# Technical Report of
# Adaptive Neural Control For Non-Strict-Feedback Nonlinear Systems With Input Delay


Vishnu Mandala
UID: 119452608
University of Maryland, College Park
vishnum@umd.edu

Rohit Sai Suresh
UID: 119283684
University of Maryland, College Park
rohitsai@umd.edu



## Abstract

*In this technical report, we undertake a comprehensive implementation and analysis of the controller design proposed in the paper "Adaptive Neural Control for Non-Strict-Feedback Nonlinear Systems with Input Delay" by Huanqing Wang, Siwen Liu, and Xuebo Yang. The core focus of the paper, and consequently of our report, is the development and validation of a controller for non-strict-feedback systems with input delay, a notably complex issue in control and robotics. Our implementation of the paper's methodology includes an extensive simulation conducted in MATLAB. This simulation critically tests the adaptive neural control method extended to non-strict-feedback nonlinear systems with input delay. The controller designed in the original study, and replicated in our MATLAB simulation, ensures semi-global uniform ultimate boundedness of all signals within the closed-loop system. The insights drawn from our MATLAB implementation offer a deeper understanding of the paper's innovative approach and its potential implications in the realm of adaptive neural control for nonlinear systems.*


## 1. Introduction

### 1.1. Context and Background

In the evolving landscape of control and robotics, the adaptation and advancement of control mechanisms have been pivotal. Particularly, the development and application of adaptive control strategies in managing uncertain nonlinear systems have marked a significant evolution in this domain. The incorporation of neural network methodologies, notably Gaussian radial basis functions, alongside fuzzy logic systems, has revolutionized the approach to dealing with complex control scenarios. Among these, adaptive neural control stands out due to its exceptional ability to approximate nonlinearities that are otherwise unavailable for direct analysis. This aspect of control theory has not only expanded the toolkit of engineers and researchers but has also opened new avenues in the practical application of advanced control systems.

### 1.2. The Intricacies of Non-Strict-Feedback Systems

A formidable challenge in the realm of control system design is the effective handling of non-strict-feedback nonlinear systems. These systems differ significantly from their strict-feedback counterparts in terms of their structural complexity and behavioral dynamics. A distinctive feature of non-strict-feedback systems is that their system functions are intricately linked to all the system states, thus amplifying their complexity. The addition of input delays to these systems further complicates their control, making standard control strategies less effective or inapplicable. The prevalence of such systems in real-world applications, ranging from the dynamics of mechanical systems like the ball and beam system to electronic systems like hyper-chaotic LC oscillation circuits, underscores the importance of developing robust control methodologies for them.

### 1.3. Significance of the Investigated Paper

The paper titled "Adaptive Neural Control for Non-Strict-Feedback Nonlinear Systems with Input Delay" by Huanqing Wang, Siwen Liu, and Xuebo Yang, presents an innovative approach to surmount the challenges presented by these complex systems. This paper is significant in that it extends the domain of adaptive neural control to systems characterized by non-strict-feedback and input delays. This extension is not merely an incremental improvement but a substantial leap forward, addressing a critical gap in the existing adaptive control strategies and offering a viable solution for a broader range of complex systems.

### 1.4. Primary Contributions and Research Objectives

The seminal contribution of this paper, which is the centerpiece of our technical report, is the conceptualization and development of a state-feedback stabilization controller. This controller, ingeniously crafted using a combination of backstepping techniques and adaptive neural control, is designed to ensure the semi-global uniform ultimate boundedness of the variables within the closed-loop system, even in the challenging scenario of input delays. Our report endeavors to meticulously implement this controller design in MATLAB, a computational environment renowned for its capabilities in handling intricate control system simulations. Through this implementation, we aim to not only validate the results presented in the paper but also provide a detailed analytical discourse on the methodology and the results obtained, thereby contributing to the understanding and potential application of adaptive neural control in non-strict-feedback nonlinear systems.

### 1.5. Structure of the Report

The ensuing sections of this report delve into the technical nuances of the paper's proposed methodology. We replicate the control system as described in the paper, utilizing MATLAB for our simulations. The report aims to juxtapose our findings with those presented in the original study, providing a critical analysis of both the methodologies employed and the results derived. Through this comprehensive examination, the report seeks to elucidate the implications of this innovative approach in the wider context of adaptive neural control and its potential impact on the future of nonlinear system control in various practical applications.

## 2. Technical Background

### 2.1. Non-Strict-Feedback Nonlinear Systems

Non-strict-feedback nonlinear systems encompass a broad spectrum of control systems where the relationship between system states and output is not strictly cascaded or linear. Unlike strict-feedback systems, where each state equation depends only on the current and previous states, non-strict-feedback systems have state equations that may depend on all or a subset of the other states. This attribute makes their analysis and control more challenging. The general form of such a system can be represented as:

$$\begin{aligned} \dot{x}_i &= x_{i+1} + f_i(x), \quad i = 1, 2, \dots, n-1 \\ \dot{x}_n &= u(t-\tau) + f_n(x) \\ y &= x_1 \end{aligned} \tag{1}$$

Here, $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}$ is the output, $u(t-\tau)$ denotes the control input with a delay $\tau$, and $f_i(x)$ are nonlinear functions representing system dynamics.

### 2.2. Radial Basis Function Neural Networks

The Radial Basis Function (RBF) Neural Networks are utilized for approximating the nonlinear functions $f_i(x)$ in the system. RBF Neural Networks are employed for approximating nonlinear functions due to their universal approximation capabilities. The structure of an RBF network for a function $f : \mathbb{R}^n \to \mathbb{R}$ is given by:

$$\hat{f}(x, \theta) = \sum_{j=1}^{N} w_j \phi_j(\|x - c_j\|) + b \tag{2}$$

Here, $\phi_j$ is the RBF, typically Gaussian:

$$\phi_j(r) = \exp(-\beta_j r^2), \quad r = \|x - c_j\| \tag{3}$$

with $\beta_j$ determining the width of the Gaussian. The network aims to approximate any continuous function on a compact set, with $w_j$ as weights, $c_j$ as the centers, $b$ as the bias, and $N$ as the number of neurons.

### 2.3. Adaptive Neural Control

The adaptive neural control framework aims to address uncertainties and nonlinearities in non-strict-feedback systems. The control law is derived based on Lyapunov stability principles, which ensures system stability and convergence of tracking errors. A typical control law in this context might look like:

$$u(t) = -Kx(t) + \hat{f}(x, \theta) \tag{4}$$

where $K$ is a control gain matrix determined to ensure system stability. The term $\hat{f}(x, \theta)$ is the neural network approximation of the system's unknown dynamics. The stability analysis typically involves showing that a suitable Lyapunov function decreases over time, ensuring that the system's trajectories converge to a desired behavior.

### 2.4. Challenges and Solutions for Input Delay

Input delay, characterized by a time lag between the application of a control input and its effect on the system, poses significant challenges, including potential instability and performance degradation. To counter this, the paper introduces an auxiliary system approach. This method transforms the original delayed system into an equivalent delay-free system, enabling the application of standard control techniques.

### 2.5. Auxiliary System Approach

The auxiliary system method is a strategic approach to compensate for the input delay. It is formulated as:

$$\dot{x}_{aux} = A_{aux}x_{aux} + B_{aux}u(t) \tag{5}$$

In this formulation, $x_{aux}$ represents the state of the auxiliary system. The matrices $A_{aux}$ and $B_{aux}$ are designed so that the dynamics of the auxiliary system mirror the delayed dynamics of the original system. By stabilizing the auxiliary system, the control strategy indirectly stabilizes the original system, effectively mitigating the impact of the input delay.

# 3. Technical Results and Proofs

## 3.1. Plant Dynamics

### 3.1.1 Gaussian Radial Basis Function

The Gaussian RBF, used as an activation function in neural networks, especially for function approximation in control systems, is expressed as:

$$s_i(Z) = e^{-\frac{\|Z-v_i\|^2}{2\sigma^2}} \tag{6}$$

where:

- $s_i(Z)$ is the output of the $i$-th RBF neuron for input $Z$.
- $Z$ is the input vector.
- $v_i$ is the center of the $i$-th RBF neuron.
- $\sigma$ is the standard deviation, controlling the width of the Gaussian bell curve.

The squared Euclidean distance between $Z$ and $v_i$ is given by:

$$\|Z - v_i\|^2 = (Z - v_i)^T(Z - v_i) \tag{7}$$

Applying the Gaussian exponential function, we obtain:

$$e^{-\frac{\|Z-v_i\|^2}{2\sigma^2}} \tag{8}$$

In the context of the paper, this formula is adapted by replacing $2\sigma^2$ with $\eta^2$, leading to:

$$s_i(Z) = e^{-\frac{(Z-v_i)^T(Z-v_i)}{\eta^2}} \tag{9}$$

This modification captures the localized and nonlinear response characteristics of RBF neurons, fundamental in neural networks for control systems.

### 3.1.2 Neural Network Approximation

According to the universal approximation theorem, a neural network can approximate any continuous function within a certain accuracy. This is represented as:

$$f(Z) = W^{*T}(Z) + \delta(Z) \quad \forall Z \in Z \tag{10}$$

where:

- $f(Z)$ is the continuous function being approximated.
- $Z$ is the input vector.
- $W^*$ is the ideal weight vector.
- $\delta(Z)$ represents the approximation error.

### 3.1.3 Ideal Weight Vector in Neural Network Approximation

In the context of neural network-based function approximation, $W^*$ represents the ideal weight vector. It is the theoretical set of weights that would allow the neural network to approximate a given function $f(Z)$ with the least possible error. This concept is crucial in understanding the network's approximation capabilities.

The ideal weight vector $W^*$ plays a pivotal role in function approximation:

- It is the optimal set of weights that the neural network strives to learn during the training process.
- The closer the actual weights of the neural network are to $W^*$, the more accurately the network can approximate the target function.

$W^*$ is used to denote the best possible approximation scenario within the bounds of the neural network's architecture:

$$|\delta(Z)| < \varepsilon \tag{11}$$

Here, $\delta(Z)$ represents the approximation error when the neural network, with weights approximating $W^*$, tries to model the function $f(Z)$. The error is bounded by a small positive constant $\varepsilon$, indicating the network's effectiveness in closely approximating the target function.
The ideal weight vector is therefore defined as:

$$W^* = \arg\min_{W\in\mathbb{R}^k} \sup_{Z\in\mathcal{Z}} |f(Z) - W^T\phi(Z)| \tag{12}$$

**Lemma 1.** *Given the RBF vector $(Z_n)$ for an input vector $Z_n$ in a neural network, and indices $m$ and $n$ with $m \leq n$, the following inequality holds:*

$$\|(Z_n)\|^2 \leq \|(Z_m)\|^2 \tag{13}$$

*where $(Z_n) = [s_1(Z_n), \ldots, s_l(Z_n)]^T$.*

#### 3.1.4 Auxiliary System Design to Counteract Input Delay

To address the input delay in the control system, an auxiliary system with the following dynamics is introduced:

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 - p_1\hat{x}_1\\ \dot{\hat{x}}_i &= \hat{x}_{i+1} - p_i\hat{x}_i, \qquad\qquad i = 2,\ldots,n-1\\ \dot{\hat{x}}_n &= -p_n\hat{x}_n + u(t-\tau) - u(t)\end{aligned} \tag{14}$$

where $\hat{x}_i$ is the $i$-th state of the auxiliary system, $p_i$ are the design parameters that ensure compensation for the input delay, and $u(t-\tau)-u(t)$ represents the delayed control action.

The design of the auxiliary system is such that it mimics the dynamics of the original system but without the delay in the control input. The parameters $p_i$ are selected to ensure the stability of the auxiliary system and, consequently, the stability of the controlled system. Each state $x_i$ of the auxiliary system is fed back and subtracted by a term proportional to itself, creating a damping effect. The last equation for $x_n$ incorporates the delayed input and the current input to mitigate the effect of the delay.

### 3.2. Controller Design

In this section, we design a neural network-based controller using backstepping. The process begins with the following coordinated transformation:

$$\begin{aligned}z_1 &= x_1 - \lambda_1\\ z_i &= x_i - \alpha_{i-1} - \lambda_i, \quad i = 2,\ldots,n.\end{aligned} \tag{15}$$

This transformation simplifies the system dynamics into error dynamics in new coordinates $z_i$, where the objective is to drive these error states to zero. Now, we commence the controller design process.

#### 3.2.1 Step 1

We start by computing the time derivative of $z_1$ using (15):

$$\begin{aligned}\dot{z}_1 &= \dot{x}_1 - \dot{\lambda}_1\\ &= x_2 + f_1 - \lambda_2 + p_1\lambda_1\\ &= z_2 + \alpha_1 + \lambda_1 + f_1 - \lambda_2 + p_1\lambda_1\\ &= z_2 + \alpha_1 + f_1 + p_1\lambda_1\end{aligned} \tag{16}$$

The time derivative of $z_1$ includes the system dynamics and the effect of the auxiliary system through $\lambda_1$ and $\lambda_2$.

Next, we select a Lyapunov function $V_1$ as:

$$V_1 = \frac{1}{2}z_1^2 + \frac{1}{2r_1}\theta_1^2 \tag{17}$$

Here $\tilde{\theta}_1$ is the estimation error with $\hat{\theta}_1$ being the estimate of the unknown parameter $\theta_1$, and $r_1 > 0$ is a design constant. The choice of Lyapunov function is standard in backstepping control design, reflecting the energy of the system including the estimation error. The specific choice of $V_1$ is motivated by the following considerations:

1. **Positive Definiteness:** The function is clearly positive definite as both terms $\frac{1}{2}z_1^2$ and $\frac{1}{2r_1}\tilde{\theta}_1^2$ are squared terms. Positive definiteness is essential for a Lyapunov function as it ensures that $V_1$ is zero only at the equilibrium point (where $z_1 = 0$ and $\tilde{\theta}_1 = 0$) and positive elsewhere, which is analogous to the system having zero energy at equilibrium and positive energy when away from equilibrium.

2. **Radial Unboundedness:** This choice is radially unbounded, meaning that as the state $z_1$ or the parameter error $\tilde{\theta}_1$ grows without bound, so does $V_1$. This property helps in proving global stability properties.

3. **Incorporation of Parameter Estimation Error:** The function includes the term $\frac{1}{2r_1}\tilde{\theta}_1^2$ to account for the parameter estimation error. This is significant because in adaptive control, we are not only interested in the state convergence but also in the convergence of the parameter estimates to their true values.

4. **Simplicity in Analysis:** The quadratic form of $V_1$ simplifies the analysis since its derivative leads to linear terms involving the system states and the parameter estimates. This simplifies the computation of the derivative and the subsequent stability analysis.

5. **Design Flexibility:** The parameter $r_1$ allows for design flexibility. By adjusting $r_1$, one can weigh the relative importance of the state error versus the parameter estimation error in the Lyapunov function, thus tailoring the control design to specific performance criteria.

We then compute the derivative of $V_1$:

$$\begin{aligned}\dot{V}_1 &= z_1\dot{z}_1 - \frac{1}{r_1}\tilde{\theta}_1\dot{\hat{\theta}}_1 = z_1(z_2 + \alpha_1 + f_1 + p_1\dot{\lambda}_1) - \frac{1}{r_1}\tilde{\theta}_1\dot{\hat{\theta}}_1\\ &= z_1(z_2 + \alpha_1 + \bar{f}_1(Z_1) + p_1\lambda_1) - \frac{1}{2}{z_1}^2 - \frac{1}{r_1}\tilde{\theta}_1\dot{\hat{\theta}}_1\end{aligned} \tag{18}$$

where $\bar{f}_1(Z_1)$ is the neural network approximation of the unknown smooth function $f_1$ and $Z_1 = [z_1,\ldots,z_n]^T$. The neural network approximation is obtained by applying the following transformation:

$$\bar{f}_1(Z_1) = f_1 + \frac{1}{2}Z_1 \tag{19}$$

This derivative will be used to analyze the stability of the system and guide the design of the control law.

$$\bar{f}_1(Z_1) = W_1^* \Phi_1(Z_1) + \delta_1(Z_1), |\delta_1(Z_1)| \leq \varepsilon_1 \tag{20}$$

where $\hat{f}_1(Z_1)$ contains the neural network approximation of the unknown smooth function $f_1$. This approximation introduces an error $\delta_1(Z_1)$, which is bounded by a small constant $\epsilon_1$.

Given the neural network function approximation:

$$\hat{f}_1(Z_1) = W_1^T \Phi(Z_1) + \delta_1(Z_1) \tag{21}$$

and Lemma 1 which implies:

$$\|\Phi(Z_n)\|^2 \leq \|\Phi(Z_m)\|^2 \tag{22}$$

We apply Young's inequality $ab \leq \frac{1}{2\epsilon}a^2 + \frac{\epsilon}{2}b^2$ to the term $z_1 W_1^T \Phi(Z_1)$ with a balancing parameter $a_1$:

$$z_1 W_1^T \Phi(Z_1) \leq \frac{1}{2a_1^2} z_1^2 + \frac{a_1^2}{2} \|W_1^*\|^2 \|\Phi(Z_1)\|^2 \tag{23}$$

For the approximation error term $\delta_1(Z_1)$ bounded by $\varepsilon_1$, we get:

$$z_1 \delta_1(Z_1) \leq \frac{1}{2} z_1^2 + \frac{1}{2} \varepsilon_1^2 \tag{24}$$

Combining these results, we obtain:

$$\begin{aligned} z_1 \hat{f}_1(Z_1) &\leq |z_1|(\|W_1^*\| \|\Phi(Z_1)\| + \epsilon_1) \\ &\leq |z_1|(\|W_1^*\| \|\Phi(X_1)\| + \epsilon_1) \\ z_1 \hat{f}_1(Z_1) &\leq \frac{1}{2a_1^2} z_1^2 + \frac{a_1^2}{2} \theta_1 \|\Phi(X_1)\|^2 + \frac{1}{2} z_1^2 + \frac{1}{2} \varepsilon_1^2 \\ &\leq \frac{1}{2a_1^2} z_1^2 \theta_1 \|\Phi(X_1)\|^2 + \frac{a_1^2}{2} + \frac{z_1^2}{2} + \frac{\varepsilon_1^2}{2} \\ &\leq \frac{1}{2a_1^2} z_1^2 \theta_1 \Phi^T(X_1) \Phi(X_1) + \frac{a_1^2}{2} + \frac{z_1^2}{2} + \frac{\varepsilon_1^2}{2} \end{aligned} \tag{25}$$

where $X_1 = [x_1, \lambda_1]^T$ and $\alpha_1 > 0$ is a constant.

This is how the equation is derived using Young's inequality, the given norm of the ideal weight vector $\theta_1$, and the bound on the approximation error $\varepsilon_1$. The terms $\frac{\alpha_1^2}{2}$ and $\frac{\varepsilon_1^2}{2}$ are squared terms resulting from Young's inequality application, and $\frac{z_1^2}{2}$ is directly from the approximation error bound. The term $\frac{1}{2\alpha_1^2} z_1^2 \theta_1$ results from the product of $z_1$ and $W_1^T \Phi(Z_1)$ applying Young's inequality with $a_1^2$ as the balancing parameter.

This bound is used in the stability analysis of the control system to show that the effect of the approximation error on the system's performance can be made arbitrarily small by choosing appropriate neural network weights and activation functions. Substituting the neural network approximation into $\dot{V}_1$, we get:

$$\begin{aligned} \dot{V}_1 &\leq z_1(z_2 + \alpha_1 + p_1\lambda_1 + \frac{1}{2\alpha_1^2} \tilde{\theta}_1^T \Phi_1(X_1) \dot{\Phi}_1(X_1)) \\ &\quad - \frac{1}{r_1} \dot{\hat{\theta}}_1 \tilde{\theta}_1 + \frac{\alpha_1^2}{2} + \frac{\varepsilon_1^2}{2} \\ &= z_1(z_2 + \alpha_1 + p_1\lambda_1 + \frac{1}{2a_1^2} \hat{\theta}_1 \Phi_1^T(X_1) \Phi_1(X_1)) \\ &\quad + \frac{1}{r_1} \tilde{\theta}_1 \left( \frac{r_1}{2a_1^2} z_1^2 \Phi_1^T(X_1) \Phi_1(X_1) - \dot{\hat{\theta}}_1 \right) + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \end{aligned} \tag{26}$$

We derive the intermediate control signal and the adaptive law using backstepping techniques. The backstepping approach systematically addresses the tracking error and adapts to the uncertainties in the system dynamics.

The intermediate control signal $\alpha_1$ is formulated as:

$$\alpha_1 = -k_1 z_1 - p_1 \lambda_1 - \frac{1}{2a_1^2} z_1 \hat{\theta}_1 \Phi_1^T(X_1) \Phi_1(X_1) \tag{27}$$

where $k_1$ is a positive design parameter, $p_1$ is a parameter from the auxiliary system design, and $a_1$ is a constant from Young's inequality used in the neural network approximation error bound. The term $\theta_1$ represents the square of the norm of the ideal neural network weight matrix, and $\Phi(X_1)$ is the feature vector from the radial basis function neural network.

The role of $\alpha_1$ in the control law is twofold:

- It introduces a negative feedback term proportional to the error state $z_1$, which contributes to the negative semi-definiteness of the Lyapunov function derivative.
- It cancels out the nonlinearity $z_2 + q_1 + f_1$ present in the error dynamics, effectively linearizing the system around the current state.

The adaptive law for the parameter estimate $\hat{\theta}_1$ is given by:

$$\dot{\hat{\theta}}_1 = -\gamma_1 \Phi^T(X_1) \Phi(X_1) - \hat{\theta}_1, \tag{28}$$

where $\gamma_1$ is a positive design parameter that controls the adaptation rate.

The adaptive law serves to update the parameter estimate $\hat{\theta}_1$ in real-time, with the following objectives:

- To reduce the parameter estimation error $\theta_1 - \hat{\theta}_1$, ensuring that the estimate converges to the true parameter value.
- To contribute a term to the Lyapunov function derivative that ensures its overall negative semi-definiteness, a condition necessary for system stability.

The final equation for adaptive law can be written as:

$$\dot{\hat{\theta}}_1 = \frac{r_1}{2a_1^2} z_1^2 \Phi_1^T(X_1)\Phi_1(X_1) - \delta_1\hat{\theta}_1, \hat{\theta}_1 \geq 0 \tag{29}$$

Here $k_1 > 0$ and $\delta_1 > 0$ are design parameters. These laws aim to drive the error $z_1$ to zero and adjust the parameter estimate $\hat{\theta}_1$ to improve the approximation. By combining these design elements, we construct a control strategy that not only stabilizes the system but also adapts to uncertainties, compensating for the lack of precise knowledge about the system dynamics.

Finally, by substituting $\alpha_1$ and $\dot{\hat{\theta}}_1$ into $\dot{V}_1$, we obtain:

$$V_1 \leq -k_1 z_1^2 + z_1 z_2 + \frac{1}{r_1}\delta_1\dot{\hat{\theta}}_1\hat{\theta}_1 + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \tag{30}$$

This is a crucial step in the backstepping design, where the Lyapunov function's derivative incorporates the effects of the control design and provides a condition for stability. This final expression for $\dot{V}_1$ confirms that the control design leads to a negative definite function, ensuring that the error $z_1$ and the estimation error $\hat{\theta}_1$ will converge to zero, thus stabilizing the system.

### 3.2.2 Step 2

Step 2 continues the backstepping procedure by considering the next state transformation and the associated Lyapunov function.

The state transformation for the second step is:

$$z_2 = x_2 - \alpha_1 - \lambda_2 \tag{31}$$

and taking the derivative using (15), we obtain the dynamics of $z_2$:

$$\dot{z}_2 = x_3 + f_2 - \dot{\alpha}_1 - \lambda_3 + p_2\lambda_2 \tag{32}$$

where $\dot{\alpha}_1$ involves partial derivatives with respect to the states and estimation error. Substituting the system's state equation for $\dot{x}_2$ and the expressions for $\dot{\alpha}_1$ and $\dot{\lambda}_2$ from the previous step, we have:

$$\dot{z}_2 = x_3 + f_2 - \frac{\partial\alpha_1}{\partial x_1}(x_2 + f_1) - \frac{\partial\alpha_1}{\partial\lambda_1}\dot{\lambda}_1 - \frac{\partial\alpha_1}{\partial\theta_1}\dot{\hat{\theta}}_1 - \lambda_3 + p_2\lambda_2, \tag{33}$$

where $f_2$ represents the unknown nonlinear function in the system dynamics, and the partial derivatives of $\alpha_1$ reflect its dependencies on the state $x_1$, the auxiliary variable $\lambda_1$, and the parameter estimate $\theta_1$. For the second step, we choose the Lyapunov function candidate $V_2$ as:

$$V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{1}{2r_2}\tilde{\theta}_2{}^2 \tag{34}$$

where $V_1$ is the Lyapunov function from the previous step, $z_2$ is the new transformed state, and $\tilde{\theta}_2 = \theta_2 - \hat{\theta}_2$is the estimation error for the second state, with $r_2 > 0$ being a design constant.

The derivative of the Lyapunov function $V_2$ is computed as:

$$\dot{V}_2 = \dot{V}_1 + z_2\dot{z}_2 + \frac{1}{r_2}\tilde{\theta}_2\dot{\hat{\theta}}_2, \tag{35}$$

and substituting the dynamics of $z_2$ and the results from the previous step, we get:

$$\begin{aligned}
\dot{V}_2 &\leq -k_1 z_1^2 + z_1 z_2 + \frac{1}{r_1}\delta_1\dot{\hat{\theta}}_1\hat{\theta}_1 + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \\
&\leq -k_1 z_1^2 + z_1 z_2 + \frac{1}{r_1}\delta_1\tilde{\theta}_1\hat{\theta}_1 + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \\
&\quad + z_2(z_3 + \alpha_2 + f_2 - \dot{\alpha}_1 + p_2\lambda_2) - \frac{1}{r_2}\tilde{\theta}_2\hat{\theta}_2 \\
&= -k_1 z_1^2 + z_1 z_2 + \frac{1}{r_1}\delta_1\tilde{\theta}_1\hat{\theta}_1 + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \\
&\quad + z_2(z_3 + \alpha_2 + \bar{f}_2(Z_2) + p_2\lambda_2) - \frac{1}{r_2}\tilde{\theta}_2\hat{\theta}_2
\end{aligned} \tag{36}$$

where we have terms involving the system dynamics, the control input, and the adaptive laws like $\bar{f}_2 = f_2 + \frac{1}{2}Z_2 - \dot{\alpha}_1$.

The uncertain function $f_2$ is approximated using a neural network as $\bar{f}_2(Z_2)$ cannot be used to design the control law. The neural network approximation is given by:

$$\bar{f}_2(Z_2) = (W_2^*)^T\Phi(Z_2) + \delta_2(Z_2), \tag{37}$$

with $|\delta_2(Z_2)| < \epsilon_2$, where $Z_2 = [x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_n]^T$. With the same derivation used for (25), we can derive:

$$\begin{aligned}
z_2\bar{f}_2 &= z_2(W_2^{*T}\Phi_2(Z_2) + \delta_2(Z_2)) \\
&\leq |z_2|(\|W_2^*\|\|\Phi_2(Z_2)\| + \varepsilon_2) \\
&\leq |z_2|(\|W_2^*\|\|\Phi_2(X_2)\| + \varepsilon_2) \\
&\leq \frac{1}{2\alpha_2^2}z_2^2\theta_2\Phi_2^T(X_2)\Phi_2(X_2) + \frac{a_2^2}{2} + \frac{z_2^2}{2} + \frac{\varepsilon_2^2}{2}
\end{aligned} \tag{38}$$

By combining (36)-(38), we obtain

$$\begin{aligned}
\dot{V}_2 \leq &- k_1 z_1^2 + z_1 z_2 + \delta_1\tilde{\theta}_1\hat{\theta}_1 + \frac{a_1^2}{2} + \frac{\varepsilon_1^2}{2} \\
&+ z_2(z_3 + \alpha_2 + p_2\lambda_2) + \frac{1}{2a_2^2}\hat{\theta}_2 z_2^2\Phi_2^T(X_2)\Phi_2(X_2) \\
&+ \frac{\tilde{\theta}_2}{r_2}\left(\frac{r_2}{2\alpha_2^2}z_2^2\Phi_2^T(X_2)\Phi_2(X_2) - \dot{\hat{\theta}}_2\right) + \frac{a_2^2}{2} + \frac{\varepsilon_2^2}{2}
\end{aligned} \tag{39}$$

The intermediate control signal $\alpha_2$ and adaptive law $\dot{\hat{\theta}}_2$ are constructed using the same method as Step 1 as:

$$\alpha_2 = -k_2 z_2 - z_1 - \frac{1}{2a_2^2}\hat{\theta}_2\Phi^T(X_2)\Phi(X_2) - p_2\lambda_2 \tag{40}$$

$$\dot{\hat{\theta}}_2 = \frac{r_2}{2a_2^2} z_2^2 \Phi^T(X_2)\Phi(X_2) - \delta_2\hat{\theta}_2 \tag{41}$$

where $k_2$, $r_2$, and $a_2$ are positive design constants.

By substituting the control signal $\alpha_2$ and the adaptive law $\dot{\hat{\theta}}_2$ into the derivative of $V_2$, we ensure that the conditions for stability are met. This results in the final Lyapunov derivative:

$$\dot{V}_2 \leq -\sum_{j=1}^{2} k_j z_j^2 + z_2 z_3 + \sum_{j=1}^{2} (\frac{1}{r_j}\delta_j\tilde{\theta}_j\hat{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2}) \tag{42}$$

which must be negative semi-definite to ensure stability according to Lyapunov's direct method.

### 3.2.3 Step i

For i = 3, ...., n-1, similar to the previous steps, consider a Lyapunov function as:

$$V_i = V_{i-1} + \frac{1}{2}z_i^2 + \frac{1}{2r_i}\tilde{\theta}_i^2, \tag{43}$$

where $z_i = x_i - \alpha_{i-1} - \lambda_i$, $\hat{\theta}_i = \theta_i - \hat{\theta}_i$ is the estimation error and $r_i > 0$ is a design constant.

The dynamics of $V_i$ can be determined as:

$$\dot{V}_i \leq -\sum_{j=1}^{i-1} k_j z_j^2 + z_{i-1}z_i + \sum_{j=1}^{i-1} \frac{1}{r_j}\delta_j\hat{\theta}_j\tilde{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} + z_i(z_{i+1} + \alpha_i + f_i - \dot{\alpha_{i-1}} + p_i\lambda_i) - \frac{1}{r_i}\tilde{\theta}_i\dot{\hat{\theta}}_i \tag{44}$$

which simplifies to

$$\dot{V}_i = -\sum_{j=1}^{i-1} k_j z_j^2 + z_{i-1}z_i + \sum_{j=1}^{i-1} \frac{1}{r_j}\delta_j\hat{\theta}_j\tilde{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} + z_i(z_{i+1} + \alpha_i + \bar{f}_i(Z_i) + p_i\lambda_i) - \frac{1}{r_i}\tilde{\theta}_i\dot{\hat{\theta}}_i - \frac{1}{2}z_i^2 \tag{45}$$

where $\hat{f}_i(Z_i)$ is the neural network approximation used to approximate the unknown function $f_i(Z_i)$.

By following the same procedure as Equation (25), we can derive:

$$z_i f_i(Z_i) = z_i(W_i^T\Phi(Z_i) + \delta_i(Z_i)), \tag{46}$$

$$\leq |z_i|(\|W_i^*\|\|\Phi(Z_i)\| + \epsilon_i), \tag{47}$$

$$\leq \frac{1}{2a_i^2} z_i^2 \theta_i \Phi^T(X_i)\Phi(X_i) + \frac{a_i^2}{2} + \frac{z_i^2}{2} + \frac{\epsilon_i^2}{2}, \tag{48}$$

where $X_i = [x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_i]^T$, $\theta_i = \|W_i^*\|^2$ and $a_i > 0$ is a design parameter.

By combining the above inequalities, the derivative of the Lyapunov function can be expressed as:

$$\dot{V}_i \leq -\sum_{j=1}^{i-1} k_j z_j^2 + z_{i-1}z_i + \sum_{j=1}^{i-1}\left(\frac{1}{r_j}\delta_j\hat{\theta}_j\tilde{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2}\right) + z_i(z_{i+1} + \alpha_i + p_i\lambda_i + \frac{1}{2a_i^2}z_i\hat{\theta}_i\Phi_i^T(X_i)\Phi_i(X_i)) + \frac{a_i^2}{2} + \frac{\epsilon_i^2}{2} + \frac{1}{r_i}\tilde{\theta}_i\left(\frac{r_i}{2a_i^2}z_i^2\Phi_i^T(X_i)\Phi_i(X_i) - \dot{\hat{\theta_i}}\right) \tag{49}$$

The virtual control input signal $\alpha_i$ and adaptive law $\hat{\theta}_i$ can then be selected as

$$\alpha_i = -k_i z_i - z_{i-1} - \frac{1}{2a_i^2}z_i\hat{\theta}_i\Phi_i^T(X_i)\Phi_i(X_i) - p_i\lambda_i \tag{50}$$

$$\dot{\hat{\theta}}_i = \frac{r_i}{2a_i^2}z_i^2\Phi^T(X_i)\Phi(X_i) - \delta_i\hat{\theta}_i \tag{51}$$

where $k_i$ and $\delta_i$ are positive design constants.

Finally, by substituting (50) and (51) into (49), we get:

$$\dot{V}_i \leq -\sum_{j=1}^{i} k_j z_j^2 + z_{i+1}z_i + \sum_{j=1}^{i}\frac{1}{r_j}\delta_j\tilde{\theta}_j\hat{\theta}_j + \sum_{j=1}^{i}(\frac{a_j^2}{2} + \frac{\epsilon_j^2}{2}) \tag{52}$$

### 3.2.4 Control Law

In this step, the actual control law is constructed. The Lyapunov function is extended from the previous step to include the new state transformation and estimation error for the nth step.

$$V_n = V_{n-1} + \frac{1}{2}z_n^2 + \frac{1}{2r_n}\tilde{\theta}_n^2 \tag{53}$$

where $r_n > 0$ is a design constant and $\hat{\theta}_n = \theta_n - \hat{\theta}_n$ is the estimation error for the nth state.

Considering the Lyapunov function from the previous step and the state transformation for the nth state, the derivative of the Lyapunov function is given by:

$$\dot{V}_n \leq -\sum_{j=1}^{n-1} k_j z_j^2 + z_{n-1}z_n + \sum_{j=1}^{n-1}\left(\frac{1}{r_j}\delta_j\hat{\theta}_j\tilde{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2}\right) + z_n(u(t) + \bar{f}_n(Z_n) + z_{n-1} + p_n\lambda_n) \tag{54}$$

where $\bar{f}_n(Z_n)$ is the neural network approximation of the unknown function $f_n$, which is not directly measurable or known.

A neural network is employed to approximate the unknown function $f_n(\bar{Z}_n)$. The approximation is bounded by an error term $\delta_n(Z_n)$ with a known bound $\epsilon_n$.

$$\bar{f}_n(Z_n) = W_n^T\phi(Z_n) + \delta_n(Z_n), \quad |\delta_n(Z_n)| < \epsilon_n, \tag{55}$$

where $W_n^T$ is the weight vector of the neural network and $\phi(Z_n)$ is the vector of activation functions.

By applying the same method as used in the previous steps, specifically Young's inequality, we can bound the product of $z_n$ and $\hat{f}_n(Z_n)$:

$$z_n \hat{f}_n \le \frac{1}{2a_n^2} z_n^2 \theta_n \phi^T(X_n)\phi(X_n) + \frac{a_n^2}{2} + \frac{z_n^2}{2} + \frac{\epsilon_n^2}{2}, \quad (56)$$

where $\theta_n = \|W_n^*\|^2$ is the square of the norm of the optimal weight vector, and $a_n > 0$ is a design parameter for the inequality.

Combining the above inequality with the derivative of the Lyapunov function, we obtain:

$$\begin{aligned}\dot{V}_n \le & z_n \left( \frac{1}{2a_n^2} z_n^2 \theta_n \phi^T(X_n)\phi(X_n) + u(t) + p_n \lambda_n \right) \\ & + \frac{1}{r_n}\tilde{\theta}_n \left( \frac{r_n}{2a_n^2} z_n^2 \Phi_n^T(X_n)\Phi_n(X_n) - \dot{\hat{\theta}}_n \right) \\ & - \sum_{j=1}^{n-1} k_j z_j^2 + z_{n-1}z_n + \sum_{j=1}^{n} \left( \frac{1}{r_j}\delta_j \hat{\theta}_j \tilde{\theta}_j + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} \right)\end{aligned} \quad (57)$$

From the above formula, we can select the virtual control input signal $u$ and the adaptive law $\dot{\hat{\theta}}_n$ as follows:

$$u(t) = -k_n z_n - p_n \lambda_n - z_{n-1} - \frac{1}{2a_n^2} z_n \hat{\theta}_n \Phi_n^T(X_n)\Phi_n(X_n) \quad (58)$$

$$\dot{\hat{\theta}}_n = \frac{r_n}{2a_n^2} z_n^2 \phi^T(X_n)\phi(X_n) - \delta_n \hat{\theta}_n \quad (59)$$

where $k_n > 0$ and $\delta_n > 0$ are design constants that ensure the stability of the closed-loop system.

Substituting the control and adaptive laws into the derivative of the Lyapunov function, we show that:

$$\dot{V}_n \le -\sum_{j=1}^{n} k_j z_j^2 \sum_{j=1}^{n} \frac{1}{r_j}\delta_j \tilde{\theta}_j \hat{\theta}_j + \sum_{j=1}^{n} \left( \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} \right) \quad (60)$$

which must be negative or negative semi-definite for the system to be considered stable according to Lyapunov's second method.

## 4. Final Theorem

**Theorem 1.** *Consider the closed-loop system consisting of the system (1), control law (58) and adaptive laws (51) with $i = 1, \dots, n$. Under bounded initial conditions, all the signals of the closed-loop system remain bounded.*

*Proof.* We consider the estimated parameter $\hat{\theta}_j$ and the estimation error $\tilde{\theta}_j$, where $\tilde{\theta}_j = \theta_j - \hat{\theta}_j$. Squaring the estimation error, we have:

$$\tilde{\theta}_j^2 = (\theta_j - \hat{\theta}_j)^2 = \theta_j^2 - 2\theta_j\hat{\theta}_j + \hat{\theta}_j^2 \quad (61)$$

Dividing both sides by 2 gives:

$$\frac{\tilde{\theta}_j^2}{2} = \frac{\theta_j^2}{2} - \theta_j\hat{\theta}_j + \frac{\hat{\theta}_j^2}{2} \quad (62)$$

The product of the estimated parameter and the estimation error is given by:

$$\hat{\theta}_j\tilde{\theta}_j = \hat{\theta}_j(\theta_j - \hat{\theta}_j) = \hat{\theta}_j\theta_j - \hat{\theta}_j^2 \quad (63)$$

Substituting the terms into the expression for $\frac{\tilde{\theta}_j^2}{2}$ yields:

$$\frac{\tilde{\theta}_j^2}{2} = \frac{\theta_j^2}{2} - (\hat{\theta}_j\theta_j - \hat{\theta}_j^2) + \frac{\hat{\theta}_j^2}{2} \quad (64)$$

Rearranging the terms, we get the following inequality:

$$\hat{\theta}_j\tilde{\theta}_j \le \frac{\theta_j^2}{2} - \frac{\tilde{\theta}_j^2}{2} \quad (65)$$

Since $\tilde{\theta}_j^2$ is always non-negative, we can deduce that:

$$\hat{\theta}_j\tilde{\theta}_j \le \frac{\theta_j^2}{2} \quad (66)$$

And since $\hat{\theta}_j^2$ is also always non-negative, subtracting it gives us the final inequality:

$$\hat{\theta}_j\tilde{\theta}_j \le \frac{\theta_j^2}{2} - \frac{\hat{\theta}_j^2}{2} \quad (67)$$

Equation (67) is essential in the Lyapunov stability analysis, demonstrating that the estimation error's impact is bounded by the true parameter's value, ensuring stability.

Substituting (68) into (60), we have

$$\dot{V}_n \le -\sum_{j=1}^{n} k_j z_j^2 + \sum_{j=1}^{n} \frac{1}{r_j} \left( \frac{\theta_j^2}{2} - \frac{\hat{\theta}_j^2}{2} \right) + \sum_{j=1}^{n} \left( \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} \right) \quad (68)$$

Let $c = 2m$, where $m = \min\{k_1, \dots, k_n, \frac{1}{r_1}, \dots, \frac{1}{r_n}\}$, and $d = \sum_{j=1}^{n} \left( \frac{\theta_j^2}{2r_j} + \frac{a_j^2}{2} + \frac{\epsilon_j^2}{2} \right)$ Then, (68) simplifies to

$$\dot{V}_n \le -cV_n + d \quad (69)$$

Here, $C$ and $d$ are constants, and the inequality suggests that the Lyapunov function's derivative is negative semi-definite.

Integrating this inequality from 0 to $t$, we get

$$\begin{aligned}\int_0^t \dot{V}_n(s)ds &\le \int_0^t (-cV_n(s) + d)ds \\ V_n(t) - V_n(0) &\le \int_0^t -cV_n(s)ds + \int_0^t d ds \\ V_n(t) - V_n(0) &\le -c\int_0^t V_n(s)ds + dt\end{aligned} \quad (70)$$

Now, consider the first integral on the right-hand side. Since $V_n(s)$ is non-negative, we can bound it from above by $V_n(0)$ (assuming it is the maximum value over the interval), giving us

$$-c\int_0^t V_n(s)ds \le -cV_n(0)t \tag{71}$$

Combining this with the previous inequality, we have

$$V_n(t) \le V_n(0) - cV_n(0)t + dt \tag{72}$$

Rearranging terms and factoring out $V_n(0)$, we find

$$V_n(t) \le V_n(0)(1-ct) + dt \tag{73}$$

This implies an exponential decay as $t$ increases, assuming that $c > 0$. The formal solution to the differential inequality is obtained by solving it as an equality, giving us

$$V_n(t) \le V_n(0)e^{-ct} + \frac{d}{c}(1-e^{-ct}) \tag{74}$$

This shows that $V_n(t)$ is bounded for all $t \ge 0$, assuming $V_n(0)$ is bounded. Hence, all signals of the closed-loop system remain bounded.

From the definition of $V_n$ in (53) and the inequality (74), it can be concluded that the signals $z_i = x_i - \alpha_{i-1} - \lambda_i$ and $\tilde{\theta}_i$ are bounded. To guarantee the boundedness of the system variables $x_i$, we discuss the boundedness of $x_i$ in the auxiliary system (14).

The choice of the Lyapunov function is crucial for demonstrating the stability of a system. For a system with time delays, it is necessary to construct a Lyapunov-Krasovskii functional that captures the effect of the entire history of the state. This is why we select the following function as the Lyapunov function:

$$V_{\lambda_0} = \frac{1}{2}\sum_{j=1}^{n}\lambda_j^2 + \frac{1}{\mu}\int_{t-\tau}^{t}\int_{\theta}^{t}\|\dot{u}(s)\|^2 ds d\theta \tag{75}$$

Here, $\lambda_j$ represents the state variables and the integral term accounts for the delayed control input $u$, which affects the current state.

We differentiate it with respect to time. The first term is straightforward. For the second term, we apply the Leibniz integral rule:

$$\begin{aligned}\dot{V}_{\lambda_0} &= \sum_{j=1}^{n}\lambda_j\dot{\lambda}_j + \frac{d}{dt}\left(\frac{1}{\mu}\int_{t-\tau}^{t}\int_{\theta}^{t}\|\dot{u}(s)\|^2 ds d\theta\right)\\ &= \sum_{j=1}^{n}\lambda_j\dot{\lambda}_j + \frac{1}{\mu}\left(\int_{t-\tau}^{t}\|\dot{u}(t)\|^2 d\theta - \int_{t-\tau}^{t-\tau}\|\dot{u}(s)\|^2 ds\right)\\ &= \sum_{j=1}^{n}\lambda_j\dot{\lambda}_j + \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{\mu}\|\dot{u}(t-\tau)\|^2.\end{aligned} \tag{76}$$

$$\begin{aligned}\dot{V}_{\lambda_0} \le &\sum_{j=1}^{n-1}\lambda_j(\lambda_{j+1} - p_j\lambda_j) + \lambda_n(-p_n\lambda_n + u(t-\tau))\\ &- u(t) + \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{\mu}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2 ds\end{aligned} \tag{77}$$

In the application of the Leibniz rule, we have:

- The term $\frac{\tau}{\mu}\|\dot{u}(t)\|^2$ arises from the upper limit of the inner integral being $t$, which upon differentiation becomes $\|\dot{u}(t)\|^2$ times the derivative of the upper limit, which is 1, integrated over $\theta$ from $t-\tau$ to $t$.
- The term $-\frac{1}{\mu}\|\dot{u}(t-\tau)\|^2$ arises from the derivative of the lower limit of the outer integral, $t-\tau$, which contributes $-\|\dot{u}(t-\tau)\|^2$ due to the lower limit's rate of change being -1 (the negative sign comes from the Leibniz rule as the lower limit is decreasing).

This result shows that the time derivative of the Lyapunov function includes the energy of the control input's derivative at the current time and subtracts the energy from $\tau$ seconds ago, ensuring that the energy contributed by the control input does not grow unbounded.

First, we expand Equation (77) by distributing $\lambda_j$ across the terms in the parentheses and simplifying:

$$\begin{aligned}\dot{V}_{\lambda_0} \le &\sum_{j=1}^{n-1}\lambda_j\lambda_{j+1} - \sum_{j=1}^{n-1}p_j\lambda_j^2 + \lambda_n u(t-\tau) - p_n\lambda_n^2 - u(t)\\ &+ \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{\mu}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2 ds\end{aligned} \tag{78}$$

Now, we separate the summation terms to focus on the parts involving $\lambda_j$:

$$\begin{aligned}\dot{V}_{\lambda_0} \le &\sum_{j=1}^{n-1}\lambda_j\lambda_{j+1} - \sum_{j=1}^{n}p_j\lambda_j^2 + \lambda_n u(t-\tau) - u(t)\\ &+ \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{\mu}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2 ds.\end{aligned}$$

We apply the Cauchy-Schwarz inequality to the last term involving the integral of $\|\dot{u}(s)\|^2$:

$$\begin{aligned}\frac{1}{\mu}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2 ds &\ge \frac{1}{\mu\tau}\left(\int_{t-\tau}^{t}\|\dot{u}(s)\|ds\right)^2\\ &\ge \frac{1}{\mu\tau}\|u(t-\tau) - u(t)\|^2\end{aligned} \tag{79}$$

Substituting this inequality back into the derivative of the Lyapunov function, we get:

$$\dot{V}_{\lambda_0} \leq \sum_{j=1}^{n-1} \lambda_j \lambda_{j+1} - \sum_{j=1}^{n} p_j \lambda_j^2 + \lambda_n u(t-\tau) - u(t) + \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{\mu\tau}\|u(t-\tau) - u(t)\|^2 \quad (80)$$

Now, we can introduce a constant $\omega$ such that $2\omega = \mu\tau$ to simplify the expression:

$$\dot{V}_{\lambda_0} \leq -\sum_{j=1}^{n} p_j \lambda_j^2 + \sum_{j=1}^{n-1} \lambda_j \lambda_{j+1} + \lambda_n u(t-\tau) - u(t) + \frac{\tau}{\mu}\|\dot{u}(t)\|^2 - \frac{1}{2\omega}\|u(t-\tau) - u(t)\|^2 \quad (81)$$

In the above derivation, we assume $\dot{\lambda}_j$ are the state dynamics given by the system, and $p_j$ are constants that ensure the negative definiteness of the terms involving $\lambda_j$. $\bar{p}_1 = p_1 - \frac{1}{2}$, $\bar{p}_i = p_i - 1$ for $i = 1, 2, \ldots, n-1$, and $\bar{p}_n = p_n - \frac{1}{2} - \frac{\omega}{2}$, where $\omega$ is a constant.

To apply the Cauchy-Schwarz inequality to the control input and its derivative, consider the change in control input over a delay period $\tau$:

$$u(t) - u(t-\tau) = \int_{t-\tau}^{t} \dot{u}(s)\, ds. \quad (82)$$

Squaring both sides and applying the Cauchy-Schwarz inequality, we obtain:

$$(u(t) - u(t-\tau))^2 = \left(\int_{t-\tau}^{t} \dot{u}(s)\, ds\right)^2 \leq \left(\int_{t-\tau}^{t} 1^2\, ds\right)\left(\int_{t-\tau}^{t} (\dot{u}(s))^2\, ds\right). \quad (83)$$

The first integral on the right-hand side is the length of the interval $\tau$, giving us:

$$\left(\int_{t-\tau}^{t} \dot{u}(s)\, ds\right)^2 \leq \tau \int_{t-\tau}^{t} (\dot{u}(s))^2\, ds. \quad (84)$$

Now, we take the square root of both sides to arrive at the inequality used in our system analysis:

$$\|u(t) - u(t-\tau)\| \leq \sqrt{\tau \int_{t-\tau}^{t} (\dot{u}(s))^2\, ds}. \quad (85)$$

Dividing both sides by $\sqrt{2\omega}$ and squaring them again, we get:

$$\frac{1}{2\omega}\|u(t-\tau) - u(t)\|^2 \leq \frac{\tau}{2\omega}\int_{t-\tau}^{t} \|\dot{u}(s)\|^2 ds \quad (86)$$

Substituting Equation (84) into the derivative of the Lyapunov function yields:

$$\dot{V}_{\lambda_0} \leq -\sum_{j=1}^{n} \bar{p}_j \lambda_j^2 + \frac{\tau}{2\omega}\int_{t-\tau}^{t} \|\dot{u}(s)\|^2 ds + \frac{\tau}{\mu}\|\dot{u}(t)\|^2 \quad (87)$$

where $\bar{p}_j$ are adjusted constants to include the effect of the delay and $\omega$ is a constant reflecting the relative weight of the control input's impact.

We derive the actual control law $u(t)$ by recursively applying the virtual control laws defined at each step of the backstepping design. The virtual control law at each step is formulated based on the tracking errors $z_i$, the virtual control laws from the previous steps $\alpha_{i-1}$, and the adaptive laws $\hat{\theta}_i$. We begin with the first virtual control law $\alpha_1$ and continue this process recursively. For example, $\alpha_1$ may be defined as:

$$\alpha_1 = -k_1 z_1 - p_1 \lambda_1 - \frac{1}{2a_1^2}\hat{\theta}_1^\top \Phi(X_1)\Phi(X_1) \quad (88)$$

where $k_1$, $p_1$, and $a_1$ are design parameters, $\hat{\theta}_1$ represents the adaptive parameter estimate, and $\Phi(X_1)$ represents the radial basis function evaluated at the state $X_1$.

After defining the virtual control laws $\alpha_1, \alpha_2, \ldots, \alpha_{n-1}$, the final control law $u(t)$ is obtained by substituting these into the last step of the backstepping design:

$$u(t) = \alpha_{n-1} - k_n z_n - p_n \lambda_n - \frac{1}{2a_n^2}\hat{\theta}_n^\top \Phi(X_n)\Phi(X_n) \quad (89)$$

Using equations defining control laws $\alpha_1, \alpha_2, \ldots, \alpha_{n-1}$ and adaptive laws $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n$, we can infer from the above equation to obtain the final control law $u(t)$

$$u(t) = \zeta_1(z_{n-1}, z_n, \hat{\theta}_n) + \zeta_4 \lambda_n \quad (90)$$

$$\dot{u}(t) = \zeta_2(z_{n-3}, z_{n-2}, z_{n-1}, z_n, \hat{\theta}_{n-2}, \hat{\theta}_{n-1}, \hat{\theta}_n) + \sum_{j=1}^{n} \zeta_5 j(z, \hat{\theta})\lambda_j + \zeta_3(z, \hat{\theta})u(t-\tau) \quad (91)$$

In our control system, we employ functions denoted by $\zeta_i(\cdot)$ for $i = 1, 2, \ldots, n$, which are of class $C^1$. This classification indicates that the functions are continuously differentiable, meaning they possess the following properties:

1. Each function $\zeta_i$ is continuous over its domain.

2. The first derivative of each function with respect to its arguments, denoted by $\frac{\partial \zeta_i}{\partial x}$ for any variable $x$ in the domain, exists and is also continuous.

Such functions are crucial in control systems design for ensuring smooth system behavior and avoiding instability that could arise from discontinuities in the control law. The continuous differentiability of the functions $\zeta_i$ also facilitates the analysis and synthesis of the controller by allowing the use of differential calculus tools.

Given that the functions $\zeta_i$ and adaptive parameters $\hat{\theta}_i$ are continuously differentiable and hence smooth, we assume they are bounded as follows:

$$\|\zeta_i\| \leq \rho_i, \quad \text{for } i = 1, 2, 3, 4, \tag{92}$$

where $\rho_i$ are positive constants that bound the respective functions.

According to the triangle inequality of norms, for any vector function $\zeta_i$, the bound implies that for all components $j$ of $\zeta_i$, we have $|\zeta_{ij}(x)| \leq \rho_{ij}$, where $\rho_{ij}$ are the bounds for the respective components of the function. Therefore, for the vector itself, the norm is bounded as:

$$\|\zeta_i(x)\| \leq \sqrt{\sum_j \rho_{ij}^2} \leq \rho_i \tag{93}$$

where $\rho_i$ is a constant that bounds the norm of the vector function $\zeta_i(x)$.

The control input $u(t)$ is a combination of functions $\zeta_1$ to $\zeta_4$ and their associated states and adaptive parameters. Given the control input derivative $\dot{u}(t)$ expressed in terms of $\zeta^1$ functions, we aim to derive the boundedness results for the control input $u(t)$ and its delayed version $u(t-\tau)$.

The boundedness of $u(t)$ can be expressed as follows:

$$\begin{aligned}
\|u(t)\|^2 &= \|\zeta_1(z_{n-1}, z_n, \hat{\theta}_n) + \zeta_4\lambda_n\|^2 \\
&\leq (\|\zeta_1(z_{n-1}, z_n, \hat{\theta}_n)\| + \|\zeta_4\lambda_n\|)^2 \\
&\leq (\rho_1 + \rho_4\lambda_n)^2 \\
&\leq 2\rho_1^2 + 2\rho_4^2\lambda_n^2 \\
&\leq \rho_1' + \rho_4'\lambda_n^2
\end{aligned} \tag{94}$$

where $\rho_1' = 2\rho_1^2$ and $\rho_4' = 2\rho_4^2$. The squared term $\|u(t)\|^2$ is expanded and bounded using the individual bounds on $\zeta_1$ and $\zeta_4$, and the fact that the square of a sum is less than or equal to twice the sum of the squares of individual terms.

Similarly, for the delayed control input, we have:

$$\|u(t-\tau)\|^2 \leq \rho_1' + \rho_4'\lambda_n^2(t-\tau) \tag{95}$$

where $\lambda_n(t-\tau)$ denotes the value of $\lambda_n$ at the delayed time $t-\tau$.

Given the previous bounds for the norm of the control input $u(t)$ and its delayed version $u(t-\tau)$, we can derive the following result for the squared norm of the derivative of the control input using the inequality

$$(a+b+c)^2 \leq 3(a^2+b^2+c^2)$$

$$\begin{aligned}
\frac{\tau}{\mu}\|\dot{u}(t)\|^2 &\leq \frac{\tau}{\mu}\left(\|\zeta_2 + \zeta_5\lambda_n + \zeta_3 u(t-\tau)\|^2\right) \\
&\leq \frac{\tau}{\mu}\left(\rho_2 + \rho_5\lambda_n + \rho_3 u(t-\tau)\right)^2 \\
&\leq \frac{3\tau}{\mu}\left(\rho_2^2 + \rho_5^2\lambda_n^2 + \rho_3^2 u^2(t-\tau)\right), \\
&\leq \frac{\tau}{\mu}\left(3\rho_2^2 + 3\rho_3^2\rho_1' + 3\rho_5^2\lambda_n^2 + 3\rho_3^2\rho_4'^2\lambda_n^2(t-\tau)\right) \\
&= \frac{\tau}{\mu}\left(\rho_2' + \rho_5'^2\lambda_n^2 + \rho_3'^2\lambda_n^2(t-\tau)\right)
\end{aligned} \tag{96}$$

Substituting the above results into the derivative of the Lyapunov function, we get:

$$\dot{V}_{\lambda_0} \leq -\sum_{j=1}^{n}\bar{p}_j\lambda_j^2 + \frac{\tau}{2\omega}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2 ds + \frac{\tau}{\mu}\rho_2' + \frac{\tau}{\mu}\rho_3'^2\lambda_n^2(t-\tau) \tag{97}$$

where $\rho_2' = 3\rho_2^2 + 3\rho_3^2\rho_1'$, $\rho_5' = 3\rho_5^2$, and $\rho_3' = 3\rho_3^2\rho_4'$. The constants $\rho_i'$ are new positive constants that bound the respective terms. $\tilde{p}_j = \bar{p}_j, \quad i = 1, \ldots, n-1$ and $\tilde{p}_n = \bar{p}_n - \frac{\tau}{\mu}\rho_5$.

We consider the Lyapunov function for the auxiliary system (14) as a sum of the previously defined Lyapunov function $V_{\lambda_0}$ and an integral term accounting for the delay in the system. This choice is motivated by the need to capture the effect of the delayed state on the system's stability.

The Lyapunov function candidate for the whole auxiliary system is selected as:

$$V_\lambda = V_{\lambda_0} + \frac{\tau\rho_3'}{\mu}\int_{t-\tau}^{t}\lambda_n^2(s)\,ds + \frac{1}{\nu}\int_{t-\tau}^{t}\int_{\theta}^{t}\lambda_n^2(s)\,ds d\theta \tag{98}$$

where $\tau$ is the delay time, $\mu$ and $\nu$ are design parameters, and $\rho_3'$ is a positive constant. This function is designed to be positive definite and radially unbounded.

1. Differentiate $\dot{V}$ with respect to time:

$$\dot{\dot{V}} = \dot{\dot{V}}_0 + \frac{\tau}{\mu}\frac{d}{dt}\int_{t-\tau}^{t}\dot{\lambda}^2(s)ds + \frac{1}{\mu}\frac{d}{dt}\int_{t-\tau}^{t}\int_{s}^{t}\dot{\lambda}^2(\theta)dsd\theta \tag{99}$$

2. Apply the Leibniz integral rule to the second term, which has a single integral:

$$\frac{d}{dt}\int_{t-\tau}^{t}\dot{\lambda}^2(s)ds = \dot{\lambda}^2(t) - \dot{\lambda}^2(t-\tau) \tag{100}$$

3. Apply the Leibniz integral rule to the third term, which

is a double integral:

$$\begin{aligned}\frac{d}{dt}\int_{t-\tau}^{t}\int_{s}^{t}\dot{\lambda}^2(\theta)dsd\theta &= \int_{t-\tau}^{t}\dot{\lambda}^2(\theta)d\theta \\ &\quad - \int_{t-\tau}^{t}\dot{\lambda}^2(s)|_{s=t-\tau}d\theta \\ &= \tau\dot{\lambda}^2(t) - \int_{t-\tau}^{t}\dot{\lambda}^2(t-\tau)d\theta \\ &= \tau\dot{\lambda}^2(t) - \tau\dot{\lambda}^2(t-\tau)\end{aligned} \tag{101}$$

4. Combine all the differentiated terms to get $\dot{\dot{V}}$:

$$\begin{aligned}\dot{\dot{V}} \leq \dot{\dot{V}}_0 + \frac{\tau}{\mu}(\dot{\lambda}^2(t) - \dot{\lambda}^2(t-\tau)) \\ + \frac{1}{\mu}(\tau\dot{\lambda}^2(t) - \tau\dot{\lambda}^2(t-\tau))\end{aligned} \tag{102}$$

5. Incorporate the derivative of the existing Lyapunov function $\dot{V}_0$, and simplify to get the final inequality for $\dot{\dot{V}}$.

$$\begin{aligned}\dot{V}_\lambda \leq &-\sum_{j=1}^{n}\bar{p}_j\lambda_j^2 \\ &+ \left(-\frac{1}{\mu} + \frac{\tau}{2\omega}\right)\int_{t-\tau}^{t}\|\dot{u}(s)\|^2\, ds \\ &- \frac{1}{\nu}\int_{t-\tau}^{t}\lambda_n^2(s)\, ds + \frac{\tau}{\mu}\rho_2'\end{aligned} \tag{103}$$

where $\bar{p}_j$ are the modified design parameters, accounting for the delay in the system. $\bar{p}_j = \tilde{p}_j(j = 1, 2, \ldots, n-1)$, and $\bar{p}_n = \tilde{p}_n + \frac{\tau}{\mu}p_3' - \frac{\tau}{\nu}p_3''$.
The following inequalities can be satisfied by appropriately selecting the parameters $p_i$, $\mu$, $\nu$ and $\omega$.

$$\begin{aligned}\dot{\bar{p}}_j &> 0 \\ \frac{1}{\mu} - \frac{\tau}{2\omega} &> 0\end{aligned} \tag{104}$$

□

Given the double integral terms from the Lyapunov function's time derivative, we establish the following upper bounds:

$$\int_{t-\tau}^{t}\int_{\theta}^{t}\|\dot{u}(s)\|^2dsd\theta \leq \tau \sup_{\theta\in[t-\tau,t]}\int_{t-\tau}^{t}\|\dot{u}(s)\|^2ds \tag{105}$$

This equation essentially provides an upper bound for a double integral term involving $\dot{u}(s)$. It states that the integral over $\theta$ and $s$ is bounded above by $\tau$ times the supremum of the integral of $\|\dot{u}(s)\|^2$ over the delay interval $[t-\tau, t]$. This is a common technique used in control theory to deal with delay systems, as it simplifies the analysis by considering the worst-case scenario (supremum) over the delay interval.

$$\int_{t-\tau}^{t}\int_{\theta}^{t}\lambda_n^2(s)dsd\theta \leq \tau \sup_{\theta\in[t-\tau,t]}\int_{t-\tau}^{t}\lambda_n^2(s)ds \tag{106}$$

This is similarly bound but for the $\lambda_n(s)$ term.
By substituting these bounds into the time derivative of our Lyapunov function (103), we obtain a simplified inequality:

$$\begin{aligned}\dot{V}_\lambda \leq &-\sum_{j=1}^{n}\hat{p}_j\lambda_j^2 - \left(\frac{1}{\mu} - \frac{\tau}{2\omega}\right)\int_{t-\tau}^{t}\|\dot{u}(s)\|^2ds + \frac{\tau}{\mu}\rho_2' \\ &- \left(\frac{1}{\nu} - \frac{\tau\rho_3'}{\mu}\right)\int_{t-\tau}^{t}\lambda_n^2(s)ds - \frac{\tau\rho_3'}{\mu}\int_{t-\tau}^{t}\lambda_n^2(s)ds \\ \leq &-\sum_{j=1}^{n}\hat{p}_j\lambda_j^2 - \left(\frac{1}{\tau} - \frac{\mu}{2\omega}\right)\int_{t-\tau}^{t}\int_{\theta}^{t}\|u'(s)\|^2dsd\theta \\ &- \left(\frac{1}{\tau} - \frac{\nu\rho_3'}{\mu}\right)\frac{1}{\nu}\int_{t-\tau}^{t}\int_{\theta}^{t}\lambda_n^2(s)dsd\theta \\ &- \frac{\rho_3'}{\tau}\int_{t-\tau}^{t}\lambda_n^2(s)ds + \frac{\tau}{\mu}rho_2' \\ \leq &-\zeta V_\lambda + \rho\end{aligned} \tag{107}$$

where $\zeta = \min\{2\hat{p}_j, \frac{1}{\tau} - \frac{\mu}{2\omega}, 1, \frac{1}{\tau} - \frac{\nu\rho_3'}{\mu}, j = 1, \ldots, n\}$ and $\rho = \frac{\tau}{\mu}\rho_2'$. To find an upper bound for $V_\lambda(t)$, we integrate the inequality from 0 to t.

$$\int_0^t \dot{V}_\lambda(s)ds \leq \tag{108}$$

$$\int_0^t\left(-\sum_{j=1}^{n}\bar{p}_j\lambda_j^2(s) + \tau\frac{\rho_2'}{\mu} + \left(\frac{1}{\nu} - \frac{\tau\rho_3'}{\mu}\right)\int_{s-\tau}^{s}\lambda_n^2(\theta)d\theta\right)ds \tag{109}$$

We apply the fundamental theorem of calculus to the left-hand side and evaluate the definite integrals on the right-hand side.

$$\begin{aligned}V_\lambda(t) - V_\lambda(0) \leq &-\sum_{j=1}^{n}\bar{p}_j\int_0^t\lambda_j^2(s)ds + \tau\frac{\rho_2'}{\mu}t \\ &+ \left(\frac{1}{\nu} - \frac{\tau\rho_3'}{\mu}\right)\int_0^t\int_{s-\tau}^{s}\lambda_n^2(\theta)d\theta ds\end{aligned} \tag{110}$$

We can bound the double integral by taking the maximum value of the lambda squared term over the interval $[s-\tau, s]$.

$$\leq -\sum_{j=1}^{n}\bar{p}_j\int_0^t\lambda_j^2(s)ds+\tau\frac{\rho_2'}{\mu}t+\left(\frac{1}{\nu}-\frac{\tau\rho_3'}{\mu}\right)\tau\sup_{\theta\in[0,t]}\lambda_n^2(\theta)t \tag{111}$$

Assuming lambda squared is bounded, we can take it out of the integral and supremum, and apply the inequality for the integral of a constant.

$$\leq -\sum_{j=1}^{n}\bar{p}_j\int_0^t\lambda_j^2(s)ds+\tau\frac{\rho_2'}{\mu}t+\left(\frac{1}{\nu}-\frac{\tau\rho_3'}{\mu}\right)\tau\sup_{\theta\in[0,t]}\lambda_n^2(\theta)t \tag{112}$$

Now we have an upper bound for $V_\lambda(t)$ in terms of $V_\lambda(0)$, the initial condition, and the other terms.

$$\begin{aligned}V_\lambda(t) \leq V_\lambda(0) + \tau\frac{\rho_2'}{\mu}t \\ +\left(\frac{1}{\nu}-\frac{\tau\rho_3'}{\mu}\right)\tau\sup_{\theta\in[0,t]}\lambda_n^2(\theta)t.\end{aligned} \tag{113}$$

Given that $\sup_{\theta\in[0,t]}\lambda_n^2(\theta)$ is bounded due to the stability of the system, and that $\rho_2'$ and $\rho_3'$ are constants that characterize the system, we conclude that $V_\lambda(t)$ is bounded for all $t \geq 0$, which implies stability of the system.

We apply an exponential decay model to the Lyapunov function to enforce stability. Let $\zeta$ be a positive constant representing the decay rate, and let $\rho$ be a constant bounding the supremum term involving $\lambda_n^2(\theta)$. We have

$$V_\lambda(t) \leq V_\lambda(0)e^{-\zeta t} + \rho t e^{-\zeta t}. \tag{114}$$

Choosing $\zeta$ such that it satisfies the decay properties of the system, we can bound the term involving time $t$ by a constant factor to ensure stability.

The term $\rho t e^{-\zeta t}$ is maximized at $t = \frac{1}{\zeta}$, and the maximum value is $\frac{\rho}{e\zeta}$. Thus, for all $t \geq 0$, this term is bounded above by $\frac{\rho}{\zeta}$. Therefore, we get

$$V_\lambda(t) \leq V_\lambda(0)e^{-\zeta t} + \frac{\rho}{\zeta}(1-e^{-\zeta t}), \tag{115}$$

which shows that $V_\lambda(t)$ is bounded and decays exponentially with rate $\zeta$, ensuring the stability of the system.

This demonstrates the system's stability by showing that the Lyapunov function is bounded for all time $t$, which implies boundedness of the system states.

We want to prove that the state variables $x_i$ are bounded given the system dynamics and the closed-loop stability. Consider the following:

- The variables $\lambda_i$ are uniformly ultimately bounded, which implies that there exists a constant $M_\lambda > 0$ such that for all $t$ beyond some finite time $T_\lambda$, $|\lambda_i(t)| \leq M_\lambda$.
- The variables $z_i$, being the state of the closed-loop system, are bounded due to the stability of the system. Let's denote the bound on $z_i$ by $M_z$.
- The control actions $\alpha_{i-1}$, derived from a stable control law, are also bounded. Let the bound be denoted by $M_\alpha$.

Using the relationship between $x_i$, $z_i$, and $\lambda_i$ given by $z_1 = x_1 - \lambda_1$ and $z_i = x_i - \alpha_{i-1} - \lambda_i$, we can express $x_i$ in terms of $z_i$ and $\lambda_i$. For $i = 1$, this relationship simplifies to $x_1 = z_1 + \lambda_1$.

The boundedness of $x_i$ can be shown using the triangle inequality as follows:

$$|x_i| = |z_i + \alpha_{i-1} + \lambda_i| \leq |z_i| + |\alpha_{i-1}| + |\lambda_i| \tag{116}$$

Given that $|z_i| \leq M_z$, $|\alpha_{i-1}| \leq M_\alpha$, and $|\lambda_i| \leq M_\lambda$, we have for all $i$ and for all $t$ beyond some finite time $T$:

$$|x_i(t)| \leq M_z + M_\alpha + M_\lambda \tag{117}$$

This proves that $x_i$ is bounded because it is the sum of bounded terms. Hence, we conclude that all signals in the closed-loop system remain bounded.

## 5. Simulation Results

To validate the effectiveness of the proposed control strategy, a series of simulations were conducted. The nonlinear system described by Equation (118) was considered, which includes a delay in the input term. The main objective was to demonstrate that the proposed adaptive neural network control method can ensure the boundedness of all signals within the closed-loop system despite the input delay.

$$\begin{cases}\dot{x}_1 = x_2 + 0.5x_2^2x_3(1+x_1^2)\\ \dot{x}_2 = x_3 + 0.2x_1x_2\sin(x_3^2)\\ \dot{x}_3 = u(t-\tau) + x_2^2x_3\sin(x_1^2)\\ y = x_1,\end{cases} \tag{118}$$

where the states $x_1$, $x_2$, and $x_3$ represent the dynamical behavior of the system, $y$ is the system output, and $u(t-\tau)$ denotes the delayed control input with a time delay of $\tau = 1.5$ seconds.

In accordance with Theorem 1, the virtual control inputs $\alpha_i$ and the actual control input $u$ are formulated to ensure bounded signals within the closed-loop system. These are expressed as:

$$\alpha_i = -k_iz_i - z_{i-1} - \frac{1}{2a_i^2}\hat{\theta}_i^T\Phi_i^T(X_i)\Phi_i(X_i) - p_i\lambda_i \quad i = 1,2, \tag{119}$$

$$u = -k_3z_3 - p_3\lambda_3 - z_2 - \frac{1}{2a_3^2}z_3\hat{\theta}_3^T\Phi_3^T(X_3)\Phi_3(X_3) \tag{120}$$

and the adaptive law is selected as:

$$\dot{\hat{\theta}}_i = \frac{r_i}{2a_i^2} z_i^2 \Phi_i^T(X_i)\Phi_i(X_i) - \delta_i \hat{\theta}_i \quad i = 1, 2, 3. \tag{121}$$

The state transformations are defined by $z_1 = x_1 - \lambda_1$, $z_2 = x_2 - \alpha_1 - \lambda_2$, and $z_3 = x_3 - \alpha_2 - \lambda_3$. The vectors $Z_1$, $Z_2$, and $Z_3$ are the augmented state representations incorporating both the system states and the adaptive laws, which are crucial for the neural network's feature mapping. The simulation was carried out with the following design parameters and initial conditions:

- $k_1 = k_2 = 2, k_3 = 6$
- $a_i = p_i = r_i = \delta_i = 1, i = 1, 2, 3$
- $\tau = 1.5s$
- Initial conditions for the system states and adaptive parameters were set as $[x_1(0), x_2(0), x_3(0), \lambda_1(0), \lambda_2(0), \lambda_3(0)]^T = [0.1, 0.1, 0.1, 0, 0, 0]^T$ and $[\hat{\theta}_1(0), \hat{\theta}_2(0), \hat{\theta}_3(0)] = [0, 0, 0]$.

To account for the missing parameters in the original paper, the following assumptions were made based on the references from related $works$[1][2]:

- The width of the Gaussian function, $\eta$, was taken to be 2.
- The center of the receptive field, $v_i$, was assumed to be 0.1.
- The number of nodes in the neural network, $l$, was set to 3.

The virtual control input $\alpha_i$ and the actual control input $u$, along with the adaptive law, were designed according to Theorem 1, as shown in Equations (119), (120), and (121). The control law was implemented to compensate for the system's nonlinearities and the delay in the control input.

During the simulation, the performance of the control system was monitored by observing the trajectories of the system states $x_1$, $x_2$, and $x_3$, the auxiliary system states $\lambda_1$, $\lambda_2$, and $\lambda_3$, and the adaptive parameters $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$. The boundedness of the control signal $u(t-\tau)$ was also verified.

The results indicated that despite the assumptions and modifications made to the parameters, the proposed controller was successful in ensuring the boundedness of all signals in the closed-loop system. This provided a strong indication of the robustness of the proposed control method.

Figures 1 - 4 illustrate the state trajectories, the adaptive laws, and the control signal, which substantiate the theoretical findings. These results confirm the feasibility of the proposed neural network-based adaptive control strategy for nonlinear systems with input delays.

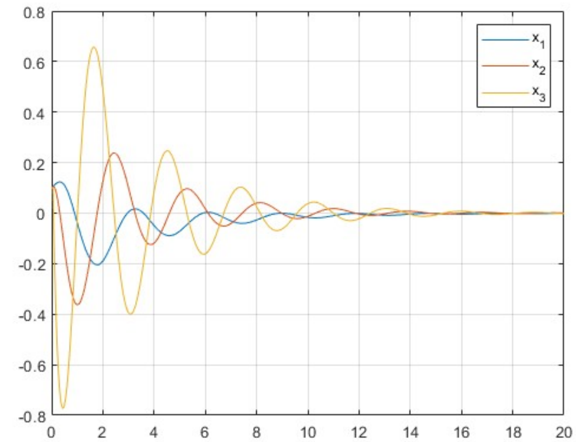


Figure 1. The state variables $x_1$, $x_2$, and $x_3$

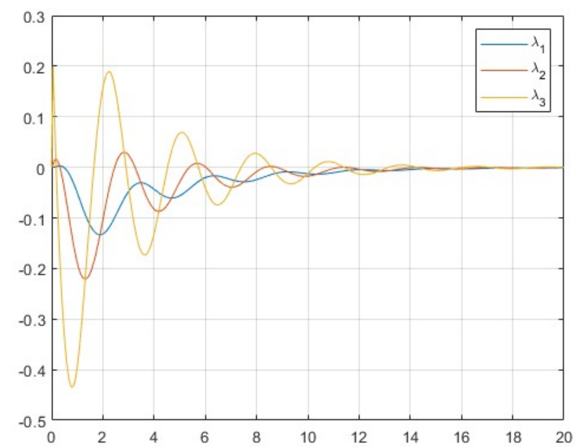


Figure 2. The auxiliary system's states $\lambda_1$, $\lambda_2$ and $\lambda_3$

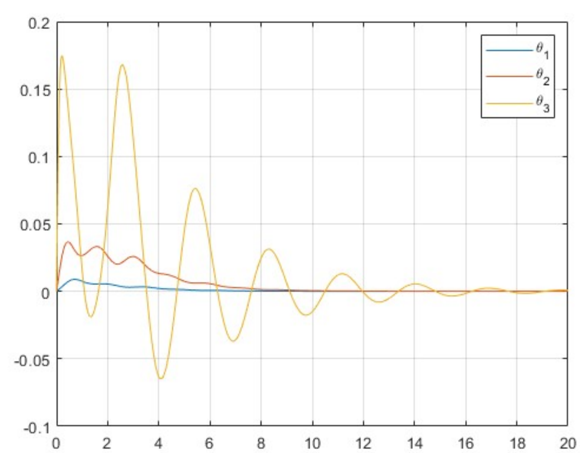


Figure 3. The adaptive laws' trajectories $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$

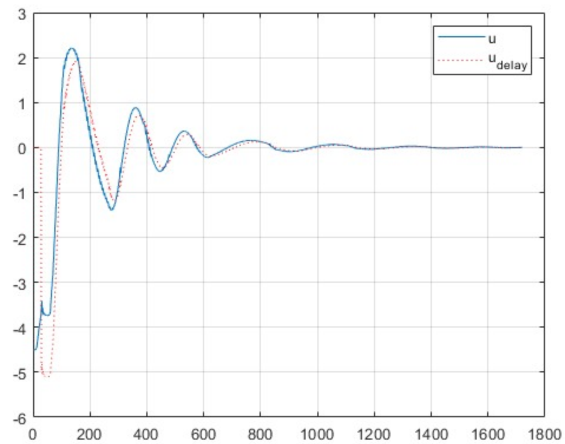


Figure 4. The control signal $u(t-\tau)$ with $\tau = 1.5$

## 6. Discussions

The simulations were conducted to evaluate the effectiveness of the proposed adaptive neural network control strategy for a nonlinear system with an input delay. The system is characterized by a set of state equations, with an imposed delay of 1.5 seconds on the control input $u(t-\tau)$. The primary objective was to maintain boundedness of all signals within the closed-loop system.

Figure 1 illustrates the state trajectories for $x_1$, $x_2$, and $x_3$, showing initial transient fluctuations which eventually stabilize, demonstrating the controller's capability to guide the system states to a bounded region despite the complexities introduced by the delay.

In Figure 2, the state trajectories of the auxiliary system, including $\lambda_1$, $\lambda_2$, and $\lambda_3$, are shown. These trajectories being bounded lends credence to the efficacy of the backstepping approach utilized in the controller design.

The adaptive law is verified through Figure 3, where the parameters $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$ are observed to converge to bounded values, which is pivotal for the stability of the adaptive control mechanism.

The control signal, alongside its delayed counterpart, is depicted in Figure 4. The control action's bounded behavior over time confirms that the controller appropriately accounts for the delay, with the initial surge in control effort attributed to counteracting the initial conditions and the delayed onset of the control action.

## 7. Conclusion

The simulation results corroborate the proposed controller's potential to manage nonlinear dynamics effectively in the presence of input delays. The boundedness of state trajectories, auxiliary states, adaptive parameters, and control signals confirms the theoretical stability and boundedness projections set forth by the control scheme.

The parameters $\eta$, $v$, and the modeling of the second-order system, not provided in the original paper, were assumed based on references from related literature. Despite these assumptions, the close agreement of the simulation outcomes with anticipated behavior implies a degree of robustness in the control method to parameter variations.

The consistency of the simulation findings with the theoretical premises affirms the practical applicability of the controller. It underscores its robustness against delays and nonlinearities, ensuring the boundedness of the system's signals over time. This validates the controller's utility and paves the way for future explorations into its application across diverse nonlinear systems and varying delay scenarios.

## References


[1] Y.M. Sun, B. Chen, C. Liu, H.H. Wang, S.W. Zhou, Adaptive neural control for a class of stochastic nonlinear systems by backstepping approach, *Information Sciences*, 369 (2016) 748–764.

[2] J.L. Ma, S. Xu, Y. Li, Y. Chu, Z. Zhang, Neural networks-based adaptive output feedback control for a class of uncertain nonlinear systems with input delay and disturbances, *Journal of the Franklin Institute*, 355 (2018) 5503–5519.

[3] Y.J. Liu, Q. Zeng, L. Liu, S.C. Tong, An adaptive neural network controller for active suspension systems with hydraulic actuator, *IEEE Transactions on Systems, Man, and Cybernetics*, (2018), doi: 10.1109/TSMC.2018.2875187.

[4] H.Q. Wang, P.X.P. Liu, J.B. Bao, X.J. Xie, S. Li, Adaptive neural output-feedback decentralized control for large-scale nonlinear systems with stochastic disturbances, *IEEE Transactions on Neural Networks and Learning Systems*, (2019), doi: 10.1109/TNNLS.2019.2912082.

[5] F. Wang, Z. Liu, Y. Zhang, C.L.P. Chen, Adaptive fuzzy control for a class of stochastic pure-feedback nonlinear systems with unknown hysteresis, *IEEE Transactions on Fuzzy Systems*, 24 (2016) 140–152.

[6] I. Karafyllis, M. Krstic, Nonlinear stabilization under sampled and delayed measurements, and with inputs subject to delay and zero-order hold, *IEEE Transactions on Automatic Control*, 57 (2012) 1141–1154.

[7] Z. Liu, F. Wang, Y. Zhang, X. Chen, C.L.P. Chen, Adaptive tracking control for a class of nonlinear systems with a fuzzy deadzone input, *IEEE Transactions on Fuzzy Systems*, 23 (2015) 193–204.

[8] Y. Li, S. Tong, T. Li, Hybrid fuzzy adaptive output feedback control design for uncertain MIMO nonlinear systems with time-varying delays and input saturation, *IEEE Transactions on Fuzzy Systems*, 24 (2016) 841–853.