# NEURAL-NETWORK BASED ONLINE POLICY ITERATION FOR CONTINUOUS-TIME INFINITE-HORIZON OPTIMAL CONTROL OF NONLINEAR SYSTEMS

*Difan Tang, Lei Chen, and Zhao Feng Tian*

School of Mechanical Engineering, The University of Adelaide, Australia

## ABSTRACT

A new policy-iteration algorithm using neural networks (NNs) is proposed in this paper to synthesize optimal control laws online for continuous-time nonlinear systems. Latest advances in this field realize synchronous policy iteration but meanwhile require an additional tuning loop or a logic switch mechanism to maintain system closed-loop stability. A new algorithm is thus derived in this paper to address this limitation. The optimal control law is found by solving the Hamilton-Jacobi-Bellman (HJB) equation for the associated value function via synchronous policy iteration in a critic-actor configuration. As a major contribution, a new form of NN approximation for the value function is proposed, offering the closed-loop system asymptotic stability without additional tuning scheme or logic switch mechanism. As a second contribution, an extended Kalman filter (EKF) is introduced to estimate the critic NN parameters for fast convergence. The efficacy of the new algorithm is verified by simulations.

***Index Terms*** **—** machine learning, neural network, policy iteration, optimal control, nonlinear system

## 1. INTRODUCTION

Optimal control for nonlinear systems involves solving a Hamilton-Jacobi-Bellman (HJB) equation. This differential equation is nonlinear and difficult to solve directly. As an alternative, the so-called 'policy iteration' can be used [1, 2], which is basically a two-step iteration between policy evaluation and policy improvement. The term 'policy' is specifically used in the field of dynamic programming [1] and refers to a control strategy or control law. By repeating these two steps, the initial non-optimal control strategy evolves to an optimal one. To implement policy iteration, the value function in the HJB equation needs to be approximated by a suitable agent. Neural networks (NNs), possessing universal approximation properties, are ideal candidates [3].

An early approach for policy iteration based on NNs is an offline method that takes control saturation into account [4]. The control law obtained is nonlinear and optimal in respect to saturated actuators, outperforming its linear counterpart, the linear-quadratic regulator (LQR), which is only optimal when actuators are not saturated. With some modification, this algorithm can be put online and eliminates the need for knowledge of system internal dynamics [5]. The associated policy iteration then becomes a sequential process that takes place with one step starting on completion of the previous step, resulting in discrete update of the control strategy. To initialize the process, an initial stabilizing control law must be specified. Discontinuities in control should be smoothed by appropriate methods but are nonetheless not considered. Though the impact of using a discount factor for the infinite-horizon cost is further discussed in [6], the limitations in [5] are not addressed. Comparatively, a synchronous policy-iteration technique in [7] offers more advantages. Instead of the step-by-step iteration, both the two steps are performed simultaneously and continuously, contributing to continuous update of the control law and hence smoother control. The critic-actor structure constructed by two NNs guarantees the stability of the entire closed-loop system during NN online tuning without the necessity of providing an initial stabilising control strategy.

The theory framework of the online synchronous policy iteration method for synthesizing optimal control in nonlinear systems have been enormously enriched by recent and latest advances in dealing with more complicated uncertainties and nonlinearities as well as performance improvement. To tackle uncertainties and nonlinearities, one may need techniques of partial/complete model-free design [8-10], NN-based states estimation [11], optimal disturbance rejection [12, 13], robust optimal control with upper-bounded costs [14], and optimal control under actuator saturation [10, 15]. To further improve algorithm performance, one may consider faster online tuning [9], handling finite approximation errors that sometimes may prevent proper convergence of the value function and the associated optimal control [16], or gaining more insights into the mechanism of the policy iteration, its convergence, uniqueness of the solution, and the sufficient conditions [17]. However, to guarantee the stability of the closed-loop system, the available online synchronous policy iteration methods either require an additional tuning loop for the actor NN, or rely on a logic algorithm to switch between different tuning modes. The additional tuning can bring more uncertainties into the system, and the logic switch mechanism can cause discontinuities in control.

The work in this paper thus derives a new synchronous policy iteration method without actor NN tuning or logic switch mechanism but still capable of maintaining closed-loop system stability. As a major contribution, a new form of value function approximation based on a single-layer NN is proposed. Similar to the aforementioned methods, the critic-actor NN configuration is adopted. But differently, only the critic NN needs to be tuned, via an extended Kalman filter

(EKF), with faster parameter convergence than traditional gradient-based methods. This is another contribution of this paper because none of the aforementioned methods use the EKF for NN tuning to solve for optimal control in nonlinear systems.

## 2. CONTINUOUS-TIME HJB EQUATION AND POLICY ITERATION

The continuous-time nonlinear system below is considered:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t)); \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1}$$

the control-affine dynamics of which is:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{g}(\boldsymbol{x}(t))\boldsymbol{u}(\boldsymbol{x}(t)); \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ denotes system states; $\boldsymbol{u}(t) \in \mathbb{R}^m$ refers to control inputs; $\boldsymbol{f}(\boldsymbol{x}(t)) \in \mathbb{R}^n$ describes system internal dynamics; $\boldsymbol{g}(\boldsymbol{u}(t)) \in \mathbb{R}^{n\times m}$ represents control input dynamics.

**Assumption 1**: $\boldsymbol{f}(0) = 0$; $\boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{g}(\boldsymbol{x}(t))\boldsymbol{u}(t)$ is Lipschitz continuous on a set $\Omega \subseteq \mathbb{R}^n$ containing the origin; the system as in Eqs. (1) and (2) can be stabilised by $\boldsymbol{u} \in \psi(\Omega)$ that are admissible [18].

**Assumption 2**: $\|\boldsymbol{f}(\boldsymbol{x}(t))\| \le b_f \|\boldsymbol{x}(t)\|$, $\|\boldsymbol{g}(\boldsymbol{x}(t))\| \le b_g$, where constants $b_f \in \mathbb{R}^+$, and $b_g \in \mathbb{R}^+$ are known.

The control problem is to determine a control policy $\boldsymbol{u}(t)$ to minimise the following performance index (cost function):

$$V(\boldsymbol{x}_0) = \int_0^\infty [\bar{Q}(\boldsymbol{x}(\tau)) + \bar{U}(\boldsymbol{u}(\tau))] d\tau, \tag{3}$$

with $\bar{Q}(\boldsymbol{x}(t))$ and $\bar{U}(\boldsymbol{u}(t)) = \boldsymbol{u}^{\mathrm{T}}(t)\boldsymbol{R}\boldsymbol{u}(t)$ being positive-definite monotonically increasing functions, in which $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ is a positive-definite weighting matrix.

Differentiating Eq. (3) yields its infinitesimal version that is a nonlinear Lyapunov equation (LE), written as:

$$\boldsymbol{V}_x^{\mathrm{T}}(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})] + \bar{Q}(\boldsymbol{x}) + \bar{U}(\boldsymbol{u}) = 0, \tag{4}$$

with $V(0) = 0$ .

Let $V^*(\boldsymbol{x})$ denote the optimal (minimal) cost function, which is named as the 'value function', and its derivative $\boldsymbol{V}_x^*(\boldsymbol{x}) \triangleq \partial V^*(\boldsymbol{x}) / \partial \boldsymbol{x}$, and then the corresponding optimal control policy is given by:

$$\boldsymbol{u}^*(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{V}_x^*(\boldsymbol{x}), \tag{5}$$

which satisfies the following Hamilton–Jacobi–Bellman (HJB) equation based on Eq. (4):

$$\bar{Q}(\boldsymbol{x}) + \bar{U}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \frac{1}{4}\left[\boldsymbol{V}_x^*(\boldsymbol{x})\right]^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{V}_x^*(\boldsymbol{x}) = 0. \tag{6}$$

That is, by solving Eq. (6) for $V^*(\boldsymbol{x})$, the optimal control policy can then be obtained as in Eq. (5) given that the system internal dynamics $\boldsymbol{f}(\boldsymbol{x})$ and control input dynamics $\boldsymbol{g}(\boldsymbol{x})$ are known.

Note that the HJB equation is nonlinear and difficult to solve directly. Instead, it can be solved recursively through the successive approximation method introduced in [19], which is generally recognised as a policy iteration approach [2]. Despite different forms of realisation, the policy iteration algorithm basically involves two basic steps – policy evaluation (solving for $V^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$ associated with $\boldsymbol{u}^{(i)}(\boldsymbol{x})$) and policy improvement (updating $\boldsymbol{u}^{(i+1)}(\boldsymbol{x})$ according to $\boldsymbol{V}_x^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$). Starting with an initial admissible control policy $\boldsymbol{u}^{(0)}(\boldsymbol{x})$, the algorithm proceeds until convergence is reached at $V^*(\boldsymbol{x})$ and $\boldsymbol{u}^*(\boldsymbol{x})$.

## 3. NN-BASED VALUE FUNCTION APPROXIMATION

Note that solving for $V^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$ in a direct way is difficult. To allow implementation of the policy iteration, an appropriately structured representation of $V^*(\boldsymbol{x})$ is necessary, which can be a neural-network approximation. Unlike the methods of previous studies discussed in Section I, we propose a new form of value function approximation as:

$$V^*(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}), \tag{7}$$

where $\phi(\cdot) = [\phi_1(\boldsymbol{x}), \cdots, \phi_N(\boldsymbol{x})]^{\boldsymbol{T}} : \mathbb{R}^n \to \mathbb{R}^N$ is a set of NN activation functions which are nonlinear; $\boldsymbol{W} \in \mathbb{R}^N$ is a vector of ideal NN weights; $\boldsymbol{P} \in \mathbb{R}^{n_x \times n_x}$ is a positive-definite matrix; $\varepsilon(\boldsymbol{x}) \in \mathbb{R}$ is the approximation error. The approximation error $\varepsilon(\boldsymbol{x})$ can be arbitrarily small with a sufficient number of hidden layer neurons $\phi(\boldsymbol{x})$ [4].

The derivative of $V^*(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is:

$$\boldsymbol{V}_x^*(\boldsymbol{x}) \triangleq \frac{\partial V^*(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{P}\boldsymbol{x} + \nabla\phi^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{W} + \nabla\varepsilon(\boldsymbol{x}), \tag{8}$$

where $\nabla\phi(\boldsymbol{x}) \triangleq \left[\frac{\partial \phi(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]^{\mathrm{T}}$ denotes the gradient of $\phi(\boldsymbol{x})$.

**Assumption 3**: For constants $b_\phi \in \mathbb{R}^+$ and $b_\varepsilon \in \mathbb{R}^+$, there exist $\|\nabla\phi(\boldsymbol{x})\| \le b_\phi \|\boldsymbol{x}\|$ and $\|\nabla\varepsilon(\boldsymbol{x})\| \le b_\varepsilon \|\boldsymbol{x}\|$.

Due to the approximation error $\varepsilon(\boldsymbol{x})$, the associated control law under this approximation scheme is a nearly optimal control as:

$$\boldsymbol{u}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\left[\boldsymbol{P}\boldsymbol{x} + \nabla\phi^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{W}\right]. \tag{9}$$

Eq. (7) is a single-layer NN, which is nonlinear in the hidden layer $\phi(\boldsymbol{x})$ but linear in the output layer weights $\boldsymbol{W}$. To implement policy iteration, $\boldsymbol{W}$ need to be tuned dynamically so that Eq. (7) approximates a target value function. Let $\hat{\boldsymbol{W}}$ be the estimate of ideal weights, then:

$$\hat{V}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \hat{\boldsymbol{W}}^{\mathrm{T}}\phi(\boldsymbol{x}). \tag{10}$$

The resulting nonlinear LE then becomes:

$$\begin{aligned}\left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\nabla\phi(\boldsymbol{x})\right][\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{u}}(\boldsymbol{x})] \\ + Q(\boldsymbol{x}) + U(\hat{\boldsymbol{u}}) = \varepsilon_H + \xi\end{aligned}, \tag{11}$$

where $\varepsilon_H$ is the difference caused by the approximation error $\varepsilon(\boldsymbol{x})$ as in Eq.(7), and $\xi$ is the error due to imperfect weight estimate during a tuning process.

An appropriate tuning algorithm is now needed for uniform convergence of $\hat{\boldsymbol{W}}$ to the ideal $\boldsymbol{W}$ so that $\xi$ is minimised. In this paper, an extended Kalman filter (EKF) is used for NN weights tuning/estimation. Since $\hat{\boldsymbol{W}}$ is the parameter vector to estimate, Eq. (11) can be rearranged in the following form:

$$\begin{cases} \dot{\boldsymbol{W}} = \boldsymbol{w} \\ y(t) = h(t, \boldsymbol{W}, \varepsilon_H) + v \end{cases}, \tag{12}$$

where $\boldsymbol{w}$ and $v$ are white-noise inputs with covariance matrix $\boldsymbol{Q}_f \succ 0$ and $\boldsymbol{R}_f \succ 0$ , respectively,

$$y(t) = -\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})] - \bar{Q}(\boldsymbol{x}) - \bar{U}(\boldsymbol{u}),$$

and $h(t, \boldsymbol{W}, \varepsilon_H) = \boldsymbol{W}^{\mathrm{T}}\nabla\phi(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})] - \varepsilon_H$.

Note that $y(t)$ is known from measurements. The unknown ideal $\boldsymbol{W}$ needs to be estimated according to $y(t)$ and the partially known dynamics $h(t, \boldsymbol{W}, \varepsilon_H)$. White-noise inputs $\boldsymbol{w}$ and $v$ are artificially introduced in order to implement EKF.

Introducing EKF into the system described by Eq. (12) yields:

$$\begin{cases} \dot{\hat{\boldsymbol{W}}} = \boldsymbol{K}_f \ y(t) - \hat{y}(t) \ + \boldsymbol{w} \\ \hat{y}(t) = h(t, \hat{\boldsymbol{W}}, \varepsilon_H) + v \end{cases}, \tag{13}$$

where $\dot{\hat{\boldsymbol{W}}}$ and $\hat{y}(t)$ denote the estimate of $\boldsymbol{W}$ and corresponding output, $\boldsymbol{K}_f \in \mathbb{R}^{N\times 1}$ is the EKF gain.

The EKF gain $\boldsymbol{K}_f$ can be computed from:

$$\boldsymbol{K}_f = \boldsymbol{S}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_f^{-1}, \tag{14}$$

with

$$\boldsymbol{H} = \frac{\partial h(t, \hat{\boldsymbol{W}}, \varepsilon_H)}{\partial \hat{\boldsymbol{W}}}, \tag{15}$$

and

$$\dot{\boldsymbol{S}} = \boldsymbol{Q}_f - \boldsymbol{S}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_f^{-1}\boldsymbol{H}\boldsymbol{S}, \tag{16}$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times 1}$ is the observation matrix of Eq. (12) linearized around $\hat{\boldsymbol{W}}$, $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ is symmetrical positive-definite diagonal matrix with $\boldsymbol{S}(t_0) = \boldsymbol{S}(t_0)^{\mathrm{T}} \succ 0$.

## 4. CONVERGENCE AND STABILITY ANALYSIS

Given the ideal NN parameter $\boldsymbol{W}$, the estimation error $\tilde{\boldsymbol{W}}$ is defined as $\tilde{\boldsymbol{W}} = \boldsymbol{W} - \hat{\boldsymbol{W}}$. Similar to most adaptive control problems that require online tuning of parameters, persistence of excitation (PE) is needed for proper convergence of NN parameters [20].

**Assumption 4:** During online tuning, system states related signals $\boldsymbol{z}(t) = [\nabla\phi(\boldsymbol{x})^{\mathrm{T}}, \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x})]^{\mathrm{T}}$ satisfy the following PE condition:

$$\mu_1\boldsymbol{I} \le \int_{t_0}^{t_0+\delta} \boldsymbol{z}(t)\boldsymbol{z}(t)^{\mathrm{T}} d\tau \le \mu_2\boldsymbol{I}; \quad \forall t_0 \ge 0,$$

where $\mu_1 \in \mathbb{R}^+$ , $\mu_2 \in \mathbb{R}^+$, $\delta \in \mathbb{R}^+$, and $\boldsymbol{I}$ is an identity matrix of appropriate dimensions.

It is one of the contributions in this paper that EKF is employed to tune single-layer NN parameters online specifically for synthesising optimal control laws in a continuous-time nonlinear system. Parameter convergence is shown in the following theorem.

**Theorem 1**: Under Assumptions 1 to 4 and the EFK estimation scheme provided by Eqs. (13) to (16), nearly optimal control laws for the nonlinear system as in Eq. (2) are given by an actor NN

$$\hat{\boldsymbol{u}}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\left[\boldsymbol{P}\boldsymbol{x} + \nabla\phi^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{W}}\right] \tag{17}$$

in an online tuning manner, with the adaptive variable $\hat{\boldsymbol{W}}$ converging to the ideal value $\boldsymbol{W}$ within an error bound:

$$\|\tilde{\boldsymbol{W}}\| \le \frac{6b_\phi^3 b_x b_G n_{Rf} + \sqrt{36b_\phi^6 b_x b_G n_{Rf}^2 - 32 m_\sigma n_{Rf} m_{\phi G}^2}}{2n_{Rf} m_{\phi G}^2} \triangleq b_{\tilde{W}}.$$

**Proof:** Due to space limit, the proof is not presented herein.

**Remark 1:** Although convergence of $\|\tilde{\boldsymbol{W}}\|$ is shown to be uniformly ultimately bounded (UUB) under the tuning scheme provided by the EKF, the evolution path of $\hat{\boldsymbol{W}}$ may not provide stabilising control. This has been widely documented in literature as discussed in Section 1. Without adding a stabilising tuning scheme to the actor NN (Eq.(17)), the closed-loop stability of the overall system needs to be further analysed.

**Theorem 2**: Given Assumptions 1 to 4 and the EFK estimation scheme provided by Eqs. (13) to (16), the nonlinear system as in Eq. (2) remains asymptotically stable during online tuning under the nearly optimal control given by Eq. (17), if $\boldsymbol{P}$ in Eq. (7) is selected large enough such that for some constant $m_P \in \mathbb{R}^+$, $\|\boldsymbol{P}\| > m_P$, where

$$m_P \triangleq \frac{2c_{p1} + 2\sqrt{c_{p1}^2 + m_G c_{p2}}}{m_G},$$

with

$$c_{p1} = \frac{1}{2}b_G b_\phi n_W + \frac{1}{2}b_G b_\varepsilon + b_\phi b_G b_{\tilde{W}},$$

and

$$c_{p2} = \frac{1}{2}b_\phi b_G b_\varepsilon n_W + \frac{1}{4}b_\varepsilon^2 b_G + b_\varepsilon b_f + b_\phi b_f b_{\tilde{W}} + b_\phi^2 b_G n_W b_{\tilde{W}} - m_Q - \frac{1}{4}m_{DW}.$$

**Proof:** Due to space limit, the proof is not presented herein.

**Remark 2**: The proposed new form of value function approximation is a major contribution of this paper. It maintains the stability of the closed-loop system during online tuning without the necessity of adding an additional stabilising tuning scheme to the actor NN or adding a stabilising switch mechanism to the critic NN. Moreover, to the best of our knowledge, the proposed algorithm is the first successful approach to provide nonlinear systems as in Eq. (2) with asymptotic stability among available methods in literature.

## 5. SIMULATIONS

Two simulation examples are given in this section. Finding the optimal control law for a linear time-invariant (LTI) model is demonstrated first to compare the proposed algorithm with the well-established linear-quadratic regulator (LQR) for LTI cases. A nonlinear model with a known value function is then introduced to further verify the convergence and stability of the proposed algorithm. These two examples are the same as in [7] and hence the performance of the proposed method can be compared with that in [7].

### 5.1. Linear Example

The following continuous-time LTI model is consider:

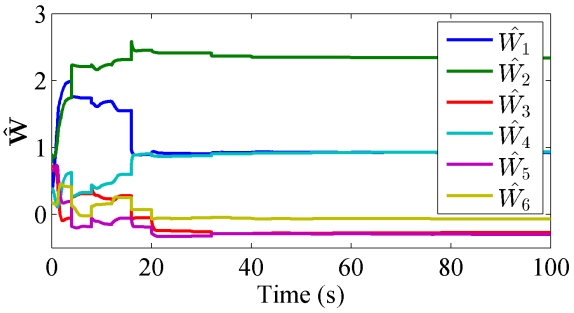


**Fig. 1.** NN parameters convergence history during online tuning

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \,.$$

Note that a specific form of $\bar{Q}(\boldsymbol{x})$ in the cost function is not given in Eq. (3). For this example, let $\bar{Q}(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x}$ with $\boldsymbol{Q} = \boldsymbol{I}_{3\times 3}$ and $R = 1$.

The activation function set is defined as

$$\phi(\boldsymbol{x}) = [x_1^2, x_1 x_2, x_2^2, x_1 x_3, x_2 x_3, x_3^2]^{\mathrm{T}} ,$$

and the critic NN weight are

$$\hat{\boldsymbol{W}} = [\hat{W}_1, \hat{W}_2, \hat{W}_3, \hat{W}_4, \hat{W}_5, \hat{W}_6]^{\mathrm{T}} .$$

In the simulation, $\boldsymbol{P} = \boldsymbol{I}_{n_x \times n_x}$, and the EKF takes $\boldsymbol{Q}_f = \boldsymbol{I}_{6\times 6}$, $R_f = 1$, and $\boldsymbol{S}(t_0) = 0$. When implementing the proposed algorithm as in Theorem 2, a small probing noise is added to perturb the system to satisfy the PE condition. The convergence of NN weights are plotted in Fig. 1. It can be seen that uniform convergence is quickly reached within 40 seconds, much faster than the 800 seconds needed by the algorithm of [7].

It is known that the linear optimal feedback control law obtained by means of the LQR algorithm is in the form of

$$\boldsymbol{u} = -\boldsymbol{K}_{LQR} \boldsymbol{x} ,$$

with $\boldsymbol{K}_{LQR} = [-0.1352, -0.1501, 0.4329]$.

Rewriting Eq. (17) as

$$\hat{\boldsymbol{u}}(\boldsymbol{x}) = -\frac{1}{2} \boldsymbol{R}^{-1} \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x}) \left[ \boldsymbol{P}\boldsymbol{x} + \nabla \phi^{\mathrm{T}}(\boldsymbol{x}) \hat{\boldsymbol{W}} \right] = -\boldsymbol{K}_{NN} \boldsymbol{x} ,$$

and using $\hat{\boldsymbol{W}} = [0.9243, 2.3371, -0.2714, 0.9343, -0.2987, -0.0677]^{\mathrm{T}}$ sampled at 100 seconds to compute the feedback gain gives $\boldsymbol{K}_{NN} = [-0.1357, -0.1493, 0.4323]$. As can be seen from $\boldsymbol{K}_{LQR} - \boldsymbol{K}_{NN} = [0.0005, -0.0008, 0.0006]$, the proposed algorithm provides adequate accuracy in finding the linear optimal control in an online tuning manner.

### 5.2. Nonlinear Example

A nonlinear system in the form of Eq. (2) is considered, with

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2 \quad 1 - \cos(2x_1) + 2^{\,2} \end{bmatrix} ,$$

and

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

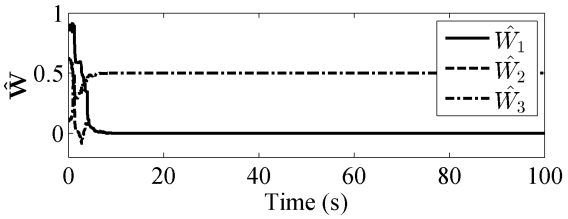


**Fig. 2.** NN parameters convergence history during online tuning

For $\boldsymbol{Q} = \boldsymbol{I}_{2\times 2}$ and $R = 1$, the corresponding optimal value function and optimal control are known to be

$$V^*(\boldsymbol{x}) = 0.5x_1^2 + x_2^2 \text{ and } u^*(x) = -\cos(2x_1) + 2\, x_2 ,$$

respectively, as given in [7].

The NN activation function set are selected as:

$$\phi(\boldsymbol{x}) = [x_1^2, x_1 x_2, x_2^2]^{\mathrm{T}} ,$$

and the NN weights are:

$$\hat{\boldsymbol{W}} = [\hat{W}_1, \hat{W}_2, \hat{W}_3]^{\mathrm{T}} .$$

In the simulation, $\boldsymbol{P} = \boldsymbol{I}_{n_x \times n_x}$, and the EKF takes $\boldsymbol{Q}_f = \boldsymbol{I}_{6\times 6}$, $R_f = 1$, and $\boldsymbol{S}(t_0) = 0$. Similar to the linear example, a small probing noise is added to meet the PE condition. The online tuning reaches convergence within 10 seconds (see Fig. 2), faster than the 80 seconds of the algorithm of [7]. At 100 seconds, $\hat{\boldsymbol{W}} = [0, 0, 0.5]^{\mathrm{T}}$, and by substituting these values back into Eqs. (10) and (17), it can be easily verified that convergence is reached with good accuracy.

**Remark 3**: It is worth emphasis that the proposed algorithm only requires to tune the critic NN without additional tuning for the actor NN. No stabilising switch mechanism is needed either. This is a major difference of the new critic-actor algorithm from its counterparts in literature. As shown by the two examples, the overall implementation is simplified with only one NN tuning loop and with faster online tuning without jeopardising the closed-loop stability. This is a significant improvement to the online synchronous policy iteration theory framework.

## 6. CONCLUSIONS

A new synchronous policy-iteration algorithm implemented online for infinite-horizon optimal control of continuous-time nonlinear systems is proposed in this paper. Major contributions lies in that a new form of value function approximation is proposed and an EKF is used for NN tuning. The new algorithm eliminates the need for additional stabilizing tuning of the actor NN or stabilizing logic-switch mechanism for the critic NN. Convergence analysis shows an important advantage of the proposed technique over other synchronous policy-iteration algorithms in that asymptotic convergence of system states is guaranteed. As a result, the new critic-actor structure with EKF tuning simplifies the online adaptive controller implementation and meanwhile provides satisfactory stabilizing control. In addition, parameter convergence of the proposed algorithm is generally faster than the method of [7] as shown in simulations.

## 7. REFERENCES


[1] R.A. Howard, *Dynamic Programming and Markov Processes*. MIT Press: Cambridge, MA, 1960.

[2] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*. MIT Press: Cambridge, MA, 1998.

[3] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, pp. 359-366, 1989.

[4] M. Abu-Khalaf and F.L. Lewis, "Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network Hjb Approach," *Automatica*, Vol. 41, No. 5, pp. 779-791, 2005.

[5] D. Vrabie and F. Lewis, "Neural Network Approach to Continuous-Time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems," *Neural Networks*, Vol. 22, No. 3, pp. 237-246, 2009.

[6] D. Liu, X. Yang, and H. Li, "Adaptive Optimal Control for a Class of Continuous-Time Affine Nonlinear Systems with Unknown Internal Dynamics," *Neural Comput. Appl.*, Vol. 23, No. 7-8, pp. 1843-1850, 2013.

[7] K.G. Vamvoudakis and F.L. Lewis, "Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *Automatica*, Vol. 46, No. 5, pp. 878-888, 2010.

[8] K.G. Vamvoudakis, D. Vrabie, and F.L. Lewis, "Online Adaptive Algorithm for Optimal Control with Integral Reinforcement Learning," *Int. J. Robust Nonlin.*, 10.1002/rnc.3018, 2013.

[9] S. Bhasin, et al., "A Novel Actor-Critic-Identifier Architecture for Approximate Optimal Control of Uncertain Nonlinear Systems," *Automatica*, Vol. 49, No. 1, pp. 82-92, 2013.

[10] H. Modares, F.L. Lewis, and M.B. Naghibi-Sistani, "Adaptive Optimal Control of Unknown Constrained-Input Systems Using Policy Iteration and Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 24, No. 10, pp. 1513-1525, 2013.

[11] D. Liu, et al., "Neural-Network-Observer-Based Optimal Control for Unknown Nonlinear Systems Using Adaptive Dynamic Programming," *Int. J. Control*, Vol. 86, No. 9, pp. 1554-1566, 2013.

[12] H.-N. Wu and B. Luo, "Neural Network Based Online Simultaneous Policy Update Algorithm for Solving the Hji Equation in Nonlinear H-Infty Control," *IEEE Trans. Neural Netw. Learn. Syst.*, Vol. 23, No. 12, pp. 1884-1895, 2012.

[13] H. Zhang, et al., "Online Adaptive Policy Learning Algorithm for H∞ State Feedback Control of Unknown Affine Nonlinear Discrete-Time Systems," *IEEE Trans. Cybern.*, Vol. 44, No. 12, pp. 2706-2718, 2014.

[14] D. Liu, et al., "Neural-Network-Based Online Hjb Solution for Optimal Robust Guaranteed Cost Control of Continuous-Time Uncertain Nonlinear Systems," *IEEE Trans. Cybern.*, Vol. 44, No. 12, pp. 2834-2847, 2014.

[15] H. Modares, M.-B. Naghibi Sistani, and F.L. Lewis, "A Policy Iteration Approach to Online Optimal Control of Continuous-Time Constrained-Input Systems," *ISA Trans.*, Vol. 52, No. 5, pp. 611-621, 2013.

[16] Q. Wei, et al., "Finite-Approximation-Error-Based Discrete-Time Iterative Adaptive Dynamic Programming," *IEEE Trans. Cybern.*, Vol. 44, No. 12, pp. 2820-2833, 2014.

[17] A. Heydari, "Revisiting Approximate Dynamic Programming and Its Convergence," *IEEE Trans. Cybern.*, Vol. 44, No. 12, pp. 2733-2743, 2014.

[18] R.W. Beard, G.N. Saridis, and J.T. Wen, "Galerkin Approximations of the Generalized Hamilton-Jacobi-Bellman Equation," *Automatica*, Vol. 33, No. 12, pp. 2159-2177, 1997.

[19] G.N. Saridis and C.-S.G. Lee, "An Approximation Theory of Optimal Control for Trainable Manipulators," *IEEE Trans. Syst., Man, Cybern.*, Vol. 9, No. 3, pp. 152-159, 1979.

[20] P.A. Ioannou and J. Sun, *Robust Adaptive Control*. PTR Prentice-Hall: Upper Saddle River, NJ, 1996.