Efficient neural learning control of nonlinear dynamics with applications

Ruixin Liu¹,²,³, Xiaogang Huang², Chunning Liu² and Fucheng Liu¹,²,³

Abstract
The control of nonlinear dynamics is gaining increasing attention since many practical systems are with such kind of characteristics. To deal with the system uncertainty, in this paper, the efficient learning control using neural network is proposed for the nonlinear strict-feedback system. The whole scheme is with the back-stepping design, while the novel learning is proposed for the neural network weights update. To deal with the approximation error, the robust item is added. The stability of the closed-loop dynamics is analysed and the effectiveness of the design is verified through flight simulation.

Keywords
Neural network, nonlinear dynamics, learning control, strict-feedback system

Introduction
Nonlinear dynamics exists in many practical systems such as robots,¹ manipulators,² flight vehicle,³ quadrotors,⁴ and MEMS (microelectromechanical system) gyroscope.⁵ Control of nonlinear dynamics⁶–⁸ is challenging since the design should be according to the structure while the nonlinearity is difficult to deal with. For the nonlinear design, the method is based on Lyapunov theory to make the energy decreasing. One important concern is on how to deal with the unknown nonlinear function. Some designs are using the knowledge of upper bound, while some works are based on the linearized parametric model. The other concern is on the form of the dynamics. For example, the controllability canonical form is different from the strict-feedback system and the pure-feedback system. For the dynamics in controllability canonical form, the main design can be with the error surface and then the robust design can be used. The control of the strict-feedback system is well studied using back-stepping and dynamic surface design. Though the design might be efficient, how to deal with the unknown dynamics is not easy since there might be not enough information to construct the linearized parametric model.

The intelligent control⁹–¹¹ can provide learning-based structure and the design is more convenient. Many works can be found as approximate control, reinforcement learning control, adaptive dynamic programming control,⁹ fault tolerant control,¹² and so on. Though the motivation might be quite different, the idea is clear that the neural network (NN) can be used as bridge between known and unknown. Some works can be further included with disturbance observer,¹³,¹⁴ sliding mode design, and H∞ design.¹⁵

For the system with unknown nonlinear functions, the NN can be used for approximation. Typically, two kinds of designs are widely designed. One way is to approximate the ideal control input, while the other is towards the nonlinear function approximation. The designs can be found in the literature,¹⁶–¹⁸ while the application can be found in robot system, ship system, flight system, and mechanical systems. If the nonlinear function can be approximated as precise as possible, the tracking performance can be better.¹⁹ However, most neural control is on the closed-loop stability using the tracking error to tune the NN weights. In this way, the system can be stable but after checking carefully, the approximation is far from the true value of the nonlinear function. Recently, some works have been towards the learning improvement using composite learning.²⁰,²¹ In the design, the theoretical analysis

¹Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China
²Shanghai Xin Yue Lian Hui Electronic Technology Co., Ltd., Shanghai, China
³Shanghai Aerospace Control Technology Institute, Shanghai, China

Corresponding author:
Ruixin Liu, Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China.
Email: ruixinliu81@163.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
is rigorously presented, while the prediction error is constructed using the dynamics and the approximation. In practice, more practical designs are expected if more system information can be obtained such as the derivative of the system state.

Based on the above-mentioned discussion, there exist many works on intelligent control. But most designs are towards the function approximation, and then the controller is constructed to obtain the system stability. During the process, the attention is on closed-loop system stability using tracking error to tune the weights. But the adaptation of the intelligent system is not sufficiently considered. Thus, in this paper, the approximation performance is considered and the new prediction error is given using the derivative of the system state. Furthermore, the efficient learning update law is constructed and the closed-loop system stability is analysed.

The structure of the paper is given as follows. Section ‘Model dynamics and problem formulation’ presents the nonlinear strict-feedback dynamics. Sections ‘Efficient learning control’ and ‘Stability analysis’ present the learning control and the closed-loop stability analysis, respectively. The verification is presented in section ‘Simulation’. Section ‘Conclusions and future works’ gives the conclusions and the future discussions.

**Model dynamics and problem formulation**

In this paper, the following dynamics with strict-feedback form is considered:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1) + g_1 x_2 \\
\dot{x}_2 &= f_2(x_2) + g_2 x_1 + g_u u \\
\dot{y} &= x_1 
\end{align*}
\]

where \(f_i\) are the nonlinear system functions, \(g_i\) are the control gain functions, \(\dot{x}_i = [\dot{x}_1, \ldots, \dot{x}_i]^T\) are the system states, \(y\) is the output, and \(u\) is the control input.

**Assumption 1.** The system states \(x_i\) and their derivatives \(\dot{x}_i\) can be obtained.

**Assumption 2.** The functions \(f_i\) are not known, while \(g_i\) are known. The control goal is to design the efficient learning algorithm for the back-stepping control so that the output \(y\) can track the reference signal \(y_r\).

**Efficient learning control**

For the strict-feedback design, the back-stepping scheme is of great interest since the design can break the complex dynamics into several simple dynamics. The main difficulty is the so-called ‘explosion of complexity’. Several designs can be introduced for simplification such as dynamic surface control and the command filtered back-stepping. In this paper, the derivative of the virtual control signal is obtained using \(\dot{\chi}(k) = \chi'(k)\), where \(\chi'(k)\) is the virtual signal and \(t_s\) is the sample period.

**Step 1.** As the first equation shown in dynamics (1), it is known that

\[
\dot{x}_1 = f_1(x_1) + g_1 x_2 = \omega_1^T \theta_1(x_1) + e_1 + g_1 x_2
\]

where \(\omega_1\) is unknown and \(e_1\) is the approximation error satisfying \(|e_1| \leq e_1^m\) and \(e_1^m\) is the upper bound of \(e_1\).

Define the tracking error

\[
e_1 = \dot{x}_1 - \dot{x}_1^d
\]

where \(\dot{x}_1^d = \gamma_d

Design virtual control \(\dot{x}_2^d\) as

\[
\dot{x}_2^d = -\dot{\omega}_1^T \theta_1(x_1) - k_1 e_1 + \dot{x}_1 - e_1^m \text{sign}(e_1) / g_1
\]

where \(\dot{\omega}_1\) is the NN weight vector and \(k_1 > 0\) is the design constant.

Define \(e_2 = \dot{x}_2 - \dot{x}_2^d\). Then, the derivative of \(e_1\) is obtained as

\[
\dot{e}_1 = \dot{\dot{x}}_1 - \dot{\dot{x}}_1^d = \omega_1^T \dot{\theta}_1(x_1) + e_1 + g_1 \dot{x}_2 - \dot{x}_1^d = \omega_1^T \dot{\theta}_1(x_1) + e_1 - e_1^m \text{sign}(e_1) - k_1 e_1 + g_1 e_2
\]

where \(\dot{\omega}_1 = \omega_1^* \dot{\theta}_1 - \dot{\omega}_1\)

Define the prediction error as

\[
z_1 = \dot{x}_1 - \dot{\dot{x}}_1^d = \omega_1^T \dot{\theta}_1(x_1) + e_1
\]

Since \(f_1\) is unknown, \(x_1\) is unknown. Here, since the dynamics does not contain the noise, the information is calculated using \(\dot{x}_1 \approx x_1(k+1) - x_1(k) / t_s\). The signal \(\dot{x}\) is calculated as

\[
\dot{x}_1 = \omega_1^T \dot{\theta}_1(x_1) + g_1 x_2
\]

Then, the following equality can be obtained

\[
z_1 = \omega_1^T \dot{\theta}_1(x_1) + e_1
\]

The NN weight update is given as

\[
\dot{\omega}_1 = \gamma_1 (e_1 + \gamma_2 z_1) \dot{\theta}_1(x_1)
\]

where \(\gamma_1\) is a positive design constant.

**Step 2.** As the ith equation shown in dynamics (1) and using NN to approximate \(f_i(\dot{x}_i)\), it is known that

\[
\dot{x}_i = f_i(\dot{x}_i) + g_i u = \omega_i^T \theta_i(\dot{x}_i) + e_i + g_i x_i + 1
\]

where \(\omega_i^*\) is unknown and \(e_i\) is the approximation error with \(|e_i| \leq e_i^m\) and \(e_i^m\) is the upper bound of \(e_i\).
Design virtual control $\xi^d_{i+1}$ as

$$\xi^d_{i+1} = -\omega_i T \theta_i(\xi_i) - k_i e_i - g_{i-1} e_{i-1} + \dot{\xi}^d_i - \omega_i^n \text{sign}(e_i)$$

(11)

where $\omega_i$ is the NN weight vector, $k_i > 0$ is the design constant, and $\ddot{\xi}^d_i = \dot{\xi}^{(k+1)} - \dot{\xi}^{(k)}$.

The derivative of $e_i$ is obtained as

$$\dot{e}_i = \dot{\xi}_i - \dot{\xi}_i = \omega_i T \theta_i(\xi_i) + e_i - \omega_i^n \text{sign}(e_i) - k_i e_i - g_{i-1} e_{i-1}$$

(12)

where $\dot{\omega}_i = \omega_i^* - \dot{\omega}_i$.

The prediction error is constructed as

$$z_i = \dot{\xi}_i - \dot{\xi}_i$$

(13)

Since $f_1$ is unknown, $\dot{\xi}_i$ is unknown. Here, since the dynamics does not contain the noise, the information is calculated using $\dot{\xi}_i = \dot{\xi}^{(k)}$. The signal $\dot{\xi}_i$ is calculated as

$$\dot{\xi}_i = \omega_i T \theta_i(\xi_i) + g_i u$$

(14)

Then, the following equality can be obtained

$$z_i = \omega_i^* \theta_i(\xi_i) + e_i$$

(15)

The NN weight update is given as

$$\dot{\omega}_i = \gamma_i (v_i + \gamma_{i2} z_i) \theta_i(\xi_i)$$

(16)

where $\gamma_i$ and $\gamma_{i2}$ are the positive design constants.

**Step 2**. As the 0th equation shown in dynamics (1) and using NN to approximate $f_2(\xi_2)$, it is known that

$$\dot{\xi}_n = f_n(\xi_n) + g_n u = \omega_n^T \theta_n(\xi_n) + e_n + g_n u$$

(17)

where $\omega_n^*$ is unknown and $e_n$ is the approximation error with $|e_n| \leq \epsilon_n^m$ and $\epsilon_n^m$ is the upper bound of $e_n$.

The final control signal $u$ is designed as

$$u = -\dot{\omega}_n^T \theta_n(\xi_n) - k_n e_n - g_{n-1} e_{n-1} + \dot{\xi}_n^d - \omega_n^n \text{sign}(e_n)$$

(18)

where $\dot{\omega}_n$ is the NN weight vector, $k_n > 0$ is the design constant, and $\ddot{\xi}^d_n = \dot{\xi}^{(n+1)} - \dot{\xi}^{(n)}$.

The derivative of $e_n$ is obtained as

$$\dot{e}_n = \dot{\xi}_n - \dot{\xi}_n = \omega_n^T \theta_n(\xi_n) + e_n - k_n e_n - g_{n-1} e_{n-1}$$

(19)

where $\dot{\omega}_n = \omega_n^* - \dot{\omega}_n$.

The prediction error is constructed as

$$z_n = \dot{\xi}_n - \dot{\xi}_n$$

(20)

Since $f_n$ is unknown, $\dot{\xi}_n$ is unknown. Here, since the dynamics does not contain the noise, the information is calculated using $\dot{\xi}_n = \dot{\xi}^{(n)}$. The signal $\dot{\xi}_n$ is calculated as

$$\dot{\xi}_n = \omega_n T \theta_n(\xi_n) + g_n u$$

(21)

Then, the following equality can be obtained

$$z_n = \omega_n^T \theta_n(\xi_n) + e_n$$

(22)

The NN weight update is given as

$$\dot{\omega}_n = \gamma_n (v_n + \gamma_{n2} z_n) \theta_n(\xi_n)$$

(23)

where $\gamma_n$ and $\gamma_{n2}$ are the positive design constants.

**Stability analysis**

**Theorem 1**. Consider the dynamics (1) with the virtual signals (4), (11), (18) and the neural weights update (9), (16), (23). Then, the tracking errors are bounded.

**Proof.** The Lyapunov function is selected as

$$L_i = \frac{1}{2} \sum_{i=1}^{n} \epsilon_i^2 + \omega_i^T \gamma_i^{-1} \omega_i$$

(24)

The derivative of $L_i$ is derived as

$$\dot{L}_i = e_i^T (\omega_i^T \theta_i(\xi_i) + e_i - \epsilon_i^n \text{sign}(e_i) - k_i e_i + g_i e_2)
- \omega_i^T (e_i + \gamma_{i2} z_i) \theta_i(\xi_i)
+ \sum_{i=2}^{n-1} e_i(-k_i e_i - g_{i-1} e_{i-1} + g_i e_i + 1)
+ e_n(\omega_n^T \theta_n(\xi_n) + e_n - k_n e_n - g_{n-1} e_{n-1})
- \omega_n^T (e_n + \gamma_{n2} z_n) \theta_n(\xi_n)$$

(25)

Then it is calculated as

$$\dot{L}_i = \sum_{i=1}^{n} -k_i \epsilon_i^2 + e_i(e_i - \epsilon_i^n \text{sign}(e_i)) - (z_i - e_i) \gamma_{i2} z_i$$

(26)

The equation is further obtained as

$$\dot{L}_i \leq \sum_{i=1}^{n} -k_i \epsilon_i^2 - k_i \gamma_i^2 + P$$

(27)

where $k_i = \gamma_i (1 - \frac{k_i}{\gamma_i})$, $P = \mu \gamma_i^2 (\epsilon_i^n)^2$, and $\mu$ is the scalar. So all the signals in equation (24) are bounded. Now it is concluded that under the proposed method, the system can track the reference very well. This completes the proof.

**Simulation**

The flight dynamics are presented with attack angle $\alpha$, flight path angle (FPA) $\gamma$, and pitch rate $q$. The
control input is the elevator deflection $\delta_e$. The attitude dynamics is listed as follows

$$\dot{\gamma} = \frac{L + T \sin \alpha}{mV} - \frac{g \cos \gamma}{V} \quad (28)$$

$$\dot{\alpha} = q - \gamma \quad (29)$$

$$\dot{q} = \frac{M_{yy}}{T_{yy}} \quad (30)$$

Define $X = [\xi_1, \xi_2, \xi_3]^T$, $\xi_1 = \gamma$, $\xi_2 = \theta_p$, and $\xi_3 = q$, where $\theta_p = \alpha + \gamma$. The general form of the dynamics can be obtained as

$$\dot{\xi}_1 = f_1(\xi_1) + g_1 \xi_2$$
$$\dot{\xi}_2 = \dot{\xi}_3$$
$$\dot{\xi}_3 = f_3(X) + g_3 u$$
$$u = \delta_e$$

The way of using the tracking error to update the neural weight is denoted as ‘Method 1’, while the design in this paper is named as ‘Proposed Method’ which means the predictor-based update design. To show the performance, the index is selected as $J_i = \int |e_i| dt$, where $i = 1, 2, 3$. The parameters are selected as in Zhang et al.\textsuperscript{23} while $\gamma_i, i = 1, 3$, are selected as 1.5. Define $f_i = \omega_i \theta_i, i = 1, 3$.

In the simulation, the altitude will climb from 86,000 to 87,000 ft in 50 s, while the altitude will decrease from 87,000 to 85,000 ft in the next 50 s. Given the reference signal of altitude, the flight path angle is generated through the similar way as in Zhang et al.\textsuperscript{23} In Example 1, there is no noise, while in Example 2 there exist noise for $a$ and $q$.

**Example 1.** The simulation results are presented in Figures 1–6. It is clear that the proposed method obtains better tracking performance with high tracking

![Figure 1. Altitude tracking.](image1)

![Figure 2. System states.](image2)
The control input responds smoother as shown in Figure 3. The neural approximation is depicted in Figure 5, while the trajectory of NN weights is shown in Figure 6. The tracking performance is demonstrated in Figure 4. Overall, the proposed method achieves the better convergence and the higher tracking accuracy.

**Example 2.** The random noises with amplitude 0.0001 and 0.001 are added for $a$ and $q$. In Figure 7, the system response is demonstrated, while in Figure 8 the elevator deflection is depicted. Furthermore, the NN response is shown in Figure 9. It is interesting to see that the proposed method can achieve much better performance in case of measurement noise. Also from the response of
the elevator deflection and the NN weights, more chattering occurs in case of noise.

Conclusions and future works

The efficient learning-based control is designed for the strict-feedback systems. The design constructs the signal to obtain the prediction error for the neural weight update. The system stability is analysed and the control performance is verified through nonlinear dynamic simulation.

For the future work, the output-feedback design can be studied. In reality, the time-varying disturbance exists in the dynamics and the new estimation design can be analyzed. For practical applications, the method can be applied to manipulators, underwater vehicles, quadrotor, and automobile dynamics for experimental purpose.
Figure 7. System states with random noises.

Figure 8. Elevator deflection.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs
Ruixin Liu https://orcid.org/0000-0001-5126-1532

References
1. He W and Dong Y. Adaptive fuzzy neural network control for a constrained robot using impedance learning. IEEE T Neural Netw Learn Syst 2018; 29(4): 1174–1186.
2. Yu S, Yu X and Stonier R. Continuous finite-time control for robotic manipulators with terminal sliding modes. In: Sixth international conference of information fusion, Cairns, QLD, Australia, 8–11 July 2003, pp. 1433–1440. New York: IEEE.
3. Yang J, Li S, Sun C, et al. Nonlinear-disturbance-observer-based robust flight control for airbreathing hypersonic vehicles. IEEE T Aerosp Electr Syst 2013; 49(2): 1263–1275.
4. Xu B. Composite learning finite-time control with application to quadrotors. IEEE T Syst Man Cyb 2018; 48(10): 1806–1815.
5. Fei JT and Yan WF. Adaptive control of MEMS gyroscope using global fast terminal sliding mode control and fuzzy-neural-network. Nonlinear Dyna 2014; 78(1): 103–116.
6. Kokotovic P. The joy of feedback: nonlinear and adaptive; 1991 bode prize lecture. IEEE Control Syst Mag 1991; 12: 7–17.
7. Polycarpou M. Stable adaptive neural control scheme for nonlinear systems. IEEE T Automat Control 1996; 41(3): 447–451.
8. Wang D and Huang J. Adaptive neural network control for a class of uncertain nonlinear systems in pure-feedback form. Automatica 2002; 38(8): 1365–1372.
9. Wang D, He H, Zhong X, et al. Event-driven nonlinear discounted optimal regulation involving a power system application. IEEE T Ind Electron 2017; 64(10): 8177–8186.
10. Chen M and Ge SS. Adaptive neural output feedback control of uncertain nonlinear systems with unknown hysteresis using disturbance observer. IEEE T Ind Electron 2015; 62(12): 7706–7716.
11. He W, Sun Y, Yan Z, et al. Disturbance observer-based neural network control of cooperative multiple manipulators with input saturation. IEEE T Neural Netw Learn Syst. Epub ahead of print 13 August 2019. DOI: 10.1109/TNNLS.2019.2923241.
12. Yin S, Jiang Y, Tian Y, et al. A data-driven fuzzy information granulation approach for freight volume forecasting. IEEE T Ind Electron 2017; 64(2): 1447–1456.
13. Chen WH, Yang J, Guo L, et al. Disturbance-observer-based control and related methods – an overview. IEEE T Ind Electron 2016; 63(2): 1083–1095.
14. Xu B, Shou Y, Luo J, et al. Neural learning control of strict-feedback systems using disturbance observer. IEEE T Neural Netw Learn Syst 2019; 30(5): 1296–1307.
15. Qian W, Xing W, Wang L, et al. New optimal analysis method to stability and $H_{\infty}$ performance of varying delayed systems. ISA Trans 2019; 93: 137–144.
16. Wang L. Design and analysis of fuzzy identifiers of nonlinear dynamic systems. *IEEE T Automat Control* 1995; 40(1): 11–23.
17. Xu B. Composite learning control of flexible-link manipulator using NN and DOB. *IEEE T Syst Man Cyb* 2018; 48(11): 1979–1985.
18. Peng Z, Wang J and Wang D. Distributed containment maneuvering of multiple marine vessels via neurodynamics-based output feedback. *IEEE T Ind Electron* 2017; 64(5): 3831–3839.
19. Hojati M and Gazor S. Hybrid adaptive fuzzy identification and control of nonlinear systems. *IEEE T Fuzzy Syst* 2002; 10(2): 198–210.
20. Pan Y, Zhou Y, Sun T, et al. Composite adaptive fuzzy $H_\infty$ tracking control of uncertain nonlinear systems. *Neurocomputing* 2013; 99: 15–24.
21. Bellomo D, Naso D, Turchiano B, et al. Composite adaptive fuzzy control. In: *Proceedings of the 16th IFAC world congress*, Prague, 4–8 July 2005.
22. Parker JT, Serrani A, Yurkovich S, et al. Control-oriented modeling of an air-breathing hypersonic vehicle. *J Guid Control Dyn* 2007; 30(3): 856–869.
23. Zhang S, Li C and Zhu J. Composite dynamic surface control of hypersonic flight dynamics using neural networks. *Sci China Inform Sci* 2015; 58(7): 1–9.