|L_{1\hat{x}}\| - \frac{\gamma}{2}(\|C\tilde{x}\| - \mathcal{B}/\gamma)^2 \\ & - \frac{\mathfrak{T}_1}{64}\left(\|\tilde{W}_c\|^2 - \frac{128\mathfrak{T}_2}{\mathfrak{T}_1}\right)^2 + \frac{256\mathfrak{T}_2^2}{\mathfrak{T}_1} \\ & + \frac{1}{2\gamma}[\mathcal{B}^2 + 5\gamma(N+1)\varepsilon_a^2] \end{aligned} \tag{48}$$

Accordingly, (48) yields $\dot{L}(t) < 0$ as long as one of the following conditions holds

$$\|L_{1\hat{x}}\| > \frac{256\mathfrak{T}_2^2}{\tau\mathfrak{T}_1} + \frac{\mathcal{B}^2 + 5\gamma(N+1)\varepsilon_a^2}{2\tau\gamma} \tag{49}$$

or

$$\|\tilde{x}\| > \frac{1}{\|C\|}\sqrt{\frac{512\mathfrak{T}_2^2}{\gamma\mathfrak{T}_1} + \frac{\mathcal{B}^2 + 5\gamma(N+1)\varepsilon_a^2}{\gamma^2}} + \frac{\mathcal{B}}{\gamma\|C\|} \tag{50}$$

or

$$\|\tilde{W}_c\| > 2\sqrt{\frac{32\mathfrak{T}_2}{\mathfrak{T}_1} + \frac{\sqrt{2\mathfrak{T}_1[\mathcal{B}^2/\gamma + 5(N+1)\varepsilon_a^2] + 1024\mathfrak{T}_2^2}}{\mathfrak{T}_1}} \tag{51}$$

Case 2: $\Pi(\hat{x}, \hat{u}) = 1$. By the definition of $\Pi(\hat{x}, \hat{u})$ in (35), we find that, in this case, the first term in (47) is non-negative which implies that the control (30) may not stabilise system (19). Then, (47) becomes

$$\begin{aligned} \dot{L}(t) \le {} & L_{1\hat{x}}^{\mathsf{T}}\left(\mathfrak{C}(\hat{x}) + \frac{1}{2}\mathfrak{A}(\hat{x})\nabla\varepsilon_2\right) \\ & - \frac{\gamma}{2}(\|C\tilde{x}\| - \mathcal{B}/\gamma)^2 + \frac{\mathcal{B}^2}{2\gamma} \\ & - \frac{\mathfrak{T}_1}{64}\left(\|\tilde{W}_c\|^2 - \frac{128\mathfrak{T}_2}{\mathfrak{T}_1}\right)^2 \\ & + \frac{256\mathfrak{T}_2^2}{\mathfrak{T}_1} + \frac{5}{2}(N+1)\varepsilon_a^2 \end{aligned} \tag{52}$$

where $\mathfrak{C}(\hat{x})$ is given in (32). By using Assumptions 5 and 6, (52) can be rewritten as

$$\begin{aligned} \dot{L}(t) \le {} & -\lambda_{\min}(\Lambda(\hat{x}))\left(\|L_{1\hat{x}}\| - \frac{\varepsilon_b\vartheta_{\sup}(\mathfrak{A}(\hat{x}))}{4\lambda_{\min}(\Lambda(\hat{x}))}\right)^2 \\ & - \frac{\gamma}{2}(\|C\tilde{x}\| - \mathcal{B}/\gamma)^2 - \frac{\mathfrak{T}_1}{64}\left(\|\tilde{W}_c\|^2 - \frac{128\mathfrak{T}_2}{\mathfrak{T}_1}\right)^2 \\ & + \frac{\varepsilon_b\vartheta_{\sup}(\mathfrak{A}(\hat{x}))}{16\lambda_{\min}(\Lambda(\hat{x}))} + \frac{256\mathfrak{T}_2^2}{\mathfrak{T}_1} \\ & + \frac{1}{2\gamma}[\mathcal{B}^2 + 5\gamma(N+1)\varepsilon_a^2] \end{aligned} \tag{53}$$

where $\lambda_{\min}(\Lambda(\hat{x}))$ represents the minimum eigenvalue of $\Lambda(\hat{x})$, $\vartheta_{\sup}(\cdot)$ is defined as in (46).

Therefore, (53) implies $\dot{L}(t) < 0$ as long as one of the following conditions holds

$$\|L_{1\hat{x}}\| > \frac{\varepsilon_b\vartheta_{\sup}(\mathfrak{A}(\hat{x}))}{4\lambda_{\min}(\Lambda(\hat{x}))} + \sqrt{\frac{d}{\lambda_{\min}(\Lambda(\hat{x}))}} \tag{54}$$

or

$$\|\tilde{x}\| > \frac{1}{\|C\|}\sqrt{\frac{2d}{\gamma}} + \frac{\mathcal{B}}{\gamma\|C\|} \tag{55}$$

or

$$\|\tilde{W}_c\| > 2\sqrt{\frac{32\mathfrak{T}_2}{\mathfrak{T}_1} + 2\sqrt{\frac{d}{\mathfrak{T}_1}}} \tag{56}$$

where

$$d = \frac{\varepsilon_b\vartheta_{\sup}(\mathfrak{A}(\hat{x}))}{16\lambda_{\min}(\Lambda(\hat{x}))} + \frac{256\mathfrak{T}_2^2}{\mathfrak{T}_1} + \frac{1}{2\gamma}[\mathcal{B}^2 + 5\gamma(N+1)\varepsilon_a^2]$$

Combining Cases 1 and 2 and by using the standard Lyapunov extension theorem [40], one can conclude that the state observer error $\tilde{x}(t)$, NN weight estimation errors $\tilde{W}_{\mathrm{o}}$, $\tilde{V}_{\mathrm{o}}$ and $\tilde{W}_c$ are UUB. □

Remark 6: It is worth pointing out that the uniform ultimate boundedness of $\tilde{W}_{\mathrm{o}}$ and $\tilde{V}_{\mathrm{o}}$ is obtained as follows: inequalities (49)–(51) [or (54)–(56)] guarantee $\dot{L}(t) < 0$. Then, we can conclude that $L(t)$ is the strictly decreasing function with respect to t $(t \ge 0)$. Hence, we can derive $L(t) < L(0)$, where $L(0)$ is a bounded positive constant. By using $L(t)$ defined as in (39), we have that $\frac{1}{2}\mathrm{tr}\big(\tilde{W}_{\mathrm{o}}^{\mathsf{T}} l_1^{-1}\tilde{W}_{\mathrm{o}}\big) + \frac{1}{2}\mathrm{tr}(\tilde{V}_{\mathrm{o}}^{\mathsf{T}} l_2^{-1}\tilde{V}_{\mathrm{o}}) < L(0)$. By using the definition of Frobenius norm and the equivalence of norms [42], we can derive that $\|\tilde{W}_{\mathrm{o}}\|$ and $\|\tilde{V}_{\mathrm{o}}\|$ are bounded. This verifies that $\tilde{W}_{\mathrm{o}}$ and $\tilde{V}_{\mathrm{o}}$ are UUB.

6 Simulation results

In this section, an example is provided to illustrate the effectiveness of the developed theoretical results.

Consider the input-affine non-linear CT systems described by

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= Cx\end{aligned} \tag{57}$$

where

$$f(x) = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2 + 0.5x_2[\cos(2x_1) + 2]^2 \end{bmatrix}$$
$$g(x) = \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The value function is given in (2), where Q and R are chosen as identity matrices of approximate dimensions. The prior knowledge of system states is assumed to be unavailable, and only the output $y(t)$ is measurable in system (57). To obtain the knowledge of system dynamics, an NN state observer given in (6) is employed. The gains for the observer NN are selected as

$$A = [-1 \;\; 1; -0.5 \;\; -0.5], \quad B = [1 \;\; 0; -0.5 \;\; 0]$$
$$l_1 = 20, \quad l_2 = 10, \quad \kappa_1 = 6.1, \quad \kappa_2 = 15, \quad N_1 = 8$$

and the gain for the critic NN is selected to be $\eta = 2.5$. The activation function for the critic NN is chosen with $N = 3$ neurons as $\sigma(x) = [x_1^2 \;\; x_2^2 \;\; x_1x_2]^{\mathrm{T}}$ and the critic NN weight is denoted as $\hat{W}_c = [W_c^1 \;\; W_c^2 \;\; W_c^3]^{\mathrm{T}}$.

Remark 7: It is significant to emphasise that, the number of neurons required for any particular application is still an open problem. Selecting the proper neurons for NNs is more of an art than science [45]. In this example, the number of neurons is obtained by computer simulations. We find that selecting eight neurons in the hidden layer for the observer NN can lead to satisfactory simulation results. Meanwhile, in order to compare our algorithm with the algorithms proposed in [27, 28], we choose three neurons in the hidden layer for the critic NN, and the simulation results are satisfied.

The initial weights $\hat{W}_o$ and $\hat{V}_o$ for the observer NN are selected randomly within an interval of $[-10, 10]$ and $[-5, 5]$, respectively. Meanwhile, the initial weights for the critic NN are chosen to be zeros, and the initial system state is selected to be $x_0 = [3.5 \;\; -3.5]^{\mathrm{T}}$. In this case, the initial control cannot stabilise system (57). In other words, no initial stabilising control is required for implementing the algorithm. In addition, by using the method proposed in [44, 46], the recorded data can be easily made qualified for Condition 1.

The computer simulation results are presented in Figs. 2–9. Figs. 2 and 3 show the trajectories of system state $x_1(t)$ and observed state $\hat{x}_1(t)$, and the trajectories of system state $x_2(t)$ and observed state $\hat{x}_2(t)$, respectively. Fig. 4 illustrates the performance of the NN state observer errors $\tilde{x}_1(t)$ and $\tilde{x}_2(t)$. Fig. 5 presents the 2-norm of the weights of the observer NN $\|\hat{W}_o\|$ and $\|\hat{V}_o\|$. Fig. 6 shows the performance of convergence of the critic NN weights. Fig. 7 presents the control u. Fig. 8 illustrates system states without considering the the third term in (34).

To make comparison with [29], we employ Fig. 9 to show the system states with the algorithm proposed in [29]. It should be mentioned that the PE condition is necessary in [29]. To guarantee the PE qualitatively, we add a small exploratory signal $n(t) = \sin^5(t)\cos(t) + \sin^5(2t)\cos(0.1t)$ to the control u for the first 9 s. In other words, Fig. 9 is obtained based on the exploratory signal $n(t)$. In addition, it should be pointed out that, by using the methods proposed in [27, 28] and employing the same exploratory signal $n(t)$, one

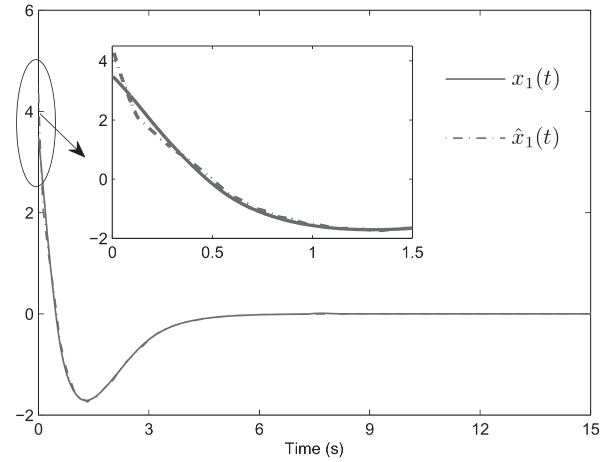


Fig. 2 *Trajectories of real state $x_1(t)$ and observed state $\hat{x}_1(t)$*

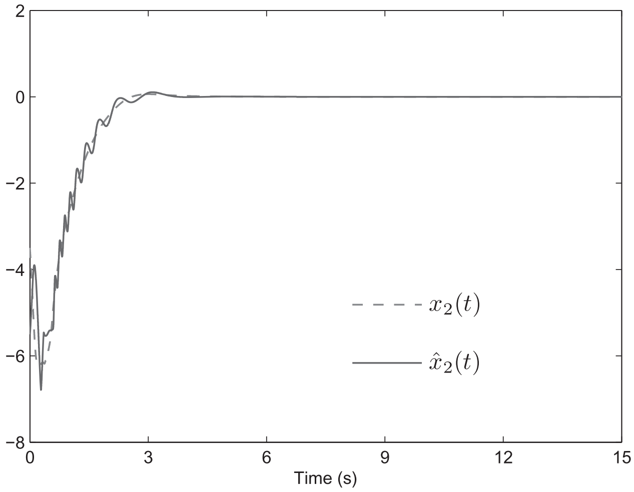


Fig. 3 *Trajectories of real state $x_2(t)$ and observed state $\hat{x}_2(t)$*

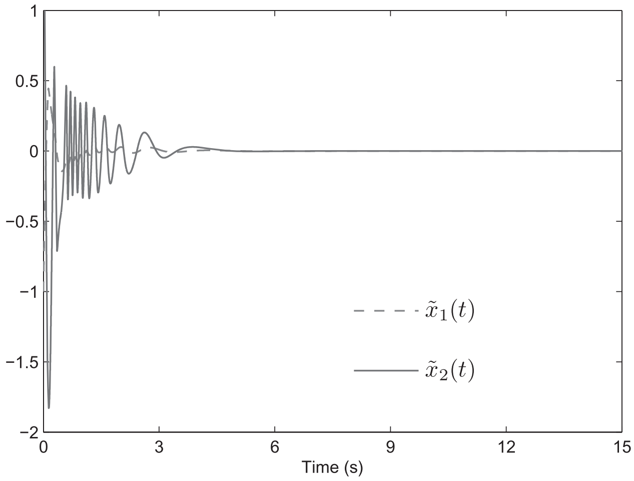


Fig. 4 *NN observer errors $\tilde{x}_1(t)$ and $\tilde{x}_2(t)$*

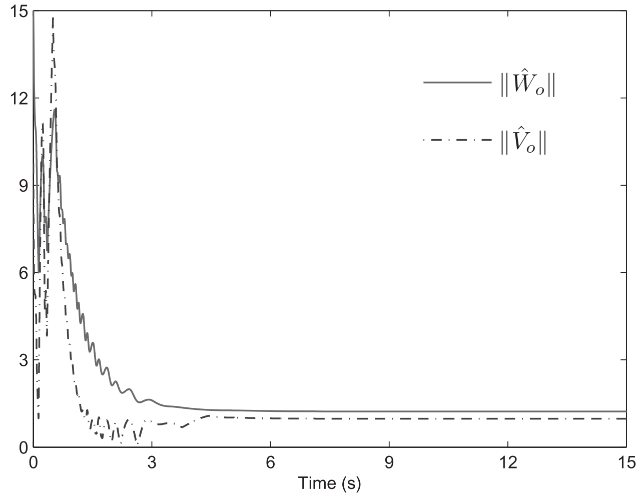


Fig. 5 *2-norm of observer NN weights $||\hat{W}_o||$ and $||\hat{V}_o||$*

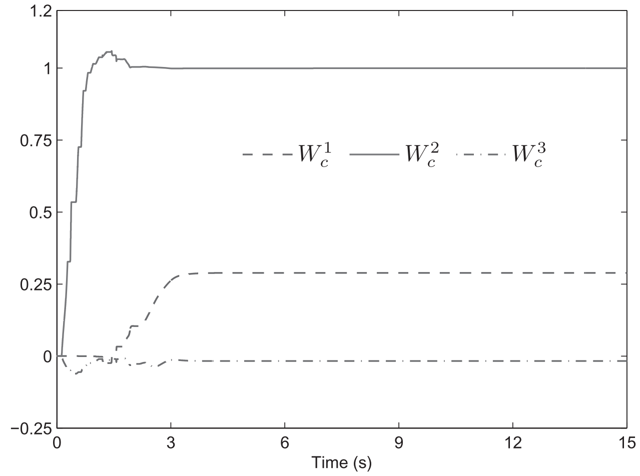


Fig. 6 *Convergence of the critic NN weight $\hat{W}_c$*

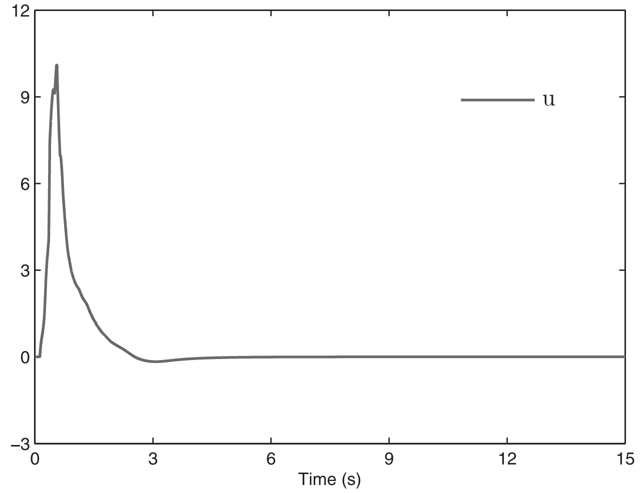


Fig. 7 *Control input u*

can also obtain stable system states, respectively. We omit the simulation results here, for they have been presented in [27, 28]. It is quite straightforward to notice that, the trajectories of system states given in [27, 28] share common

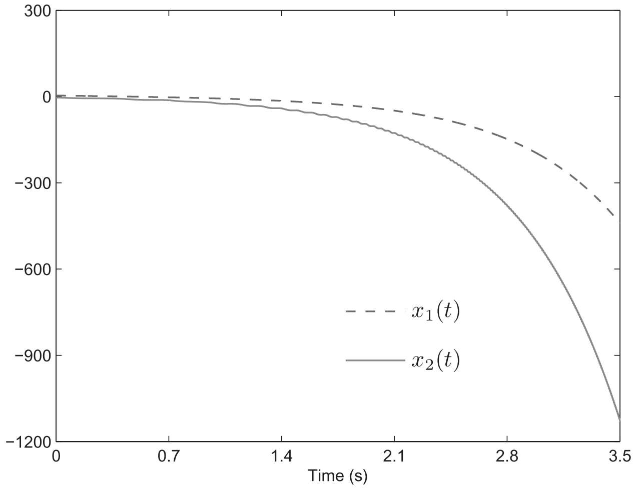


Fig. 8 *Trajectories of states without considering the third term in (34)*

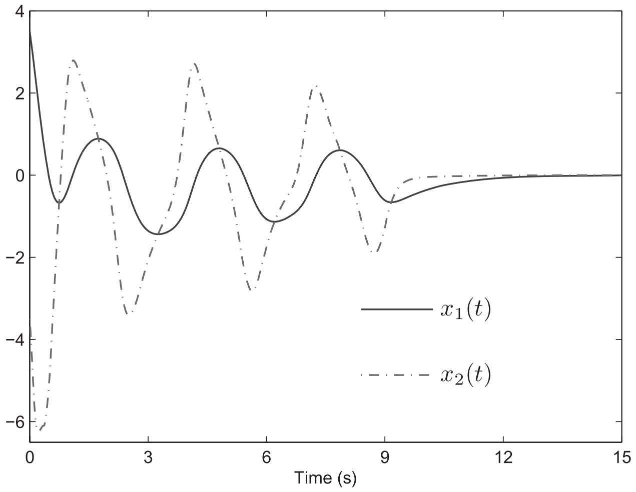


Fig. 9 *System states with the algorithm proposed in [29]*

feature with Fig. 9, which is oscillatory before it converges to the equilibrium point. This feature is caused by adding the PE signal.

Several notes about the simulation results are listed as follows:

- From Figs. 2–4, it is observed that the NN observer can approximate the real system very fast and well.
- From Figs. 5 and 6, one shall find that the observer NN and the critic NN are tuned simultaneously.
- From Figs. 2–7, one can observe that the system states, and the estimated weights of the observer NN and the critic NN are all guaranteed to be UUB, while keeping the closed-loop system stable.
- From Figs. 2 and 3, one can observe that there are almost no oscillations of system states. As aforementioned, the PE signal always leads to oscillations of system states (see Fig. 9 and the simulation results presented in [27, 28]). This verifies that the restrictive PE condition is removed by using recorded and instantaneous data simultaneously. Hence, a significant advantage of the present algorithm as compared with the methods proposed in [27–29] lies in that the PE condition is relaxed.

doi: 10.1049/iet-cta.2014.0186

- From Figs. 6 and 7, one can find that both the initial weights for the critic NN and the initial control are zeros. In this circumstance, the initial control cannot stabilise the system since the initial state x_0 is non-zero. Nevertheless, the initial control must stabilise the system in [27, 28]. Consequently, in comparison with the methods proposed in [27, 28], a distinct advantage of the developed algorithm in this paper lies in that the initial stabilising control is not required any more.
- From Fig. 8, one shall find that, without the third term involving in (34), the system is unstable during the learning process of the critic NN. In addition, compared Figs. 2 and 3 with Fig. 9, it is observed that our algorithm can make system states converge to the equilibrium point faster than the algorithm proposed in [29].

7 Conclusion

In this paper, we have developed a new ADP-based algorithm which solves the optimal control problem for input-affine non-linear CT systems in the presence of unknown internal dynamics. The algorithm constructs an observer–critic architecture. Based on the present algorithm, the observer NN and the critic NN are tuned simultaneously. Meanwhile, the conditions that the initial stabilising control and the PE condition are both relaxed. In our future work, we shall focus on how to develop online algorithms for solving optimal control problems of non-affine non-linear CT systems.

8 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants nos. 61034002, 61233001, 61273140, 61304086 and 61374105, and in part by the Beijing Natural Science Foundation under Grant no. 4132078.

9 References

1 Bryson, A.E., Ho, Y.C.: ‘Applied optimal control: optimization’, *Estimation and Control* (Taylor & Francis, 1975)

2 Lewis, F.L., Vrabie, D., Syrmos, V.L.: ‘Optimal control’ (John Wiley & Sons, 2012)

3 Li, H., Liu, D.: ‘Optimal control for discrete-time affine nonlinear systems using general value iteration’, *IET Control Theory Appl.*, 2012, **6**, (18), pp. 2725–2736

4 Yang, X., Liu, D., Huang, Y: ‘Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints’, *IET Control Theory Appl.*, 2013, **7**, (17), pp. 2037–2047

5 Yang, X., Liu, D., Wang, D.: ‘Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints’, *Int. J. Control*, 2014, **87**, (3), pp. 553–566

6 Bellman, R.E.: ‘Dynamic programming’ (Princeton University Press, 1957)

7 Werbos, P.J.: ‘Beyond regression: new tools for prediction and analysis in the behavioral sciences’. PhD thesis, Harvard University, 1974

8 Werbos, P.J.: ‘Approximate dynamic programming for real-time control and neural modeling’, in White, D.A., Sofge, D.A. (Eds.): ‘Handbook of intelligent control: neural, fuzzy, and adaptive approaches’ (Van Nostrand Reinhold, 1992)

9 Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R.: ‘Adaptive dynamic programming’, *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, 2002, **32**, (2), pp. 140–153

10 Wang, F.Y., Zhang, H., Liu, D.: ‘Adaptive dynamic programming: an introduction’, *IEEE Comput. Intell. Mag.*, 2009, **4**, (2), pp. 39–47

11 Liu, D., Wang, D., Yang, X.: ‘An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs’, *Inf. Sci.*, 2013, **220**, pp. 331–342

12 Liu, D., Wei, Q.: ‘Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems’, *IEEE Trans. Cybern.*, 2013, **43**, (2), pp. 779–789

13 Wei, Q., Liu, D.: ‘Numerical adaptive learning control scheme for discrete-time non-linear systems’, *IET Control Theory Appl.*, 2013, **7**, (11), pp. 1472–1486

14 Powell, W.B.: ‘Approximate dynamic programming: solving the curses of dimensionality’ (Wiley, 2011, 2nd edn.)

15 Liu, D., Zhang, Y., Zhang, H.: ‘A self-learning call admission control scheme for CDMA cellular networks’, *IEEE Trans. Neural Netw.*, 2005, **16**, (5), pp. 1219–1228

16 Liu, D., Javaherian, H., Kovalenko, O., Huang, T.: ‘Adaptive critic learning techniques for engine torque and air-fuel ratio control’, *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2008, **38**, (4), pp. 988–993

17 Prokhorov, D.V., Wunsch, D.C.: ‘Adaptive critic designs’, *IEEE Trans. Neural Netw.*, 1997, **8**, (5), pp. 997–1007

18 Si, J., Wang, Y.T.: ‘On-line learning control by association and reinforcement’, *IEEE Trans. Neural Netw.*, 2001, **12**, (2), pp. 264–276

19 Sutton, R.S., Barto, A.G.: ‘Reinforcement learning–an introduction’ (MIT Press, 1998)

20 Lewis, F.L., Vrabie, D., Vamvoudakis, K.G.: ‘Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers’, *IEEE Control Syst. Mag.*, 2012, **32**, (6), pp. 76–105

21 Liu, D., Yang, X., Li, H.: ‘Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics’, *Neural Comput. Appl.*, 2013, **23**, (7–8), pp. 1843–1850

22 Wu, H.N., Luo, B.: ‘Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H^∞ Control’, *IEEE Trans. Neural Netw. Learn. Syst.*, 2012, **23**, (12), pp. 1884–1895

23 Ni, Z., He, H., Wu, J.: ‘Adaptive learning in tracking control based on the dual critic network design’, *IEEE Trans. Neural Netw. Learn. Syst.*, 2013, **24**, (6), pp. 913–928

24 Zhang, H., Cui, L., Zhang, X., Luo, Y.: ‘Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method’, *IEEE Trans. Neural Netw.*, 2011, **22**, (12), pp. 2226–2236

25 Abu-Khalaf, M., Lewis, F.L.: ‘Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach’, *Automatica*, 2005, **41**, (5), pp. 779–791

26 Vrabie, D., Lewis, F.L.: ‘Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems’, *Neural Netw.*, 2009, **22**, (3), pp. 237–246

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

28 Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K.G., Lewis, F.L., Dixon, W.E.: ‘A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems’, *Automatica*, 2013, **49**, (1), pp. 82–92

29 Dierks, T., Jagannathan, S.: ‘Optimal control of affine nonlinear continuous-time systems’. Am. Control Conf., Baltimore, MD, USA, June–July 2010, pp. 1568–1573

30 Zhang, H., Cui, L., Luo, Y.: ‘Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP’, *IEEE Trans. Cybern.*, 2013, **43**, (1), pp. 206–216

31 Nodland, D., Zargarzadeh, H., Jagannathan, S.: ‘Neural network-based optimal adaptive output feedback control of a helicopter UAV’, *IEEE Trans. Neural Netw. Learn. Syst.*, 2013, **24**, (7), pp. 1061–1073

32 Haykin, S.: ‘Neural networks and learning machines’ (Prentice-Hall, 2008, 3rd edn.)

33 Khalil, H.K.: ‘Nonlinear systems’ (Prentice-Hall, 2001, 3rd edn.)

34 Abdollahi, F., Talebi, H.A., Patel, R.V.: ‘A stable neural network-based observer with application to flexible-joint manipulators’, *IEEE Trans. Neural Netw.*, 2006, **17**, (1), pp. 118–129

35 Hornik, K., Stinchcombe, M., White, H.: ‘Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks’, *Neural Netw.*, 1990, **3**, (5), pp. 551–560

36 Lewis, F.L., Yesildirek, A., Liu, K.: ‘Multilayer neural-net robot controller with guaranteed tracking performance’, *IEEE Trans. Neural Netw.*, 1996, **7**, (2), pp. 388–399

37 Yu, W.: ‘Recent advances in intelligent control systems’ (Springer-Verlag, 2009)

38 Yang, X., Liu, D., Wang, D., Wei, Q.: ‘Discrete-time online learning control for a class of unknown nonaffine nonlinear systems using reinforcement learning’, *Neural Netw.*, 2014, **55**, pp. 30–41

39 Rudin, W.: ‘Principles of mathematical analysis’ (McGraw-Hill’, Inc., 1976, 3rd edn.)

40 Lewis, F.L., Jagannathan, S., Yesildirek, A.: ‘Neural network control of robot manipulators and nonlinear systems’ (Taylor & Francis, 1999)

41 Gampbell, S.L., Meger, C.D.: ‘Generalized inverses of linear transformations’ (Dover Publications, 1991)

42 Horn, R.A., Johnson, C.R.: ‘Matrix analysis’ (Cambridge University Press, 2012, 2nd edn.)

43 Beard, R., Saridis, G., Wen, J.: ‘Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation’, *Automatica*, 1997, **33**, (12), pp. 2159–2177

44 Chowdhary, G.V.: ‘Concurrent learning for convergence in adaptive control without persistency of excitation’. PhD thesis, Georgia Institute of Technology, 2010

45 Padhi, R., Unnikrishnan, N., Wang, X., Balakrishnan, S.N.: ‘A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems’, *Neural Netw.*, 2006, **19**, (10), pp. 1648–1660

46 Chowdhary, G.V.: ‘A singular value maximizing data recording algorithm for concurrent learning’. American Control Conf., San Francisco, CA, USA, 2011, pp. 3547–3552

doi: 10.1049/iet-cta.2014.0186