Automatica 180 (2025) 112462

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Lyapunov-based adaptive deep system identification for approximate dynamic programming☆

Wanjiku A. Makumi [a,*], Omkar Sudhir Patil [b], Warren E. Dixon [b]

[a] Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542, USA
[b] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville FL 32611-6250, USA

ARTICLE INFO

Article history:
Received 14 March 2024
Received in revised form 26 February 2025
Accepted 6 June 2025

ABSTRACT

Recent developments in approximate dynamic programming (ADP) use deep neural network (DNN)-based system identifiers to solve the infinite horizon state regulation problem; however, the DNN weights do not continually adjust for all layers. In this paper, ADP is performed using a Lyapunov-based DNN (Lb-DNN) adaptive identifier that involves online weight updates. Provided the Jacobian of the Lb-DNN satisfies the persistence of excitation condition, the Lb-DNN weights exponentially converge to a residual approximation error, and the corresponding control policy converges to a neighborhood of the optimal policy. Simulation results show that the Lb-DNN yields 49.85% improved root mean squared (RMS) function approximation error in comparison to a baseline ADP DNN result and faster convergence of the RMS regulation error, RMS controller error, and RMS function approximation error.

Published by Elsevier Ltd.

1. Introduction

Traditional optimal controllers aim to minimize predetermined performance metrics and are typically designed *a priori*, relying on knowledge of the system's dynamical model, by solving the Hamilton–Jacobi–Bellman (HJB) equation (Liberzon, 2012). However, accurately modeling practical systems can be challenging, especially for nonlinear systems where solving the nonlinear HJB equation analytically may be infeasible. Many numerical solution techniques are available to solve the HJB equation; however, numerical solution techniques require exact model knowledge and typically involve open-loop implementation of offline solutions. Open-loop implementations are sensitive to disturbances, changes in objectives, and changes in the system dynamics; hence, online closed-loop solutions of optimal control problems are sought-after.

Reinforcement learning (RL) is a method to solve optimization, optimal control, and decision making problems where an agent interacts with its environment adjusting its actions or control policies based on the feedback it receives (Sutton & Barto, 1998). In contrast to traditional optimal control techniques, RL methods provide an approximate solution to the HJB equation by adjusting the control policy based on state feedback (Gao, Mynuddin, Wunsch, & Jiang, 2021; Modares, Lewis, Kang, & Davoudi, 2018; Pang & Jiang, 2020; Vamvoudakis, Vrabie, & Lewis, 2014; Vrabie, Pastravanu, Abu-Khalaf, & Lewis, 2009). A common RL tool used to approximately solve the HJB equation is approximate dynamic programming (ADP), where the optimal value function is approximated using a neural network (NN)-based actor–critic architecture (Kamalapurkar, Walters, Rosenfeld, & Dixon, 2018). The actor NN is tasked with learning the optimal control policy, and the critic NN is tasked with learning the optimal value function. In results such as Kamalapurkar, Walters, et al. (2018), Lewis and Liu (2013) and Vrabie, Vamvoudakis, and Lewis (2013) the actor and critic NN update laws are designed based on a Lyapunov-based analysis to minimize the optimal value function approximation error in approximating the HJB equation, known as the Bellman error (BE). The BE provides an indirect measure of the quality of the value function estimate at each point of evaluation. Through a method called BE extrapolation the BE can be evaluated at off-trajectory points and used in the update laws to yield better performance by enhanced exploration (Kamalapurkar, Walters, & Dixon, 2016).

The evaluation and extrapolation of the BE requires full knowledge of the system model. Motivated by engineering systems that contain uncertain dynamics, ADP methods have been developed that use an approximate model such as linear parameterization (Makumi, Greene, Bell, Bialy et al., 2023; Makumi, Greene, Bell, Nivison et al., 2023; Walters, Kamalapurkar, Voight,

☆ This research is supported in part by AFOSR, United States grant FA9550-19-1-0169, Office of Naval Research, United States grant N00014-21-1-2481, and AFRL, United States grant FA8651-21-F-1027. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agencies. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Kyriakos G. Vamvoudakis under the direction of Editor Miroslav Krstic.

* Corresponding author.
E-mail addresses: wanjiku.makumi@us.af.mil (W.A. Makumi), patilomkarsudhir@ufl.edu (O.S. Patil), wdixon@ufl.edu (W.E. Dixon).

https://doi.org/10.1016/j.automatica.2025.112462
0005-1098/Published by Elsevier Ltd.

Schwartz, & Dixon, 2018), single-layer NNs (Deptula, Bell, Doucette, Curtis, & Dixon, 2020; Deptula, Bell, Zegers, Licitra, & Dixon, 2021; Kamalapurkar, Andrews, Walters, & Dixon, 2017; Kamalapurkar, Klotz, Walters, & Dixon, 2018; Wang, Gao, Zhao, & Ahn, 2020), and deep NNs (DNNs) (Greene, Bell, Nivison, & Dixon, 2023; Makumi, Bell, & Dixon, 2023, 2024; Philor, Makumi, Bell, & Dixon, 2024). Recent advancements in Le, Patil, Nino, and Dixon (2024), LeCun, Bengio, and Hinton (2015), and Makumi, Bell, et al. (2023) showcase the advantages of DNN-based approximators when compared to single-layer NNs. However, the DNNs used in Greene et al. (2023), Makumi, Bell, et al. (2023), Makumi et al. (2024) and Philor et al. (2024), known as multi-timescale DNNs, update only the output-layer weights in real-time, whereas the inner-layer weights are updated iteratively by minimizing a loss function based on datasets obtained over discrete training intervals. As a result, the inner-layer weights are not updated via adaptive update laws. Moreover, there are no guarantees provided on the identification of inner-layer weights under any sufficient excitation condition.

Recent advancements in adaptive control provide Lyapunov-based DNN (Lb-DNN) controllers with real-time updates for all weights for several architectures in Griffis, Patil, Bell, and Dixon (2023), Griffis, Patil, Hart, and Dixon (2024), Griffis, Patil, Makumi, and Dixon (2023), Hart, Griffis, Patil, and Dixon (2024), Hart, Patil, Griffis, and Dixon (2023), Patil, Le, Greene, and Dixon (2022), Griffis, Patil, Bell, and Dixon (2022). Although these recent online DNN results eliminate the restriction of offline training and allow for sustained learning, they are designed to address the trajectory tracking problem based on tracking error feedback, and are not applicable to perform system identification for ADP due to the lack of parameter convergence guarantees. Thus, it is desirable to construct adaptation laws to identify the system for incorporation in ADP.

A common challenge in system identification is the lack of availability of state-derivative information. Previous results such as Makumi, Bell, et al. (2023), Makumi et al. (2024) and Philor et al. (2024) use integral concurrent learning (ICL) to eliminate the requirement of state derivative information. However, ICL only identifies the outer-layer weights due to the nonlinear parameterization of the DNN. The nonlinear parameterization also makes it difficult to yield performance guarantees on the system identification.

This paper introduces the first ADP method involving an Lb-DNN as an adaptive system identifier. The developed identifier uses a least squares adaptation law with a bounded gain forgetting factor to update the weights of all layers of the DNN. To overcome the challenges posed by the lack of state-derivative information, we construct a robust integral of the sign of the error (RISE)-based observer that provides a secondary estimate of the dynamics. While the RISE-based observer can provide an estimate of the dynamics, it would only be an instantaneous estimate and could not be used in BE extrapolation. The difference between the two estimates is calculated as an identification error which is used to develop a least squares adaptive update law (Patil, Griffis, Makumi, & Dixon, 2023). Through a combined Lyapunov-based stability analysis, the system identifier composed of the DNN and RISE-based dynamics observer is shown to exponentially converge to a neighborhood of the DNN weight estimation error, provided the Jacobian of the DNN satisfies the persistence of excitation (PE) condition. Simultaneously, the approximate DNN-based model is used to achieve approximate BE extrapolation. The proposed DNN is used for system identification, while standard NNs are used for the actor and critic. The resulting ADP formulation achieves convergence of the developed control policy to a neighborhood of the optimal control policy. Comparative simulation results demonstrate a significant performance improvement with the developed adaptive DNN controller in comparison to the multi-timescale DNN. Specifically, the developed controller yields 49.85% improved root mean squared (RMS) function approximation error, as well as faster convergence of the RMS regulation error, RMS controller error, and RMS function approximation error.

2. Background

2.1. Notation

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. The pseudo-inverse of full row rank matrix $A \in \mathbb{R}^{n\times m}$ is denoted by A^+, where $A^+ \triangleq A^\top \left(AA^\top\right)^{-1}$. The right-to-left matrix product operator is represented by $\overset{\curvearrowleft}{\prod}$, i.e., $\overset{\curvearrowleft}{\prod_{p=1}^{m}} A_p = A_m \ldots A_2 A_1$ and $\overset{\curvearrowleft}{\prod_{p=a}^{m}} A_p = I$ if $a > m$. The Kronecker product is denoted by $\otimes$. The Jacobian $\left[\frac{\partial f(x,y)}{\partial x_1}^\top, \ldots, \frac{\partial f(x,y)}{\partial x_n}^\top\right]$ is denoted by $\nabla_x f(x,y)$. Unless otherwise specified, let $\nabla \triangleq \nabla_x$. Function compositions are denoted using the symbol $\circ$, e.g., $(g \circ h)(x) = g(h(x))$, given suitable functions g and h. Given $w \in \mathbb{R}$ and some functions f and g, the notation $f(w) = \mathcal{O}^m(g(w))$ means that there exists some constants $M \in \mathbb{R}_{>0}$ and $w_0 \in \mathbb{R}$ such that $\|f(w)\| \leq M \|g(w)\|^m$ for all $w \geq w_0$. Given some matrix $A \triangleq \left[a_{i,j}\right] \in \mathbb{R}^{n\times m}$, where $a_{i,j}$ denotes the element in the ith row and jth column of A, the vectorization operator is defined as $\text{vec}(A) \triangleq [a_{1,1}, \ldots, a_{n,1}, \ldots, a_{1,m}, \ldots, a_{n,m}]^\top \in \mathbb{R}^{nm}$. A square diagonal matrix with elements of vector y on the main diagonal is denoted by $\text{diag}(y)$. An $n \times n$ identity matrix is denoted by $I_{n\times n}$. Matrices of ones and zeros with n rows and m columns are denoted by $\mathbf{1}_{n\times m}$ and $\mathbf{0}_{n\times m}$, respectively. Both the Euclidean norm for vectors and the Frobenius norm for matrices are denoted by $\|\cdot\|$, and the 1-norm is denoted by $\|\cdot\|_1$. The space of continuous functions with continuous first n derivatives is denoted by $\mathcal{C}^n$. The notation $\overset{a.a.t.}{(\cdot)}$ denotes that the relation $(\cdot)$ holds for almost all time (a.a.t.). Given any $A \in \mathbb{R}^{p\times a}$, $B \in \mathbb{R}^{a\times r}$, and $C \in \mathbb{R}^{r\times s}$, the vectorization operator satisfies the property (Bernstein, 2009, Proposition 7.1.9)

$$\text{vec}(ABC) = (C^\top \otimes A)\text{vec}(B). \tag{1}$$

Differentiating (1) on both sides with respect to vec (B) yields the property

$$\frac{\partial}{\partial \text{vec}(B)} \text{vec}(ABC) = C^\top \otimes A. \tag{2}$$

A function $y : \mathcal{I}_y \to \mathbb{R}^n$ is called a *Filippov solution* of $\dot{y} = h(y,t)$ on the interval $\mathcal{I}_y \subseteq \mathbb{R}_{\geq 0}$, given some Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, if y is absolutely continuous on $\mathcal{I}_y$, and $\dot{y} \in K[h](y,t)$ for almost all $t \in \mathcal{I}_y$, where $K[\cdot]$ denotes the Filippov set-valued map defined in Paden and Sastry (1987, Equation 2b).

2.2. Problem formulation

Consider a continuous-time, control-affine nonlinear dynamical system

$$\dot{x} = f(x) + g(x)u \tag{3}$$

where $x \in \mathbb{R}^n$ denotes the system state, $u \in \mathbb{R}^m$ denotes the control input, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the unknown drift dynamics, and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the control effectiveness. For simplicity in this paper, we assume the function $g(x)$ is known, though one could leverage the methods in Deptula et al. (2020) to also account for an unknown $g(x)$.

Assumption 1. *The function f is $\mathcal{C}^2$ and $f(0) = 0$.*

Assumption 2. *The function g is a known locally Lipschitz function, bounded such that $0 < \|g(x)\| \leq \overline{g}\ \forall x \in \mathbb{R}^n$, where $\overline{g} \in \mathbb{R}_{>0}$ is a known bound.*

The control objective is to solve the infinite horizon optimal regulation problem online, i.e. find an optimal control policy u that minimizes the cost function

$$J(x,u) = \int_0^\infty Q(x(\tau)) + u(\tau)^\top R u(\tau)\, d\tau. \tag{4}$$

In (4), $Q : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a positive definite (PD) cost function where Q satisfies $\underline{q}(\|x\|) \leq Q(x) \leq \overline{q}(\|x\|)$ for $\underline{q}, \overline{q} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $R \in \mathbb{R}^{m\times m}$ is a user-defined constant PD symmetric cost matrix.

The cost-to-go (i.e. the infinite horizon value function) $V^* : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is defined as

$$V^*(x) \triangleq \min_{u\in U} \int_t^\infty Q(x(\tau)) + u(\tau)^\top R u(\tau)\, d\tau, \tag{5}$$

where $U \subseteq \mathbb{R}$ is the action space for u.

A major roadblock in finding the approximately optimal control policy is that the drift dynamics f are unknown and involve complex nonlinearities. The following section provides a method for identifying the unknown drift dynamics in real-time using DNNs.

3. System identification

DNNs are known to be effective at approximating unknown nonlinear functions such as the drift dynamics f. Previous results in Greene et al. (2023), Makumi, Bell, et al. (2023), Makumi et al. (2024) and Philor et al. (2024) have used DNNs for system identification in the approximate optimal control problem. However, in those results, the inner-layer weights of the DNN were updated in batches using offline training techniques. Offline training techniques require large amounts of data and do not account for disturbances and uncertainties in real-time. In this section, the system dynamics are identified using an Lb-DNN with real-time weight adaptation laws for all layers of the DNN. The developed DNN weight estimates are shown to approximately converge to their true values under an explicit PE condition, unlike the previously cited results.

3.1. Dynamics estimate

Let $\Phi : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ denote a generalized DNN defined in the Appendix where p represents the total number of DNN weights. DNNs are known to approximate continuous functions on a compact set using the Universal Function Approximation theorem (Kidger & Lyons, 2020).[1] The drift dynamics can be approximated with a DNN on a compact set $\Omega \subset \mathbb{R}^n$ as

$$f(x) = \Phi(x,\theta^*) + \varepsilon(x) \tag{6}$$

where $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown function reconstruction error that can be bounded as $\sup_{x\in\Omega} \|\varepsilon(x)\| \leq \overline{\varepsilon}$, and $\theta^* \in \mathbb{R}^p$ denotes ideal weights such that $\sup_{x\in\Omega} \|f(x) - \Phi(x,\theta^*)\| \leq \overline{\varepsilon}$. An estimate of the dynamics is represented as $\Phi\left(x,\hat{\theta}\right)$ where $\hat{\theta} \in \mathbb{R}^p$ is the subsequently designed adaptive estimate of the ideal DNN weights θ^*.[2]

[1] The subsequent stability analysis guarantees that if x is initialized in an appropriately-sized subset of Ω, then it will stay in Ω.

[2] Various different architectures such as fully-connected DNNs, ResNets, LSTMs found in Griffis, Patil, Bell, et al. (2023), Griffis, Patil, et al. (2022), and Patil, Le, et al. (2022) respectively, can be used in the system identifier.

Assumption 3. *There exists a known constant $\overline{\theta} \in \mathbb{R}_{>0}$ such that the unknown ideal weights can be bounded as $\|\theta^*\| \leq \overline{\theta}$.*

Real-time system identifiers typically use an identification error as feedback. However, the identification error of the dynamics cannot be directly evaluated due to the absence of state-derivative information. To overcome this challenge, a RISE-based dynamics observer is used to obtain an instantaneous second estimate of the dynamics (Isaly, Patil, Sweatland, Sanfelice, & Dixon, 2024). The subsequent RISE-based dynamics observer is capable of exponentially identifying uncertainty and disturbances in the function and is designed as

$$\dot{\hat{x}} = \hat{f} + gu + \alpha_1 \tilde{x} \tag{7}$$

$$\dot{\hat{f}} = \tilde{x} + k_f\left(\dot{\tilde{x}} + \alpha_1 \tilde{x}\right) + \beta_f \operatorname{sgn}(\tilde{x}) \tag{8}$$

where $\hat{x}, \hat{f} \in \mathbb{R}^n$ are the observer estimates of x and f, respectively, $\tilde{x}, \tilde{f} \in \mathbb{R}^n$ are the observer errors $\tilde{x} \triangleq x - \hat{x}$ and $\tilde{f} \triangleq f(x) - \hat{f}$, respectively, and $\alpha_1, k_f, \beta_f \in \mathbb{R}_{>0}$ denote constant observer gains. The observer error $\tilde{x}$ is known because x and $\hat{x}$ are known. However, since $\dot{\tilde{x}}$ is unknown, (8) can be implemented by integrating both sides and using the relation $\int_0^t \dot{\tilde{x}}(\tau)\, d\tau = \tilde{x}(t) - \tilde{x}(0)$ to obtain $\hat{f}(t) = \hat{f}(0) + k_f\tilde{x}(t) - k_f\tilde{x}(0) + \int_0^t \left[(k_f\alpha_1 + 1)\tilde{x}(\tau) + \beta_f \operatorname{sgn}(\tilde{x}(\tau))\right] d\tau$ which is a solution to (8). Taking the derivative of $\tilde{x}$ and substituting (7) yields

$$\dot{\tilde{x}} = \tilde{f} - \alpha_1 \tilde{x}. \tag{9}$$

Additionally, taking the derivative of $\tilde{f}$ and substituting (8) and (9) yields

$$\dot{\tilde{f}} = \dot{f} - \tilde{x} - k_f\tilde{f} - \beta_f \operatorname{sgn}(\tilde{x}), \tag{10}$$

where $\dot{f} \triangleq \frac{\partial f}{\partial x}\dot{x}$. An identification error $E \in \mathbb{R}^n$ based on the observer estimate of f and the DNN estimate of f is calculated as

$$E = \hat{f} - \Phi\left(x,\hat{\theta}\right) \tag{11}$$

and is used to update the weights of the DNN in real-time.

Remark 1. Although a RISE-based observer is capable of producing an instantaneous estimate of the drift dynamics by essentially acting as a state-derivative estimator, the subsequent control development requires extrapolation of the dynamics to unexplored areas of the state space that can only be achieved using the identified DNN.

3.2. Adaptation laws

To facilitate the subsequent analysis, the least squares adaptive update law is designed as

$$\dot{\hat{\theta}} = \Gamma_\theta\left(-k_{\hat{\theta}}\hat{\theta} + \Phi'^\top\left(x,\hat{\theta}\right)E\right), \tag{12}$$

where the Jacobian $\Phi'\left(x,\hat{\theta}\right) \in \mathbb{R}^{n\times p}$ is calculated using (A.2) and (A.3), and the term $\Gamma_\theta \in \mathbb{R}^{p\times p}$ denotes a symmetric, PD time-varying least squares adaptation gain matrix that is a solution to Slotine and Li (1989, Eqns. (16) and (17))

$$\frac{d}{dt}\Gamma_\theta^{-1} = -\beta(t)\Gamma_\theta^{-1} + \Phi'^\top\left(x,\hat{\theta}\right)\Phi'\left(x,\hat{\theta}\right), \tag{13}$$

where the bounded-gain time-varying forgetting factor $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is designed as

$$\beta(t) \triangleq \beta_0\left(1 - \frac{\lambda_{\max}\{\Gamma_\theta\}}{\kappa_0}\right), \tag{14}$$

where $\beta_0, \kappa_0 \in \mathbb{R}_{>0}$ are user-selected constants that denote the maximum forgetting rate and the prescribed bound on $\lambda_{\max}\{\Gamma_\theta\}$, respectively. The adaptation gain in (13) is initialized to be PD such that $\lambda_{\max}\{\Gamma_\theta(0)\} < \kappa_0$, and it can be shown that $\Gamma_\theta(t)$ remains PD for all $t \in \mathbb{R}_{\geq 0}$ (Slotine & Li, 1989). The term $\beta(t)$ can be lower bounded as $\beta \geq \beta_1$, where $\beta_1 \in \mathbb{R}_{\geq 0}$ is a constant which satisfies the properties stated in the subsequent remark.

If $\Phi'\left(x, \hat{\theta}\right)$ satisfies the PE condition, i.e., there exists constants $\varphi_1, \varphi_2 \in \mathbb{R}_{>0}$ for all $t_1 \in \mathbb{R}_{\geq 0}$ and some $T \in \mathbb{R}_{>0}$ such that $\varphi_1 I_p \leq \int_{t_1}^{t_1+T} \Phi'^{\top}\left(x(\tau), \hat{\theta}(\tau)\right) \Phi'\left(x(\tau), \hat{\theta}(\tau)\right) d\tau \leq \varphi_2 I_p$, then it can be shown that $\beta_1 > 0$ (Slotine & Li, 1989, Sec. 4.2).

Remark 2. The PE condition requires the Jacobian Φ' to be sufficiently rich, which yields sufficient exploration of the state–space. Under this condition, the weight estimates are shown to converge to a neighborhood of their ideal values. As a result, the DNN can generalize beyond the explored trajectory, thus allowing extrapolation in the subsequent control development.

Analyzing the convergence properties of the adaptive update law in (12) is challenging due to the nested nonlinear parameterization of the DNN. A first-order Taylor series approximation is used to overcome the challenges of the nested nonlinear parameterization introduced by the DNN. Applying a first-order Taylor series approximation to the generalized DNN illustrated in the Appendix yields

$$\Phi(x, \theta^*) - \Phi\left(x, \hat{\theta}\right) = \Phi'\left(x, \hat{\theta}\right)\tilde{\theta} + \mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right), \tag{15}$$

where $\tilde{\theta} \in \mathbb{R}^p$ denotes the parameter estimation error $\tilde{\theta} \triangleq \theta - \hat{\theta}$, and $\mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right)$ denotes the higher-order terms. By adding and subtracting f and substituting (6) and (15), the identification error E can be rewritten as

$$E = -\tilde{f} + \Phi'\left(x, \hat{\theta}\right)\tilde{\theta} + \Delta, \tag{16}$$

where $\Delta \triangleq \mathcal{O}\left(\left\|\tilde{\theta}\right\|^2\right) + \varepsilon(x)$. Since $\dot{\tilde{\theta}} \triangleq -\dot{\hat{\theta}}$, by substituting (12) and (16) the time derivative of $\tilde{\theta}$ is calculated as

$$\begin{aligned}\dot{\tilde{\theta}} = {} & \Gamma_\theta k_{\hat{\theta}}\theta^* - \Gamma_\theta\left(k_{\hat{\theta}} + \Phi'^{\top}\left(x, \hat{\theta}\right)\Phi'\left(x, \hat{\theta}\right)\right)\tilde{\theta} \\ & + \Gamma_\theta \Phi'^{\top}\left(x, \hat{\theta}\right)\tilde{f} - \Gamma_\theta \Phi'^{\top}\left(x, \hat{\theta}\right)\Delta.\end{aligned} \tag{17}$$

3.3. Stability analysis

To achieve exponential convergence, a P-function is included in the subsequent Lyapunov analysis in addition to the typical sum of norm squared error terms (Patil, Isaly, Xian, & Dixon, 2022). The P-function is used to prove exponential convergence with the RISE-based observer error $\tilde{f}$, therefore facilitating faster function approximation which in turn strengthens the accuracy of the control policy. The P-function is designed as

$$\begin{aligned}P \triangleq {} & \beta\left\|\tilde{x}\right\|_1 - \tilde{x}^{\top}\tilde{f} + e^{-\lambda_P t} * \left(\tilde{x}^{\top}\ddot{\tilde{f}}\right) \\ & + e^{-\lambda_P t} * \left((\alpha_1 - \lambda_P)\left(\beta\left\|\tilde{x}\right\|_1 - \tilde{x}^{\top}\tilde{f}\right)\right),\end{aligned} \tag{18}$$

where $\lambda_p \in \mathbb{R}_{>0}$ is a user-selected constant, and $\ddot{\tilde{f}} \triangleq \dot{x}^{\top}\left(\frac{\partial^2 f}{\partial x^2}\right)\dot{x} + \frac{\partial f}{\partial x}\ddot{x}$. The convolutional integral $e^{-\lambda_P t} * q = \int_{t_0}^{t} e^{-\lambda_P(t-\sigma)}q(\sigma)d\sigma$ is denoted by '$*$' for any given $q : [t_0, \infty) \to \mathbb{R}$, and can be verified using the Leibniz rule that $\frac{d}{dt}\left(\int_{t_0}^{t} e^{-\lambda_P(t-\sigma)}q(\sigma)d\sigma\right) = q(t) - \lambda_P \int_{t_0}^{t} e^{-\lambda_P(t-\sigma)}q(\sigma)d\sigma$. The convolutional integral therefore satisfies the property $\frac{d}{dt}\left(e^{-\lambda_P t} * q\right) = q(t) - \lambda_P e^{-\lambda_P t} * q$. The mapping $t \mapsto \|e(t)\|_1$ is differentiable for almost all time since $t \mapsto e(t)$ is absolutely continuous and $\|\cdot\|_1$ is globally Lipschitz; hence, the use of the chain rule in Shevitz and Paden (1994, Theorem 2.2) yields $\frac{d}{dt}(\|e\|_1) \overset{a.a.t.}{\in} K[\text{sgn}](e)$. Taking the time-derivative of (18), using Leibniz's rule, and substituting (9) and (18) results in[3]

$$\dot{P} \overset{a.a.t.}{\in} -\lambda_P P + \tilde{f}^{\top}\left(\beta_f \text{sgn}\left(\tilde{x}\right) - \dot{f}\right).$$

Let $z_\theta \triangleq \begin{bmatrix} \tilde{x}^{\top} & \tilde{f}^{\top} & \tilde{\theta}^{\top} & \sqrt{2P}\end{bmatrix}^{\top} \in \mathbb{R}^{2n+p+1}$ denote the concatenated state. The candidate Lyapunov function $V_\theta : \mathbb{R}^{2n+p+1} \to \mathbb{R}$ is defined as

$$V_\theta(z_\theta) = \frac{1}{2}\tilde{x}^{\top}\tilde{x} + \frac{1}{2}\tilde{f}^{\top}\tilde{f} + \frac{1}{2}\tilde{\theta}^{\top}\Gamma_\theta^{-1}\tilde{\theta} + P. \tag{19}$$

The Lyapunov function is bounded as

$$\lambda_1\|z_\theta\|^2 \leq V_\theta(z_\theta) \leq \lambda_2\|z_\theta\|^2, \tag{20}$$

where $\lambda_1 \triangleq \min\{\frac{1}{2}, \frac{1}{2\lambda_{\max}\{\Gamma_\theta\}}\}$ and $\lambda_2 \triangleq \max\{\frac{1}{2}, \frac{1}{2\lambda_{\min}\{\Gamma_\theta\}}\}$. Consider the compact domain $\mathcal{D} \triangleq \left\{\zeta \in \mathbb{R}^{4n+p} : \|\zeta\| \leq \chi\right\}$ where $\chi \in \mathbb{R}_{>0}$ is a bounding constant. The subsequent analysis shows that the concatenated state $z_\theta(t) \in \mathcal{D}$ for all $t \in \mathbb{R}_{\geq 0}$ if z is initialized in the set $\mathcal{S} \triangleq \left\{\zeta \in \mathbb{R}^{2n+p+1} : \|\zeta\| \in \sqrt{\frac{\lambda_1}{\lambda_2}}\chi^2 - \frac{C}{\lambda_3}\right\}$ in the subsequent stability analysis. Using (Patil, Isaly, et al., 2022, Lemma 4) it can be shown that $P \geq 0$ if the gain conditions

$$\alpha > \lambda_P \tag{21}$$

and

$$\beta > \Upsilon_1 + \frac{\Upsilon_2}{\alpha - \lambda_P} \tag{22}$$

are satisfied, where bounds $\left\|\dot{f}\right\| \leq \Upsilon_1$ and $\left\|\ddot{f}\right\| \leq \Upsilon_2$ hold where Υ_1 and Υ_2 are bounding constants based on Assumption 1 and the fact that $\dot{x}$ and $\ddot{x}$ are bounded when $z \in \mathcal{D}$.

Theorem 1. *Provided Assumptions 1–3, the gain conditions in (21), (22), and (25), and the feasibility condition $\chi > \sqrt{\frac{\lambda_2 C}{\lambda_1\lambda_3}}$ are satisfied, the adaptive update laws in (12) and (13) ensure that the estimation errors defined in z_θ are uniformly ultimately bounded (UUB) such that $\|z_\theta(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1}\|z_\theta(0)\|^2 e^{-\frac{\lambda_3}{\lambda_2}t} + \frac{\lambda_2 C}{\lambda_1\lambda_3}\left(1 - e^{-\frac{\lambda_3}{\lambda_2}t}\right)}$.*

Proof. Taking the time derivative of (19) yields

$$\begin{aligned}\dot{V}_\theta \overset{a.a.t.}{\in} {} & \tilde{x}^{\top}\dot{\tilde{x}} + \tilde{f}^{\top}\dot{\tilde{f}} + \tilde{\theta}^{\top}\Gamma_\theta^{-1}\dot{\tilde{\theta}} + \frac{1}{2}\tilde{\theta}^{\top}\left(\frac{d}{dt}\Gamma_\theta^{-1}\right)\tilde{\theta} \\ & - \lambda_P P + \tilde{f}^{\top}\left(\beta_f \text{sgn}\left(\tilde{x}\right) - \dot{f}\right).\end{aligned} \tag{23}$$

By substituting (9), (10), and (17), and canceling coupling terms, (23) can be upper bounded as

$$\begin{aligned}\dot{V}_\theta \overset{a.a.t.}{\leq} {} & -\alpha_1\tilde{x}^2 - k_f\left\|\tilde{f}\right\|^2 - \left(k_{\hat{\theta}} + \frac{\beta(t)}{2}\right)\|\tilde{\theta}\|^2 \\ & - \tilde{\theta}^{\top}\left(\frac{1}{2}\Phi'^{\top}\left(x, \hat{\theta}\right)\Phi'\left(x, \hat{\theta}\right)\right)\tilde{\theta} - \lambda_P P \\ & + \tilde{\theta}^{\top}\Phi'^{\top}\left(x, \hat{\theta}\right)\left(\tilde{f} - \Delta\right) + k_{\hat{\theta}}\tilde{\theta}^{\top}\theta^*.\end{aligned}$$

The parameter estimation error can be bounded as $\|\tilde{\theta}\| \leq \chi$ when $z \in \mathcal{D}$. Additionally, since f and Φ are continuously differentiable, the bounds $\|\Delta\| \leq \gamma_1$ and $\left\|\Phi'\left(x, \hat{\theta}\right)\right\| \leq \gamma_2$ hold when $z_\theta \in \mathcal{D}$, where $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ denote bounding

[3] A more detailed derivation can be found in Patil, Isaly, et al. (2022, Proof of Lemma 3).

constants. Therefore, using Young's inequality yields the bound $\tilde{\theta}^\top \Phi'^\top \left(x, \hat{\theta}\right)\left(\tilde{f} - \Delta\right) \leq \gamma_2 \left\|\tilde{\theta}\right\|^2 + \frac{\gamma_2}{2}\left\|\tilde{f}\right\|^2 + \frac{\gamma_2 \gamma_1^2}{2}$. As a result, $\dot{V}_\theta$ can further be upper-bounded as

$$\dot{V}_\theta \overset{a.a.t.}{\leq} -\lambda_3 \|z\|^2 + C - \frac{1}{2}\tilde{\theta}^\top \Phi'^\top \left(x, \hat{\theta}\right) \Phi'\left(x, \hat{\theta}\right)\tilde{\theta}, \tag{24}$$

when $z_\theta \in \mathcal{D}$, where $\lambda_3 \triangleq \min\{\alpha_1, k_f - \frac{\gamma_2}{2}, \frac{k_{\hat{\theta}}+\beta_1}{2} - \gamma_2, \lambda_P\}$ and $C \triangleq \frac{\gamma_2 \gamma_1^2 + k_{\hat{\theta}} \tilde{\theta}^2}{2}$. Using (20) and (24), when the gain condition

$$\lambda_3 > 0 \tag{25}$$

is satisfied, $\dot{V}$ can be upper-bounded as

$$\dot{V}_\theta \overset{a.a.t.}{\leq} -\frac{\lambda_3}{\lambda_2} V_\theta + C, \tag{26}$$

when $z_\theta \in \mathcal{D}$. Solving the differential inequality in (26) yields

$$V_\theta\left(z(t)\right) \leq V_\theta\left(z(0)\right) \mathrm{e}^{-\frac{\lambda_3}{\lambda_2}t} + \frac{\lambda_2 C}{\lambda_3}\left(1 - \mathrm{e}^{-\frac{\lambda_3}{\lambda_2}t}\right),$$

when $z_\theta \in \mathcal{D}$, and applying (20) yields the bound

$$\|z_\theta(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1}\|z_\theta(0)\|^2 \mathrm{e}^{-\frac{\lambda_3}{\lambda_2}t} + \frac{\lambda_2 C}{\lambda_1 \lambda_3}\left(1 - \mathrm{e}^{-\frac{\lambda_3}{\lambda_2}t}\right)}, \tag{27}$$

when $z_\theta \in \mathcal{D}$. To guarantee $z_\theta(t) \in \mathcal{D}$ for all $t \in \mathbb{R}_{\geq 0}$, (27) can be upper bounded as $\|z_\theta(t)\| \leq \sqrt{\frac{\lambda_2}{\lambda_1}\|z_\theta(0)\|^2 + \frac{\lambda_2 C}{\lambda_1 \lambda_3}}$ for all $t \in \mathbb{R}_{\geq 0}$. Due to the fact that $\mathcal{D} \triangleq \left\{\zeta \in \mathbb{R}^{4n+p} : \|\zeta\| \leq \chi\right\}$, the relation $z_\theta(t) \in \mathcal{D}$ holds if $\sqrt{\frac{\lambda_2}{\lambda_1}\|z_\theta(0)\|^2 + \frac{\lambda_2 C}{\lambda_1 \lambda_3}} \leq \chi$, which is achieved if $\|z_\theta(0)\| \leq \sqrt{\frac{\lambda_1}{\lambda_2}\chi^2 - \frac{C}{\lambda_3}}$, i.e., $z_\theta(0) \in S$; in this case, (27) holds for all $t \in [0, \infty)$.

Remark 3. The gains α_1, k_f, and λ_P can be selected to be sufficiently high, so that $\lambda_3 = \frac{k_{\hat{\theta}}+\beta_1}{2} - \gamma_2$. In that case, the rate of convergence, $\frac{\lambda_3}{\lambda_2}$, primarily depends on Γ_θ, β_1 and $k_{\hat{\theta}}$. Since β_1 is positive under the PE condition as mentioned in Remark 2, a larger value of λ_3 is obtained, thus achieving faster convergence. When the PE condition does not hold, the gain $k_{\hat{\theta}}$, which is based on the sigma modification technique in Ioannou and Sun (1996, Sec. 8.4.1), still ensures UUB convergence by assisting the gain condition in (25). However, it is desirable not to select $k_{\hat{\theta}}$ to be very high and rather leverage more of the PE condition (if satisfied), since a high value of $k_{\hat{\theta}}$ results in a higher value for C, which can worsen the parameter estimation performance.

4. Approximate optimal control

The optimal value function in (5) is the solution to the corresponding HJB equation

$$0 = \nabla V^*(x)\left(f(x) + g(x)u^*\right) Q(x) + u^*(x)^\top R u^*(x), \tag{28}$$

with the condition $V^*(0) = 0$ and where $u^* : \mathbb{R}^n \to \mathbb{R}^m$ is the optimal control policy. Taking the partial derivative of (28) with respect to the minimizing argument $u^*(x)$, setting it equal to zero, and solving for $u^*(x)$ results in the optimal control policy

$$u^*(x) = -\frac{1}{2}R^{-1}g(x)^\top \left(\nabla V^*(x)\right)^\top. \tag{29}$$

Assumption 4. *The optimal value function V^* is continuously differentiable (Kamalapurkar et al., 2016).*

4.1. Value function approximation

The optimal value function is generally unknown for nonlinear systems. To solve for the optimal control policy in (29), the optimal value function can be approximated with a NN in a compact set $\Omega \subset \mathbb{R}^n$ using the Universal Function Approximation Theorem as

$$V^*(x) = W^\top \phi(x) + \epsilon(x) \ \forall x \in \Omega, \tag{30}$$

where $W \in \mathbb{R}^L$ is a vector of unknown weights, $\phi : \mathbb{R}^n \to \mathbb{R}^L$ is a user-defined vector of basis functions, and $\epsilon : \mathbb{R}^n \to \mathbb{R}$ is the bounded function reconstruction error. Substituting (30) into (29), the optimal control policy in (29) can be approximated with a NN as

$$u^*(x) = -\frac{1}{2}R^{-1}g(x)^\top \left(\nabla\phi(x)^\top W + \nabla\epsilon(x)^\top\right). \tag{31}$$

Assumption 5. *There exists a set of known positive constants $\overline{W}, \overline{\phi}, \overline{\nabla\phi}, \overline{\epsilon}, \overline{\nabla\epsilon} \in \mathbb{R}_{>0}$ such that $\sup\|W\| \leq \overline{W}$, $\sup_{x\in\Omega}\|\phi(x)\| \leq \overline{\phi}$, $\sup_{x\in\Omega}\|\nabla\phi(x)\| \leq \overline{\nabla\phi}$, $\sup_{x\in\Omega}\|\epsilon(x)\| \leq \overline{\epsilon}$, and $\sup_{x\in\Omega}\|\nabla\epsilon(x)\| \leq \overline{\nabla\epsilon}$ (Vrabie et al., 2013, Assumptions 9.1.c-e).*

The ideal weights W in (30) and (31) are unknown *a priori*. In this paper, an actor–critic NN architecture is used where actor and critic weight estimates are used to approximate W. The critic weight estimate vector $\hat{W}_c \in \mathbb{R}^L$ is used to approximate (30), resulting in the optimal value function estimate $\hat{V} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}$, defined as

$$\hat{V}\left(x, \hat{W}_c\right) \triangleq \hat{W}_c^\top \phi(x). \tag{32}$$

The actor weight estimate vector $\hat{W}_a \in \mathbb{R}^L$ is used to approximate (31), resulting in the optimal control policy estimate $\hat{u} : \mathbb{R}^n \times \mathbb{R}^L \to \mathbb{R}^m$, defined as

$$\hat{u}\left(x, \hat{W}_a\right) \triangleq -\frac{1}{2}R^{-1}g(x)^\top \left(\nabla\phi(x)^\top \hat{W}_a\right). \tag{33}$$

4.2. Bellman error

The error resulting from approximating the system dynamics, the optimal value function, and the optimal control input introduces an error in the HJB equation in (28). This error, termed the BE, is representative of the performance of the developed method, and is used to update the actor–critic weights in the subsequent development. Replacing the drift dynamics f with the estimate $\Phi\left(x, \hat{\theta}\right)$, the optimal value function $V^*(x)$ with the estimate $\hat{V}\left(x, \hat{W}_c\right)$, and the optimal control policy $u^*(x)$ with the estimate $\hat{u}\left(x, \hat{W}_a\right)$ in (28) results in the BE $\hat{\delta} : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$ defined as

$$\hat{\delta}\left(x, \hat{W}_c, \hat{W}_a\right) \triangleq Q(x) + \hat{u}\left(x, \hat{W}_a\right)^\top R\hat{u}\left(x, \hat{W}_a\right) + \nabla\hat{V}\left(x, \hat{W}_c\right)\left(\Phi\left(x, \hat{\theta}\right) + g(x)\hat{u}\left(x, \hat{W}_a\right)\right). \tag{34}$$

The BE represents the difference between the actor and critic weight approximations and their ideal weight values. While (34) is used for implementation, to facilitate the subsequent stability analysis, (34) can be rewritten in terms of the weight approximation errors $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$. Subtracting (28) from (34) and substituting (30)–(33), the analytical form of the BE in (34) can be expressed as

$$\hat{\delta}\left(x, \hat{W}_c, \hat{W}_a\right) = -\omega^\top \tilde{W}_c + \frac{1}{4}\tilde{W}_a^\top G_\phi(x)\tilde{W}_a + O(x), \tag{35}$$

where the Bellman regressor $\omega : \mathbb{R}^n \times \mathbb{R}^L \times \mathbb{R}^p \to \mathbb{R}^n$ is $\omega\left(x, \hat{W}_a, \hat{\theta}\right) \triangleq \nabla\phi(x)\left(\Phi\left(x, \hat{\theta}\right) + g(x)\hat{u}\left(x, \hat{W}_a\right)\right)$ and $O(x) \triangleq \frac{1}{2}\nabla\epsilon(x) G_R \nabla\phi(x)^\top W + \frac{1}{4}G_\epsilon - \nabla\epsilon(x) f(x)$, where $G_R(x) \triangleq g(x) R^{-1} g(x)^\top$, $G_\phi(x) \triangleq \nabla\phi(x) G_R(x) \nabla\phi(x)^\top$, and $G_\epsilon(x) \triangleq \nabla\epsilon(x) G_R(x) \nabla\epsilon(x)^\top$. Based on Assumption 2, the function G_R is bounded as $\sup_{x\in\Omega} \|G_R\| \leq \overline{g}^2 \lambda_{\max}\left\{R^{-1}\right\} \triangleq \overline{G_R}$, and G_ϕ is bounded as $\sup_{x\in\Omega} \left\|G_\phi\right\| \leq \left(\overline{\nabla\phi}\overline{g}\right)^2 \lambda_{\max}\left\{R^{-1}\right\} \triangleq \overline{G}_\phi$.

The BE in (34) can be evaluated at any user-defined point in the state space using a user-selected state x_i, the critic weight estimate $\hat{W}_c$, and the actor weight estimate $\hat{W}_a$. Using the DNN system identifier and adaptive update laws developed in Section 3, experience can be simulated by extrapolating the BE over unexplored off-trajectory points in the state space via BE extrapolation. BE extrapolation uses the estimated dynamics to yield simultaneous exploration and exploitation providing simulation of experience and yielding faster policy learning. To gain experience for sufficient exploration, the BE is extrapolated to user-defined off-trajectory points $\{x_i : x_i \in \Omega\}_{i=1}^N$, where $N \in \mathbb{N}$ is a user-specified number of total extrapolation trajectories in the compact set Ω (Kamalapurkar et al., 2016). As the estimate of the identified dynamics becomes more accurate, the estimate of the control policy becomes more accurate.

4.3. Update laws for actor and critic weights

The experience gained along the state trajectory and from the extrapolated points is used to update the actor and critic weights simultaneously. In the subsequent adaptive weight update laws, η_{c1}, η_{c2}, η_{a1}, η_{a2}, $\lambda \in \mathbb{R}_{>0}$ are positive constant adaptation gains, $\rho = 1 + \nu\omega^\top \Gamma\omega$, $\rho_i = 1 + \nu\omega_i^\top \Gamma\omega_i$, $\nu \in \mathbb{R}_{>0}$ is a user-defined gain, $\Gamma \in \mathbb{R}^{L\times L}$ is a time-varying least squares gain matrix, and $\underline{\Gamma}$, $\overline{\Gamma} \in \mathbb{R}_{>0}$ denote lower and upper bounds for Γ. The normalized regressors $\frac{\omega}{\rho}$ and $\frac{\omega_i}{\rho_i}$ are bounded as $\sup_{t\in\mathbb{R}_{\geq 0}} \left\|\frac{\omega}{\rho}\right\| \leq \frac{1}{2\sqrt{\nu\underline{\Gamma}}}$ and $\sup_{t\in\mathbb{R}_{\geq 0}} \left\|\frac{\omega_i}{\rho_i}\right\| \leq \frac{1}{2\sqrt{\nu\underline{\Gamma}}}$ for all $x \in \Omega$ and $x_i \in \Omega$, respectively. The critic update law $\hat{W}_c \in \mathbb{R}^L$ is defined as

$$\dot{\hat{W}}_c \triangleq -\eta_{c1}\Gamma\frac{\omega}{\rho}\hat{\delta} - \eta_{c2}\Gamma\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i}{\rho_i}\delta_i. \tag{36}$$

The least squares gain matrix update law $\dot{\Gamma} \in \mathbb{R}^{L\times L}$ is defined as

$$\dot{\Gamma} \triangleq \left(\lambda\Gamma - \eta_{c1}\frac{\Gamma\omega\omega^\top\Gamma}{\rho^2} - \frac{\eta_{c2}\Gamma}{N}\sum_{i=1}^{N}\frac{\omega_i\omega_i^\top\Gamma}{\rho_i^2}\right)\cdot \mathbf{1}_{\{\underline{\Gamma}\leq\|\Gamma\|\leq\overline{\Gamma}\}}, \tag{37}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function ensuring that $\underline{\Gamma} \leq \|\Gamma\| \leq \overline{\Gamma}$ for all $t \in \mathbb{R}_{>0}$. The actor update law $\dot{\hat{W}}_a \in \mathbb{R}^L$ is defined as

$$\begin{aligned}\dot{\hat{W}}_a \triangleq{}& -\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a \\ &+ \frac{\eta_{c1}G_\phi^\top\hat{W}_a\omega^\top}{4\rho}\hat{W}_c + \eta_{c2}\frac{1}{N}\sum_{i=1}^{N}\frac{G_{\phi i}^\top\hat{W}_a\omega_i^\top}{4\rho_i}\hat{W}_c.\end{aligned} \tag{38}$$

The following assumption aids in the subsequent stability analysis by imposing a condition on sufficient richness of the Bellman regressor ω.

Assumption 6. *On the compact set, Ω, a finite set of off-trajectory points $\{x_i : x_i \in \Omega\}_{i=1}^N$ are user-selected such that $0 < \underline{c} \triangleq \inf_{t\in\mathbb{R}_{\geq 0}} \lambda_{\min}\left\{\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i\omega_i^\top}{\rho_i^2}\right\}$, where $\underline{c}$ is a constant scalar lower bound of the value of each history stack's minimum eigenvalues (Kamalapurkar et al., 2016).*

5. Stability analysis

To facilitate the stability analysis, let a concatenated state $z \in \mathbb{R}^{n+2L}$ be defined as $z \triangleq \left[x^\top, \tilde{W}_c^\top, \tilde{W}_a^\top\right]^\top$, and let the candidate Lyapunov function $V_L : \mathbb{R}^{n+2L} \to \mathbb{R}_{\geq 0}$ be defined as

$$V_L(z) \triangleq V^*(x) + \frac{1}{2}\tilde{W}_c^\top\Gamma^{-1}\tilde{W}_c + \frac{1}{2}\tilde{W}_a^\top\tilde{W}_a. \tag{39}$$

According to Kamalapurkar, Walters, et al. (2018, Lemma 4.3), (39) can generally be bounded as

$$\underline{v_l}(\|z\|) \leq V_L(z) \leq \overline{v_l}(\|z\|) \tag{40}$$

using class $\mathcal{K}$ functions $\underline{v_l}, \overline{v_l} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. To facilitate the subsequent analysis, let $v_l(\|z\|) = \frac{1}{2}\underline{q}(\|x\|) + \frac{1}{12}\eta_{c2}\underline{c}\left\|\tilde{W}_c\right\|^2 + \frac{1}{16}(\eta_{a1}+\eta_{a2})\left\|\tilde{W}_a\right\|^2$ and define $r \in \mathbb{R}_{>0}$ to be the prescribed radius of a compact ball $\mathcal{B}_r \in \mathbb{R}^{n+2L}$ centered at the origin where the convergence of z is desired.

Theorem 2. *Taking the time derivative of (39) yields*

$$\begin{aligned}\dot{V}_L \overset{a.a.t.}{\leq}{}& \nabla V^*\dot{x} - \tilde{W}_c^\top\Gamma^{-1}\dot{\hat{W}}_c - \tilde{W}_a^\top\dot{\hat{W}}_a \\ &- \frac{1}{2}\tilde{W}_c^\top\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c.\end{aligned} \tag{41}$$

Provided the weight update laws in (36)–(38) are implemented, Assumptions 1–6 hold, and the conditions

$$\eta_{a1} + \eta_{a2} > \frac{1}{\sqrt{\nu\underline{\Gamma}}}(\eta_{c1}+\eta_{c2})\overline{W}\overline{G_\phi} \tag{42}$$

$$\underline{c} > 3\frac{\eta_{a1}}{\eta_{c2}} + \frac{3(\eta_{c1}+\eta_{c2})^2\overline{W}^2}{8\eta_{c2}\nu\underline{\Gamma}}\left(\frac{\overline{G_\phi}^2}{2(\eta_{a1}+\eta_{a2})}\right) \tag{43}$$

$$l < v_l\left(\overline{v_l}^{-1}\left(\underline{v_l}(r)\right)\right) \tag{44}$$

$$\|z(0)\| \leq \overline{v_l}^{-1}\left(\underline{v_l}(r)\right) \tag{45}$$

are satisfied, where l is a positive constant that depends on the NN bounding constants in Assumption 5, then x, $\tilde{W}_c$, and $\tilde{W}_a$ are UUB. Hence, each control policy $\hat{u}$ converges to a neighborhood of its respective optimal control policy u^*.

Proof. Using (3), the HJB equation in (28), and the definitions of $G_R(x) \triangleq g(x)R^{-1}g(x)^\top$, $G_\phi(x) \triangleq \nabla\phi(x)G_R(x)\nabla\phi(x)^\top$, and $G_\epsilon(x) \triangleq \nabla\epsilon(x)G_R(x)\nabla\epsilon(x)^\top$, the time derivative in (41) can be written as

$$\begin{aligned}\dot{V}_L = {}& -Q(x) - \frac{1}{4}W^\top G_\phi W - \frac{1}{2}W^\top\nabla\phi^\top G_R\nabla\epsilon^\top - \frac{1}{4}G_\epsilon \\ &- \nabla V^* g u^* + \nabla V^* g u^* - \tilde{W}_c^\top\Gamma^{-1}\dot{\hat{W}}_c \\ &- \tilde{W}_a^\top\dot{\hat{W}}_a - \frac{1}{2}\tilde{W}_c^\top\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c.\end{aligned}$$

Substituting the NN optimal value function representation in (30), the NN optimal control policy representation in (31), and the approximate control policy in (33) yields

$$\begin{aligned}\dot{V}_L = {}& -Q(x) - \frac{1}{4}W^\top G_\phi W - \frac{1}{2}W^\top\nabla\phi^\top G_R\nabla\epsilon^\top - \frac{1}{4}G_\epsilon \\ &+ \frac{1}{2}\nabla\epsilon G_R\nabla\phi^\top\tilde{W}_a^\top + \frac{1}{2}W^\top\nabla\phi G_R\nabla\epsilon^\top + \frac{1}{2}G_\epsilon \\ &+ \frac{1}{2}W^\top G_\phi\tilde{W}_a^\top - \tilde{W}_c^\top\Gamma^{-1}\dot{\hat{W}}_c - \tilde{W}_a^\top\dot{\hat{W}}_a \\ &- \frac{1}{2}\tilde{W}_c^\top\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c.\end{aligned}$$

Substituting the weight update laws in (36)–(38) yields

$$\begin{aligned}\dot{V}_L = &-Q(x) - \frac{1}{4}W^\top G_\phi W + \frac{1}{2}W^\top G_\phi \tilde{W}_a \\ &+ \frac{1}{4}G_\epsilon + \frac{1}{2}\nabla\epsilon G_R \nabla\phi^\top \tilde{W}_a \\ &- \tilde{W}_c^\top\left(-\eta_{c1}\Gamma\frac{\omega}{\rho}\hat{\delta} - \eta_{c2}\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i}{\rho_i}\delta_i\right) \\ &- \tilde{W}_a^\top\left(-\eta_{a1}\left(\hat{W}_a - \hat{W}_c\right) - \eta_{a2}\hat{W}_a\right) \\ &- \tilde{W}_a^\top\left(\frac{\eta_{c1}G_\phi^\top \hat{W}_a\omega^\top}{4\rho}\hat{W}_c + \eta_{c2}\frac{1}{N}\sum_{i=1}^{N}\frac{G_{\phi i}^\top \hat{W}_a\omega_i^\top}{4\rho_i}\hat{W}_c\right) \\ &- \frac{1}{2}\tilde{W}_c^\top\left(\lambda\Gamma^{-1} - \eta_{c1}\frac{\Gamma\omega\omega^\top\Gamma}{\rho^2} - \eta_{c2}\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i\omega_i^\top}{\rho_i^2}\right)\tilde{W}_c.\end{aligned}$$

Substituting the BE in (35) and upper bounding terms using Assumptions 1–6 yields

$$\begin{aligned}\dot{V}_L \le &-\underline{q}(\|x\|) - \frac{1}{2}\eta_{c2}\underline{c}\left\|\tilde{W}_c\right\|^2 \\ &- (\eta_{a1}+\eta_{a2})\left\|\tilde{W}_a\right\|^2 + \frac{1}{4}\|G_\epsilon\| \\ &+ \left(\frac{1}{2}\overline{W}\|G_\phi\| + \frac{1}{2}\overline{\nabla\epsilon}\|G_R\|\overline{\nabla\phi}\right)\left\|\tilde{W}_a\right\| \\ &+ \left(+\eta_{c1}\overline{W}^2\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\|G_\phi\|\right)\left\|\tilde{W}_a\right\| \\ &+ \left(-\eta_{c2}\overline{W}^2\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\|G_\phi\| + \eta_{a2}\overline{W}\right)\left\|\tilde{W}_a\right\| \\ &+ \left(\eta_{c1}\frac{1}{2\sqrt{\nu\underline{\Gamma}}}\|O\| + \eta_{c2}\frac{1}{2\sqrt{\nu\underline{\Gamma}}}\|O\|\right)\left\|\tilde{W}_c\right\| \\ &+ \left(\eta_{c1}\overline{W}\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\|G_\phi\| + \eta_{c2}\overline{W}\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\|G_\phi\|\right)\left\|\tilde{W}_a\right\|^2 \\ &+ \left(\eta_{c2}\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\overline{W}\|G_\phi\| + \eta_{a1}\right)\left\|\tilde{W}_a\right\|\left\|\tilde{W}_c\right\| \\ &+ \left(\eta_{c1}\frac{1}{8\sqrt{\nu\underline{\Gamma}}}\overline{W}\|G_\phi\|\right)\left\|\tilde{W}_a\right\|\left\|\tilde{W}_c\right\|.\end{aligned}$$

Implementing nonlinear damping and substituting the gain conditions in (42) and (43) yields

$$\begin{aligned}\dot{V}_L \le &-\underline{q}(\|x\|) - \frac{1}{6}\eta_{c2}\underline{c}\left\|\tilde{W}_c\right\|^2 - \frac{1}{8}(\eta_{a1}+\eta_{a2})\left\|\tilde{W}_a\right\|^2 \\ &+ \frac{2a^2}{\eta_{a1}+\eta_{a2}} + \frac{3(\eta_{c1}+\eta_{c2})^2\|O\|^2}{8\nu\underline{\Gamma}\eta_{c2}\underline{c}} + \frac{1}{4}\|G_\epsilon\| \\ &+ \eta_{a2}\frac{1}{2}\overline{W}^2.\end{aligned}$$

By consolidating terms into v_l and l, the time derivative of (39) can be bounded as

$$\dot{V}_L \le -v_l(\|z\|) \quad \forall \|z\| \ge v_l^{-1}(l), \tag{46}$$

for all $t \in \mathbb{R}_{>0}$. Using (40) and (46), Khalil (2002, Theorem 4.18) can be invoked to conclude that every trajectory $z(t)$ that satisfies the initial condition $\|z(0)\| \le \overline{v_l}^{-1}\left(\underline{v_l}(r)\right)$ is bounded for all $t \in \mathbb{R}$, z is UUB such that $\limsup_{t\to\infty}\|z\| \le \underline{v_l}^{-1}\left(\overline{v_l}\left(v_l^{-1}(l)\right)\right)$, and the control policy $\hat{u}$ converges to a neighborhood of the optimal control policy u^*. Since $z \in \mathcal{L}_\infty$, it follows that $x, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty$; hence, $x, \hat{W}_c, \hat{W}_a \in \mathcal{L}_\infty$ and $u \in \mathcal{L}_\infty$. Additionally, every trajectory z that is initialized in the ball $\mathcal{B}_r$ is bounded such that $z \in \mathcal{B}_r$, $\forall t \in \mathbb{R}_{\ge 0}$. Since $z \in \mathcal{B}_r$, the states $x, \hat{W}_c, \hat{W}_a$ similarly lie in a compact set.

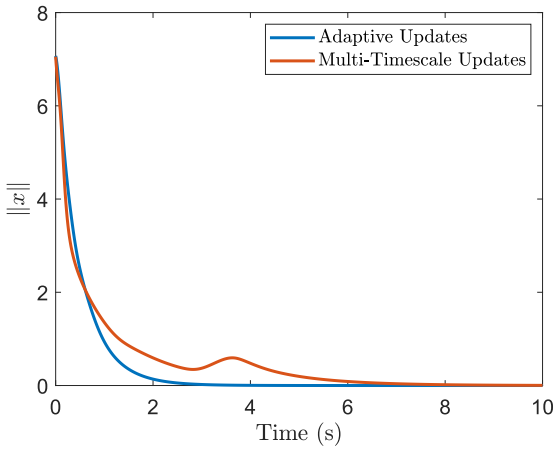

Fig. 1. Comparative plots of the regulation error norm $\|x\|$ for the developed method consisting of adaptive updates of all the DNN layers compared to the previous method consisting of multi-timescale updates of the DNN.

6. Simulations

To demonstrate the effectiveness of the developed ADP technique, comparative simulations are performed on a control-affine nonlinear dynamical system with a two dimensional state $x = [x_1, x_2]^\top$. The developed method results are compared with the multi-timescale DNN technique in Greene et al. (2023) as the baseline. For the baseline method, the inner-layer weights are retrained and updated once online, and the mean squared error is used as the loss function for training.

For value function approximation, the basis function is selected as $\phi = \left[x_1^2, x_1x_2, x_2^2\right]$. The initial conditions for the system are $x(0) = [-5, 5]^\top$, $\Gamma(0) = 100 \cdot I_{3\times 3}$, and $\hat{W}_c(0) = \hat{W}_a(0) = 0.01 \cdot \mathbf{1}_{3\times 1}$. The system dynamics in (3) are

$$f = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1 & x_2\left(1 - (\cos(2x_1)+2)^2\right)\end{bmatrix}\theta,$$

$$g = \begin{bmatrix} 0 \\ \cos(2x_1)+2 \end{bmatrix},$$

where $\theta = [-1, 1, -0.5, -0.5]^\top$ (Vrabie et al., 2013). The simulation parameters are selected as $\eta_{c1} = 0.005$, $\eta_{c2} = 0.1$, $\eta_{a1} = 15$, $\eta_{a2} = 0.1$, $\lambda = 0.4$, $\nu = 0.005$, $N = 100$, $\Gamma_\theta = 0.02 \cdot I_{L\times L}$, $\alpha_1 = 30$, $k_f = 100$, $\beta_f = 0.2$, $k_\theta = 0.001$. The cost parameters in (4) are selected as $Q = x^T \text{diag}([.01,\ 3])\,x$ and $R = .6$. The optimal value function for the system is given by $V^*(x) = \frac{1}{2}x_1^2 + x_2^2$ (Vrabie et al., 2013). The implemented DNN contains 7 hidden layers with 7 neurons in each layer.[4]

Fig. 1 presents the state errors of the infinite horizon regulation problem. It is shown that using the developed system identifier to learn the dynamics in real-time successfully yields faster convergence of the system states. The state error rapidly converges to steady state at approximately 3 s, whereas it takes approximately 7 s for the state errors to converge with the baseline method.

Fig. 2 shows the comparative plots of the RMS function approximation error norm with the developed and baseline method.

[4] The time and memory computational complexity of the DNN forward and backward pass is $O(kL^2)$, where k is the number of layers and L is the number of neurons in each layer. The computational complexity grows linearly in depth and quadratically in width. Note that kL^2 is also approximately the total number of individual weights in the DNN. Therefore, it is the total number of weights in the DNN that decides the computational complexity, and a shallow neural network consumes equal computational resources as a DNN with an equal number of total individual weights.

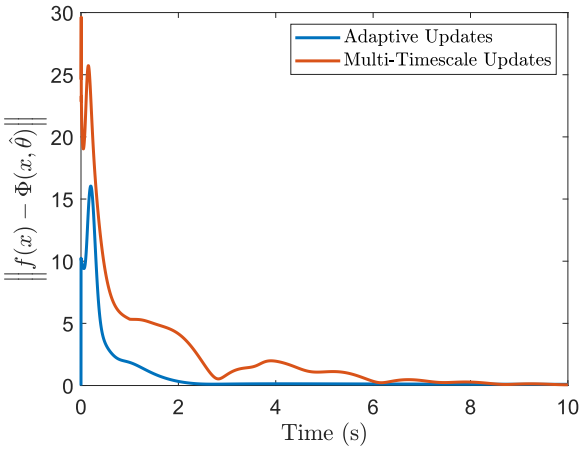

Fig. 2. Comparative plots of the RMS function approximation error norm $\left\|f(x)-\Phi\left(X,\hat{\theta}\right)\right\|$ for the developed method consisting of adaptive updates of all the DNN layers compared to the previous method consisting of multi-timescale updates of the DNN.

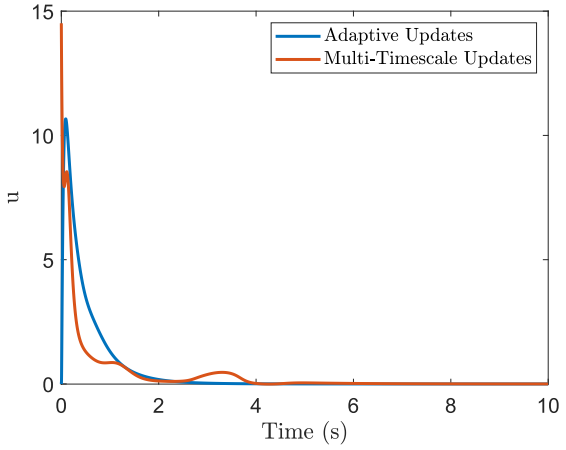

Fig. 3. Comparative plots of the control input $\|u\|$ for the developed method consisting of adaptive updates of all the DNN layers compared to the previous method consisting of multi-timescale updates of the DNN.

The simultaneous update on all weights results in a rapid function approximation error convergence with the developed method, i.e., within 3 s. In contrast, the baseline method does not yield function approximation error convergence until after 8 s. The developed method results in an RMS error of 2.407 while the baseline method results in an RMS error of 4.800. The RMS function approximation error is shown to decrease by 49.85% when using the developed method compared to the baseline. The improved learning of the dynamics is beneficial to the ADP framework as it provides a more accurate model to be used in BE extrapolation which results in faster convergence to the optimal control policy as shown subsequently in Fig. 3.

The aforementioned control objective is to find an optimal control policy u that minimizes the cost function. Fig. 3 shows that the control policy reaches convergence faster in the developed method before 3 s, while the baseline method does not reach convergence until approximately 4 s.

Fig. 4 shows that the actor and critic weight estimates for the value function and control policy are bounded and converge. Additionally, Fig. 5 shows the value function approximation $\hat{V}$ learned by the critic NN compared to the optimal value function V^*.

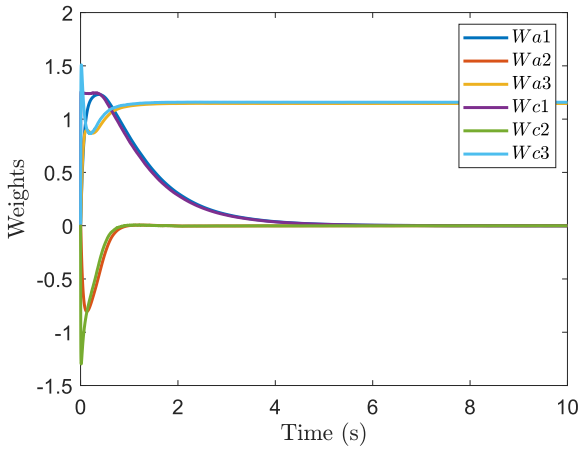

Fig. 4. Evolution of value function and control policy weights.

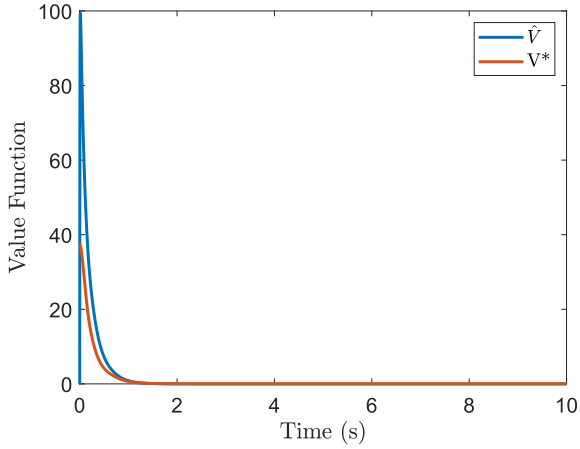

Fig. 5. Comparative plots of the approximated value function compared to the optimal value function.

An additional simulation is performed on a third-order dynamical system represented as

$$f = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1^3 & x_2\left(1-(cos\,(2x_1)+2)^2\right) \end{bmatrix}\theta,$$

$$g = \begin{bmatrix} 0 \\ cos\,(2x_1)+2 \end{bmatrix},$$

where $\theta = [-1, 1, -0.5, -0.5]^\top$, to demonstrate the efficacy of the proposed DNN on a more complex, higher-order system. Fig. 6 shows that the function approximation error of the developed DNN converges within 2 s, and a shallow NN (SNN) consisting of only one layer takes approximately 6 s to learn the complex dynamics.

7. Conclusion

The developed method uses an adaptive Lb-DNN system identifier in conjunction with a RISE-based dynamics observer within an ADP framework. A least squares continuous-time update law is used to update all layers of DNN weights online. The system identifier is used to obtain an estimate of the unknown system dynamics. Exponential convergence to a neighborhood of the DNN weight estimation error, provided the Jacobian of the DNN satisfies the PE condition, is shown via a Lyapunov-based stability analysis. The entire system is shown to be UUB such that the

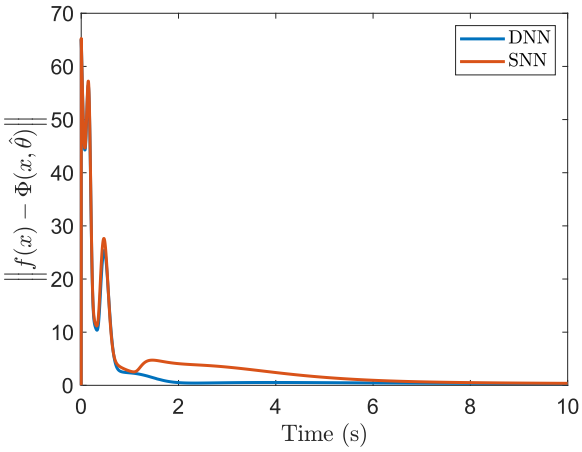

Fig. 6. Comparative plots of the RMS function approximation error norm $\left\|f(x) - \Phi\left(X, \hat{\theta}\right)\right\|$ for the developed DNN method compared to a SNN.

developed control policy is shown to converge to a neighborhood of the optimal control policy. Simulation results show that the adaptive DNN yields 49.85% improved function approximation error in comparison to the previously developed multi-timescale DNN and faster convergence of the RMS regulation error, RMS controller error, and RMS function approximation error.

Appendix

DNNs are known to approximate any given continuous function on a compact set, based on the universal function approximation theorem (Kidger & Lyons, 2020). Although various DNN architectures can be used, a fully-connected DNN is described here as an example. Let $\sigma \in \mathbb{R}^{L_{\text{in}}}$ denote the DNN input with size $L_{\text{in}} \in \mathbb{Z}_{>0}$, and $\theta \in \mathbb{R}^p$ denote the vector of DNN parameters (i.e., weights and bias terms) with size $p \in \mathbb{Z}_{>0}$. Then, a fully-connected feedforward DNN $\Phi(\sigma, \theta)$ with output size $L_{\text{out}} \in \mathbb{Z}_{>0}$ is defined using a recursive relation $\Phi_j \in \mathbb{R}^{L_{j+1}}$ modeled as

$$\Phi_j \triangleq \begin{cases} V_j^\top \phi_j\left(\Phi_{j-1}\right), & j \in \{1, \ldots, k\}, \\ V_j^\top \sigma_a, & j = 0, \end{cases} \tag{A.1}$$

where $\Phi(\sigma, \theta) = \Phi_k$, and $\sigma_a \triangleq \begin{bmatrix} \sigma^\top & 1 \end{bmatrix}^\top$ denotes the augmented input that accounts for the bias terms, $k \in \mathbb{Z}_{>0}$ denotes the total number of hidden layers, $V_j \in \mathbb{R}^{L_j \times L_{j+1}}$ denotes the matrix of weights and biases, $L_j \in \mathbb{Z}_{>0}$ denotes the number of nodes in the jth layer for all $j \in \{0, \ldots, k\}$ with $L_0 \triangleq L_{\text{in}} + 1$ and $L_{k+1} = L_{\text{out}}$. The vector of smooth activation functions is denoted by $\phi_j : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j}$ for all $j \in \{1, \ldots, k\}$. If the DNN involves multiple types of activation functions at each layer, then ϕ_j may be represented as $\phi_j \triangleq \begin{bmatrix} \varsigma_{j,1} & \cdots & \varsigma_{j,L_j - 1} & 1 \end{bmatrix}^\top$, where $\varsigma_{j,p} : \mathbb{R} \to \mathbb{R}$ denotes the activation function at the p^{th} node of the j^{th} layer. For the DNN architecture in (A.1), the vector of DNN weights is $\theta \triangleq \begin{bmatrix} \text{vec}(V_0)^\top & \ldots & \text{vec}(V_k)^\top \end{bmatrix}^\top$ with size $p = \Sigma_{j=0}^{k} L_j L_{j+1}$. The Jacobian of the activation function vector at the j^{th} layer is denoted by $\phi_j' : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j \times L_j}$, and $\phi_j'(y) \triangleq \frac{\partial}{\partial z}\phi_j(z)\big|_{z=y}$, $\forall y \in \mathbb{R}^{L_j}$. Let the Jacobian of the DNN with respect to the weights be denoted by $\Phi'(\sigma, \theta) \triangleq \frac{\partial}{\partial \theta}\Phi(\sigma, \theta)$, which can be represented using $\Phi'(\sigma, \theta) = \begin{bmatrix} \Phi_0', & \Phi_1', & \ldots, & \Phi_k' \end{bmatrix}$, where $\Phi_j' \triangleq \frac{\partial}{\partial \text{vec}(V_j)}\Phi(\sigma, \theta)$ for all $j \in \{0, \ldots, k\}$. Then, using (A.1) and the property of the vectorization operator in (2) yields

$$\Phi_0' = \left(\prod_{l=1}^{\overset{\curvearrowleft}{k}} \hat{V}_l^\top \phi_l'\left(\Phi_{l-1}\right)\right)\left(I_{L_1} \otimes \sigma_a^\top\right), \tag{A.2}$$

and

$$\Phi_j' = \left(\prod_{l=j+1}^{\overset{\curvearrowleft}{k}} \hat{V}_l^\top \phi_l'\left(\Phi_{l-1}\right)\right)\left(I_{L_{j+1}} \otimes \phi_j^\top\left(\Phi_{j-1}\right)\right), \tag{A.3}$$

for all $j \in \{1, \ldots, k\}$.

References

Bernstein, D. S. (2009). *Matrix mathematics*. Princeton University Press.

Deptula, P., Bell, Z., Doucette, E., Curtis, W. J., & Dixon, W. E. (2020). Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with control effectiveness faults. *Automatica*, *116*, 1–10.

Deptula, P., Bell, Z., Zegers, F., Licitra, R., & Dixon, W. E. (2021). Approximate optimal influence over an agent through an uncertain interaction dynamic. *Automatica*, *134*, 1–13.

Gao, W., Mynuddin, M., Wunsch, D. C., & Jiang, Z.-P. (2021). Reinforcement learning-based cooperative optimal output regulation via distributed adaptive internal model. *IEEE Transactions on Neural Networks and Learning Systems*.

Greene, M., Bell, Z., Nivison, S., & Dixon, W. E. (2023). Deep neural network-based approximate optimal tracking for unknown nonlinear systems. *IEEE Transactions on Automatic Control*, *68*(5), 3171–3177.

Griffis, E., Patil, O., Bell, Z., & Dixon, W. E. (2023). Lyapunov-based long short-term memory (Lb-LSTM) neural network-based control. *IEEE Control Systems Letters*, *7*, 2976–2981.

Griffis, E., Patil, O., Hart, R., & Dixon, W. E. (2024). Lyapunov-based long short-term memory (Lb-LSTM) neural network-based adaptive observer. *IEEE Control Systems Letters*, *8*, 97–102.

Griffis, E., Patil, O., Makumi, W., & Dixon, W. E. (2023). Deep recurrent neural network-based observer for uncertain nonlinear systems. In *IFAC world congr.*.

Hart, R., Griffis, E., Patil, O., & Dixon, W. E. (2024). Lyapunov-based physics-informed long short-term memory (LSTM) neural network-based adaptive control. *IEEE Control Systems Letters*, *8*, 13–18.

Hart, R., Patil, O., Griffis, E., & Dixon, W. E. (2023). Deep Lyapunov-based physics-informed neural networks (DeLb-PINN) for adaptive control design. In *Proc. IEEE conf. decis. control*.

Ioannou, P., & Sun, J. (1996). *Robust adaptive control*. Prentice Hall.

Isaly, A., Patil, O., Sweatland, H., Sanfelice, R., & Dixon, W. E. (2024). Adaptive safety with a RISE-based disturbance observer. *IEEE Transactions on Automatic Control*.

Kamalapurkar, R., Andrews, L., Walters, P., & Dixon, W. E. (2017). Model-based reinforcement learning for infinite-horizon approximate optimal tracking. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 753–758.

Kamalapurkar, R., Klotz, J. R., Walters, P., & Dixon, W. E. (2018). Model-based reinforcement learning for differential graphical games. *IEEE Transactions on Control of Network Systems*, *5*(1), 423–433.

Kamalapurkar, R., Walters, P., & Dixon, W. E. (2016). Model-based reinforcement learning for approximate optimal regulation. *Automatica*, *64*, 94–104.

Kamalapurkar, R., Walters, P. S., Rosenfeld, J. A., & Dixon, W. E. (2018). *Reinforcement learning for optimal feedback control: A Lyapunov-based approach*. Springer.

Khalil, H. K. (2002). *Nonlinear Systems* (3rd ed.). Prentice Hall.

Kidger, P., & Lyons, T. (2020). Universal approximation with deep narrow networks. In *Conf. learn. theory* (pp. 2306–2327).

Le, D. M., Patil, O. S., Nino, C. F., & Dixon, W. E. (2024). Accelerated gradient approach for deep neural network-based adaptive control of unknown nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. http://dx.doi.org/10.1109/TNNLS.2024.3395064.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(436444).

Lewis, F. L., & Liu, D. (2013). *Reinforcement learning and approximate dynamic programming for feedback control: Vol. 17*, John Wiley & Sons.

Liberzon, D. (2012). *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press.

Makumi, W., Bell, Z., & Dixon, W. E. (2023). Approximate optimal indirect regulation of an unknown agent with a Lyapunov-based deep neural network. *IEEE Control Systems Letters*, *7*, 2773–2778.

Makumi, W., Bell, Z., & Dixon, W. E. (2024). Cooperative approximate optimal indirect regulation of uncooperative agents with Lyapunov-based deep neural network. In *AIAA sciTech*.

Makumi, W., Greene, M., Bell, Z., Bialy, B., Kamalapurkar, R., & Dixon, W. E. (2023). Hierarchical reinforcement learning and gain scheduling-based control of a hypersonic vehicle. In *AIAA sciTech*.

Makumi, W., Greene, M., Bell, Z., Nivison, S., Kamalapurkar, R., & Dixon, W. E. (2023). Hierarchical reinforcement learning-based supervisory control of unknown nonlinear systems. In *IFAC world congr.*.

Modares, H., Lewis, F. L., Kang, W., & Davoudi, A. (2018). Optimal synchronization of heterogeneous nonlinear systems with unknown dynamics. *IEEE Transactions on Automatic Control*, *63*(1), 117–131.

Paden, B. E., & Sastry, S. S. (1987). A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators. *IEEE Transactions on Circuits and Systems*, *34*(1), 73–82.

Pang, B., & Jiang, Z.-P. (2020). Adaptive optimal control of linear periodic systems: An off-policy value iteration approach. *IEEE Transactions on Automatic Control*, *66*(2), 888–894.

Patil, O. S., Griffis, E. J., Makumi, W. A., & Dixon, W. E. (2023). Composite adaptive Lyapunov-based deep neural network (Lb-DNN) controller. arXiv preprint arXiv:2311.13056.

Patil, O., Isaly, A., Xian, B., & Dixon, W. E. (2022). Exponential stability with RISE controllers. *IEEE Control Systems Letters*, *6*, 1592–1597.

Patil, O., Le, D., Greene, M., & Dixon, W. E. (2022). Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network. *IEEE Control Syst Letters*, *6*, 1855–1860.

Patil, O. S., Le, D. M., Griffis, E., & Dixon, W. E. (2022). Deep residual neural network (ResNet)-based adaptive control: A Lyapunov-based approach. In *Proc. IEEE conf. decis. control*.

Philor, J., Makumi, W., Bell, Z., & Dixon, W. E. (2024). Approximate optimal indirect control of an unknown agent within a dynamic environment using a Lyapunov-based deep neural network. In *Proc. am. control conf.*.

Shevitz, D., & Paden, B. (1994). Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on Automatic Control*, *39 no. 9*, 1910–1914.

Slotine, J. J., & Li, W. (1989). Composite adaptive control of robot manipulators. *Automatica*, *25*(4), 509–519.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT Press.

Vamvoudakis, K. G., Vrabie, D., & Lewis, F. L. (2014). Online adaptive algorithm for optimal control with integral reinforcement learning. *International Journal of Robust and Nonlinear Control*, *24*(17), 2686–2710.

Vrabie, D., Pastravanu, O., Abu-Khalaf, M., & Lewis, F. L. (2009). Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, *45*(2), 477–484.

Vrabie, D., Vamvoudakis, K. G., & Lewis, F. L. (2013). *optimal adaptive control and differential games by reinforcement learning principles*. The Institution of Engineering and Technology.

Walters, P., Kamalapurkar, R., Voight, F., Schwartz, E., & Dixon, W. E. (2018). Online approximate optimal station keeping of a marine craft in the presence of an irrotational current. *IEEE Transactions on Robotics*, *34*(2), 486–496.

Wang, N., Gao, Y., Zhao, H., & Ahn, C. K. (2020). Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(7), 3034–3045.

Wanjiku A. Makumi received her B.S. from the Joint Department of Biomedical Engineering at NC State University and UNC Chapel Hill. In 2020, she joined the Nonlinear Controls and Robotics Laboratory at the University of Florida under the guidance of Dr. Warren Dixon where she received her M.S. in mechanical engineering and her Ph.D. in aerospace engineering in 2021 and 2024, respectively. In 2025 she was awarded the Best Dissertation Award for the Department of Mechanical and Aerospace Engineering. She is a recipient of the DoD SMART Scholarship and is currently a research engineer at the Air Force Research Laboratory. Her research focuses on machine learning and adaptive control for unknown nonlinear systems.

Omkar Sudhir Patil received his Bachelor of Technology (B.Tech.) degree in production and industrial engineering from Indian Institute of Technology (IIT) Delhi in 2018, where he was honored with the BOSS award for his outstanding bachelor's thesis project. In 2019, he joined the Nonlinear Controls and Robotics (NCR) Laboratory at the University of Florida under the guidance of Dr. Warren Dixon to pursue his doctoral studies. Omkar received his Master of Science (M.S.) degree in mechanical engineering in 2022 and Ph.D. in mechanical engineering in 2023 from the University of Florida. During his Ph.D. studies, he was awarded the Graduate Student Research Award for outstanding research. In 2023, he started working as a postdoctoral research associate at NCR Laboratory, University of Florida. His research focuses on the development and application of innovative Lyapunov-based nonlinear, robust, and adaptive control techniques.

Warren E. Dixon received his Ph.D. from the Department of Electrical and Computer Engineering from Clemson University. He worked as a research staff member and Eugene P. Wigner Fellow at Oak Ridge National Laboratory (ORNL) until 2004, when he joined the University of Florida in the Mechanical and Aerospace Engineering Department, where he currently serves as a Distinguished Professor and Department Chair. His main research interest has been the development and application of Lyapunov-based control techniques for uncertain nonlinear systems. His work has been recognized by various awards, including: the 2019 IEEE Control Systems Technology Award, 2015 & 2009 American Automatic Control Council (AACC) O. Hugo Schuck (Best Paper) Awards, the 2013 Fred Ellersick Award for Best Overall MILCOM Paper, the 2011 American Society of Mechanical Engineers (ASME) Dynamics Systems and Control Division Outstanding Young Investigator Award, and the 2006 IEEE Robotics and Automation Society (RAS) Early Academic Career Award. He is an ASME Fellow (2016) and IEEE Fellow (2016), and his technical contributions and service to the IEEE CSS were recognized by the IEEE CSS Distinguished Member Award (2020). He was awarded the Air Force Commander's Public Service Award (2016) for his contributions to the U.S. Air Force Science Advisory Board.