Integral Error-Based Adaptive Neural Identifier for a Class of Uncertain Nonlinear Systems

Donghwa Hong[1] and Kyunghwan Choi[2*]

[1]Department of Mechanical and Robotics Engineering, GIST,
Gwangju, 61005, Republic of Korea (fairytalef00@gm.gist.ac.kr)
[2]Cho Chun Shik Graduate School of Mobility, KAIST,
Daejeon, 34051, Republic of Korea (kh.choi@kaist.ac.kr) * Corresponding author

Abstract: This study proposes an integral error-based adaptive law for neural identifiers, aimed at enhancing the performance of online system identification for nonlinear systems. Unlike conventional adaptive laws that update the neural network based on instantaneous errors, the proposed approach performs updates using identification errors accumulated over time. This mechanism enables the neural network to achieve more consistent and accurate function approximation over the entire time interval, ensuring stable online learning of unknown nonlinear dynamics. A Lyapunov-based theoretical analysis guarantees the uniform ultimate boundedness of the neural identifier. Simulation results on a nonlinear robot manipulator system demonstrate the effectiveness and improved convergence properties of the proposed method compared to a conventional instantaneous error-based approach.

Keywords: Online System Identification, Neural Networks, Adaptive Control, Stability Analysis, Nonlinear Dynamics

1. INTRODUCTION

System identification is the process of estimating the dynamic model of an unknown plant from input-output data, which is essential for controller design and state estimation [1], [2], [3]. In particular, when the system dynamics are nonlinear or time-varying, the importance of online identification becomes more pronounced. Offline identification techniques assume large batches of data and lack real-time adaptability, making them unsuitable for dynamic environments [2], [4]. In contrast, online identification updates the model in real time, enabling the tracking of changing system characteristics, and is thus indispensable in dynamic control applications [1], [2].

Neural network-based identification methods have been actively studied due to their flexible modeling capabilities for nonlinear systems [8], [9]. Neural networks possess universal approximation properties [5], meaning that with appropriate size, structure, and weights, they can approximate any continuous nonlinear function arbitrarily well over a compact set, thus effectively learning complex nonlinearities. For example, multilayer perceptrons or neural networks can be used as identification models to learn input-output relationships.

The conventional backpropagation algorithm updates neural network weights by minimizing an instantaneous squared error cost function, but this approach inherently focuses only on instantaneous errors [6], making it difficult to guarantee long-term parameter convergence. To enhance robustness, stabilization techniques such as σ-modification and composite adaptation have been proposed [11].

The limitations of existing methods are as follows. First, most algorithms use cost functions that minimize instantaneous errors, causing model parameters to focus only on momentary discrepancies and making it difficult to achieve accurate function approximation over the entire time interval. Second, in nonlinear system identification and control, stability proofs are often lacking, making practical application to real systems difficult.

This paper proposes a online neural network identification method that applies a cumulative error cost function with a forgetting factor to overcome the limitations of previous studies. Specifically, by defining a new cost function that accumulates identification errors at all time points, the method aims to minimize the overall error in a balanced manner. The proposed method is rigorously analyzed using Lyapunov-based stability analysis to guarantee the uniform ultimate boundedness of the error.

2. PROBLEM FORMULATION

2.1 Model Dynamics

Consider the nonlinear system

$$\dot{\boldsymbol{x}}(t) = \underbrace{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}_{known} + \underbrace{\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})}_{unknown} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector and $\boldsymbol{u} \in \mathbb{R}^m$ is the control input vector, and $\boldsymbol{f}(\cdot)$ represents the known part of the system dynamics, and $\boldsymbol{h}(\cdot)$ denotes the unknown nonlinear dynamics. It is assumed that the open-loop system (1) is stable, which implies that the state vector $\boldsymbol{x}(t)$ is bounded in L_∞.

2.2 Neural Network Identifier

By adding and subtracting $\mathbf{A}\boldsymbol{x}$ from (1), the system is described by:

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is an arbitrary Hurwitz matrix, $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \mathbf{A}\boldsymbol{x}$, which is the known nonlinear function. A neural network identifier can be constructed by expressing the mapping g in (2) using feedforward nerual network architectures. Thus the identifier model can be selected as:

$$\dot{\hat{\boldsymbol{x}}}(t) = \mathbf{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) + \hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{3}$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^n$ is the estimated state vector, $\hat{\boldsymbol{h}}$ is the estimation of the unknown dynamics $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$. According to the universal approximation theorem [6], a nerual network with a sufficiently large number of hidden layer neurons and an appropriate nonlinear activation function can approximate the unknown dynamics $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$. Thus, the neural network approximation of the unknown dynamics can be expressed as:

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \mathbf{W}\sigma(\mathbf{V}\hat{\bar{\boldsymbol{x}}}) + \epsilon(\boldsymbol{x}) \tag{4}$$

where the input vector to the neural network is defined as $\hat{\bar{\boldsymbol{x}}} = [\hat{\boldsymbol{x}}^T, \boldsymbol{u}^T]^T$, $\mathbf{W} \in \mathbb{R}^{n\times h}$ and $\mathbf{V} \in \mathbb{R}^{h\times(n+m)}$ are the ideal weight matrices of the neural network. $\sigma(\cdot)$ denotes the activation function (e.g., the hyperbolic tangent function $\tanh(\cdot)$), and $\epsilon(\boldsymbol{x}) \leq \epsilon_N$ represents the neural network approximation error, which are assumed to be bounded. And it can be approximated by neural network as follows:

$$\hat{\boldsymbol{h}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) = \hat{\mathbf{W}}\sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}) \tag{5}$$

where, $\hat{\mathbf{W}} \in \mathbb{R}^{n\times h}$ and $\hat{\mathbf{V}} \in \mathbb{R}^{h\times(n+m)}$ are the estimated weight matrices. Defining the errors $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$, $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$, and $\tilde{\mathbf{V}} = \mathbf{V} - \hat{\mathbf{V}}$, the error dynamics are given by:

$$\dot{\tilde{\boldsymbol{x}}} = \mathbf{A}\tilde{\boldsymbol{x}} + \tilde{\mathbf{W}}\sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}) + w(t) \tag{6}$$

where $w(t) = \mathbf{W}\left(\sigma(\mathbf{V}\bar{\boldsymbol{x}}) - \sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})\right) + \epsilon(x)$ is the lumped disturbance term.

In this study, we propose novel adaptive laws for the neural network weights $\hat{\mathbf{W}}$ and $\hat{\mathbf{V}}$ to improve the identification performance. The proposed adaptive laws are designed to ensure more consistent and accurate function approximation over time, as well as stable online learning of the unknown nonlinear dynamics.

3. PROPOSED METHOD

3.1 Conventional Adpative Law

A common approach to update the weights of a neural network is the gradient descent method based on the backpropagated error [7], which minimizes a cost function defined as the squared error. This method updates the weights to reduce the instantaneous error at each time step. Consider the plant model (2) and the identifier model (3). The weights of the neural network are updated according to

$$\dot{\hat{\mathbf{W}}} = -\eta_1\left(\frac{\partial J}{\partial \hat{\mathbf{W}}}\right) - \rho_1\|\tilde{\boldsymbol{x}}\|\hat{\mathbf{W}} \tag{7}$$

$$\dot{\hat{\mathbf{V}}} = -\eta_2\left(\frac{\partial J}{\partial \hat{\mathbf{V}}}\right) - \rho_2\|\tilde{\boldsymbol{x}}\|\hat{\mathbf{V}}, \tag{8}$$

where $\eta > 0$ is the learning rate, $\rho > 0$ is the leakage rate, and $\frac{\partial J}{\partial \hat{\mathbf{W}}}$ and $\frac{\partial J}{\partial \hat{\mathbf{V}}}$ are the gradients of the cost function with respect to the weights. The cost function is defined as

$$J = \frac{1}{2}\tilde{\boldsymbol{x}}(t)^T\tilde{\boldsymbol{x}}(t) \tag{9}$$

where $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ is the state error. The cost function J is minimized by updating the weights of the neural network using the gradient descent method.

However, since the cost function is based solely on the instantaneous error, the model parameters tend to focus on minimizing short-term discrepancies rather than ensuring consistent function approximation over the entire time interval. Consequently, it becomes challenging to guarantee accurate and uniform approximation performance throughout the whole time horizon, particularly when the system dynamics are time-varying. Therefore, it is necessary to consider a cost function that captures the overall behavior of the system in the time domain.

3.2 Integral Adaptive Law based Gradient Descent

To address the limitations of the instantaneous error cost function, an integrated squared error cost functional is defined as follows:

$$J = \frac{1}{2}\int_0^t e^{-\lambda(t-\tau)}\tilde{\boldsymbol{x}}(\tau)^T\tilde{\boldsymbol{x}}(\tau)\mathrm{d}\tau \tag{10}$$

where $\lambda > 0$ is called forgetting factor. This function accumulates the squared error over time, with a forgetting factor that gradually reduces the influence of past errors.

Since the cost functional is of an integral form, the filtered error signal $\boldsymbol{z}(t)$ is introduced to construct the update laws.

$$\boldsymbol{z}(t) = \int_0^t e^{-\lambda(t-\tau)}\tilde{\boldsymbol{x}}(\tau)\mathrm{d}\tau \tag{11}$$

$$\dot{\boldsymbol{z}} = -\lambda\boldsymbol{z} + \tilde{\boldsymbol{x}} \tag{12}$$

Let us define

$$\mathrm{net}_{\hat{\mathbf{V}}} = \hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}$$
$$\mathrm{net}_{\hat{\mathbf{W}}} = \hat{\mathbf{W}}\sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}).$$

Therefore, by using the chain rule $\frac{\partial J}{\partial \hat{\mathbf{W}}}$ and $\frac{\partial J}{\partial \hat{\mathbf{V}}}$ can be computed according to

$$\frac{\partial J}{\partial \hat{\mathbf{W}}} = \frac{\partial J}{\partial \mathrm{net}_{\hat{\mathbf{W}}}} \cdot \frac{\partial \mathrm{net}_{\hat{\mathbf{W}}}}{\partial \hat{\mathbf{W}}}$$
$$\frac{\partial J}{\partial \hat{\mathbf{V}}} = \frac{\partial J}{\partial \mathrm{net}_{\hat{\mathbf{V}}}} \cdot \frac{\partial \mathrm{net}_{\hat{\mathbf{V}}}}{\partial \hat{\mathbf{V}}},$$

where

$$\frac{\partial J}{\partial \mathrm{net}_{\hat{\mathbf{W}}}} = \frac{\partial J}{\partial \tilde{\boldsymbol{x}}}\frac{\partial \tilde{\boldsymbol{x}}}{\partial \hat{\boldsymbol{x}}}\frac{\partial \hat{\boldsymbol{x}}}{\partial \mathrm{net}_{\hat{\mathbf{W}}}} = -\boldsymbol{z}^T\frac{\partial \hat{\boldsymbol{x}}}{\partial \mathrm{net}_{\hat{\mathbf{W}}}},$$
$$\frac{\partial J}{\partial \mathrm{net}_{\hat{\mathbf{V}}}} = \frac{\partial J}{\partial \tilde{\boldsymbol{x}}}\frac{\partial \tilde{\boldsymbol{x}}}{\partial \hat{\boldsymbol{x}}}\frac{\partial \hat{\boldsymbol{x}}}{\partial \mathrm{net}_{\hat{\mathbf{V}}}} = -\boldsymbol{z}^T\frac{\partial \hat{\boldsymbol{x}}}{\partial \mathrm{net}_{\hat{\mathbf{V}}}}$$

and

$$\frac{\partial \text{net}_{\hat{\mathbf{W}}}}{\partial \hat{\mathbf{W}}} = \sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})$$
$$\frac{\partial \text{net}_{\hat{\mathbf{V}}}}{\partial \hat{\mathbf{V}}} = \hat{\bar{\boldsymbol{x}}}.$$

The original BP algorithm is modified such that the static approximations of $\frac{\partial \hat{\boldsymbol{x}}}{\partial \text{net}_{\hat{\mathbf{W}}}}$ and $\frac{\partial \hat{\boldsymbol{x}}}{\partial \text{net}_{\hat{\mathbf{V}}}}$ ($\dot{\hat{\boldsymbol{x}}} = 0$) can be used.

$$\frac{\partial \hat{\boldsymbol{x}}}{\partial \text{net}_{\hat{\mathbf{W}}}} \approx -\mathbf{A^{-1}}$$
$$\frac{\partial \hat{\boldsymbol{x}}}{\partial \text{net}_{\hat{\mathbf{V}}}} \approx -\mathbf{A^{-1}}\hat{\mathbf{W}}(\mathbf{I} - \mathbf{\Lambda}(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})),$$

where $\mathbf{\Lambda}(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}) = \text{diag}\{\sigma_i^2(\hat{\mathbf{V}}_i\hat{\bar{\boldsymbol{x}}})\}, \quad (i = 1, 2, \ldots, m)$. Then, the update laws for the weights $\hat{\mathbf{W}}$ and $\hat{\mathbf{V}}$ can be expressed as:

$$\dot{\hat{\mathbf{W}}} = -\eta_1 \left(\boldsymbol{z}^T \boldsymbol{A}^{-1}\right)^T \sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})^T - \rho_1 \|\tilde{\boldsymbol{x}}\| \hat{\mathbf{W}} \tag{13}$$
$$\dot{\hat{\mathbf{V}}} = -\eta_2 \hat{\bar{\boldsymbol{x}}} \left(\boldsymbol{z}^T \boldsymbol{A}^{-1} \hat{\mathbf{W}} (\mathbf{I} - \mathbf{\Lambda}(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}))\right)^T - \rho_2 \|\tilde{\boldsymbol{x}}\| \hat{\mathbf{V}} \tag{14}$$

Given $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ and $\tilde{\mathbf{V}} = \mathbf{V} - \hat{\mathbf{V}}$, where $\mathbf{W}$ and $\mathbf{V}$ are the fixed ideal wieghts, the weight error dynamics can be rewritten as:

$$\dot{\tilde{\mathbf{W}}} = \eta_1 \left(\boldsymbol{z}^T \boldsymbol{A}^{-1}\right)^T \sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})^T + \rho_1 \|\tilde{\boldsymbol{x}}\| \hat{\mathbf{W}} \tag{15}$$
$$\dot{\tilde{\mathbf{V}}} = \eta_2 \hat{\bar{\boldsymbol{x}}} \left(\boldsymbol{z}^T \boldsymbol{A}^{-1} \hat{\mathbf{W}} (\mathbf{I} - \mathbf{\Lambda}(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}}))\right)^T + \rho_2 \|\tilde{\boldsymbol{x}}\| \hat{\mathbf{V}} \tag{16}$$

3.3 Stability Analysis

To analyze the stability of the system described by (6) with the update laws (13)-(14), Lyapunov's direct method will be utilized. The goal is to demonstrate that the errors $\tilde{\boldsymbol{x}}$, $\tilde{\mathbf{W}}$, and $\tilde{\mathbf{V}}$ are uniformly ultimately bounded.

Theorem 1: For the system given by (6) with the update laws (13)-(14), all signals in the system ($\tilde{\boldsymbol{x}}, \tilde{\mathbf{W}}, \tilde{\mathbf{V}}$) are uniformly ultimately bounded.

Proof: To prove the theorem, the two subsystems are analyzed separately: 1) the estimation error dynamics (6) and the output layer weight error dynamics (15), and 2) the hidden layer weight error dynamics (16). The boundedness of $\tilde{\mathbf{V}}$ does not affect the stability of the first subsystem, as the activation function $\sigma(\hat{\mathbf{V}}\hat{\bar{\boldsymbol{x}}})$ in (15) remains bounded irrespective of $\tilde{\mathbf{V}}$. However, the second subsystem is affected by the first subsystem. Thus, the stability of the first subsystem is established first, followed by the analysis of the second subsystem.

Consider the Lyapunov function candidate for the first subsystem:

$$L = \frac{1}{2}\tilde{\boldsymbol{x}}^T \mathbf{P_1} \tilde{\boldsymbol{x}} + \frac{1}{2}\text{tr}(\tilde{\mathbf{W}}^T \rho_1^{-1} \tilde{\mathbf{W}}) + \frac{1}{2}\int_0^t e^{-\lambda(t-\tau)} \tilde{\boldsymbol{x}}(\tau)^T \mathbf{P_2} \tilde{\boldsymbol{x}}(\tau) \mathrm{d}\tau \tag{17}$$

where $\mathbf{P_1} > 0$ and $\mathbf{P_2} > 0$ are positive definite matrices. The first two terms represent the energy of the error dynamics, while the last term represents the accumulated error over time. After substituting the error dynamics, its time derivative is:

$$\dot{L} = -\frac{1}{2}\tilde{\boldsymbol{x}}^T(\mathbf{Q_1} - \mathbf{P_2})\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T \mathbf{P_1}(\tilde{\mathbf{W}}^{\mathbf{T}} \sigma_{\mathbf{v}} + \mathbf{w}) + \text{tr}(\dot{\tilde{\mathbf{W}}}^{\mathbf{T}} \rho_1^{-1} \tilde{\mathbf{W}}) - \lambda \mathbf{L}_{\text{int}}$$

where $\sigma_v = \sigma(\hat{\mathbf{V}}^T \hat{\bar{\boldsymbol{x}}})$, $\mathbf{Q} = \mathbf{Q_1} - \mathbf{P_2} > \mathbf{0}$, and $\boldsymbol{L}_{\text{int}} = \frac{1}{2}\int_0^t e^{-\lambda(t-\tau)} \tilde{\boldsymbol{x}}(\tau)^T \mathbf{P_2} \tilde{\boldsymbol{x}}(\tau) \mathrm{d}\tau$.

Substituting the update law (13) using $\dot{\tilde{\mathbf{W}}} = -\dot{\hat{\mathbf{W}}}$ yields

$$\text{tr}(\dot{\tilde{\mathbf{W}}}^{\mathbf{T}} \rho_1^{-1} \tilde{\mathbf{W}}) = \eta_{\mathbf{W}} \,\text{tr}(\sigma_{\mathbf{v}} \boldsymbol{z}^{\mathbf{T}} \mathbf{l_1} \tilde{\mathbf{W}}) + \|\tilde{\boldsymbol{x}}\| \,\text{tr}(\hat{\mathbf{W}}^{\mathbf{T}} \tilde{\mathbf{W}})$$

The leakage term can be expanded by substituting $\hat{\mathbf{W}} = \mathbf{W} - \tilde{\mathbf{W}}$:

$$\|\tilde{\boldsymbol{x}}\| \,\text{tr}(\hat{\mathbf{W}}^T \tilde{\mathbf{W}}) = \|\tilde{\boldsymbol{x}}\| \,\text{tr}((\mathbf{W} - \tilde{\mathbf{W}})^{\mathbf{T}} \tilde{\mathbf{W}}) = -\|\tilde{\boldsymbol{x}}\| \|\tilde{\mathbf{W}}\|^2 + \|\tilde{\boldsymbol{x}}\| \|\mathbf{W}\| \|\tilde{\mathbf{W}}\|$$

Substituting this into the $\dot{L}$ expression gives:

$$\begin{aligned}\dot{L} \le &-\frac{1}{2}\tilde{\boldsymbol{x}}^T \mathbf{Q} \tilde{\boldsymbol{x}} - \|\tilde{\boldsymbol{x}}\| \|\tilde{\mathbf{W}}\|^2 - \lambda \boldsymbol{L}_{\text{int}} \\ &+ \tilde{\boldsymbol{x}}^T \mathbf{P_1}(\tilde{\mathbf{W}}^T \sigma_v + w) \\ &+ \eta_W \,\text{tr}(\sigma_v \boldsymbol{z}^T \|\mathbf{A^{-1}} \rho_1^{-1}\| \tilde{\mathbf{W}}) \\ &+ \|\tilde{\boldsymbol{x}}\| \|\mathbf{W}\| \|\tilde{\mathbf{W}}\|\end{aligned}$$

Additionally, the following inequalities hold:

$$\begin{aligned}|\tilde{\boldsymbol{x}}^T \mathbf{P_1} \tilde{\mathbf{W}}^{\mathbf{T}} \sigma_{\mathbf{v}}| &\le \|\tilde{\boldsymbol{x}}\| \|\mathbf{P_1}\| (\|\tilde{\mathbf{W}}\| \sigma_{\mathbf{M}} + \bar{\mathbf{w}}) \\ \|\tilde{\boldsymbol{x}}\| \|\mathbf{W}\| \|\hat{\mathbf{W}}\| &\le \|\tilde{\boldsymbol{x}}\| \|\tilde{\mathbf{W}}\| W_M \\ |\eta_W \,\text{tr}(\sigma_v \boldsymbol{z}^T l_1 \tilde{\mathbf{W}})| &\le \eta_W \|\sigma_v\| \|\boldsymbol{z}\| \|\mathbf{A^{-1}} \rho_1^{-1}\| \|\tilde{\mathbf{W}}\| \\ &\le \eta_W \sigma_M \frac{1}{\lambda} \|\tilde{\boldsymbol{x}}\| \|\mathbf{A^{-1}} \rho_1^{-1}\| \|\tilde{\mathbf{W}}\|.\end{aligned}$$

where $\|W\| \le W_M$, $\|\sigma(\hat{\bar{\boldsymbol{x}}})\| \le \sigma_M$, and because $\boldsymbol{z}(t)$ is the state of the first-order filter (11) driven by $\tilde{\boldsymbol{x}}(t)$, its 2-norm satisfies

$$\begin{aligned}\|\boldsymbol{z}(t)\| &= \left\|\int_0^t e^{-\lambda(t-\tau)} \tilde{\boldsymbol{x}}(\tau) \mathrm{d}\tau\right\| \\ &\le \int_0^t e^{-\lambda(t-\tau)} \|\tilde{\boldsymbol{x}}(\tau)\| \mathrm{d}\tau \\ &\le \|\tilde{\boldsymbol{x}}\|_\infty \int_0^t e^{-\lambda(t-\tau)} \mathrm{d}\tau \\ &= \frac{1 - e^{-\lambda t}}{\lambda} \|\tilde{\boldsymbol{x}}\|_\infty \\ &\le \frac{1}{\lambda} \|\tilde{\boldsymbol{x}}\|_\infty \le \frac{1}{\lambda} \|\tilde{\boldsymbol{x}}\| \quad .\end{aligned}$$

with n denoting the state dimension. Then, the inequality becomes:

$$\begin{aligned}\dot{L} \le &-\frac{1}{2}\lambda_{\min}(\mathbf{Q}) |\tilde{\boldsymbol{x}}\|^2 - \|\tilde{\boldsymbol{x}}\| \|\tilde{\mathbf{W}}\|^2 - \lambda \boldsymbol{L}_{\text{int}} \\ &+ \|\tilde{\boldsymbol{x}}\| \|\mathbf{P_1}\| (\|\tilde{\mathbf{W}}\| \sigma_{\mathbf{M}} + \bar{\mathbf{w}}) \\ &+ \|\tilde{\boldsymbol{x}}\| W_M \|\tilde{\mathbf{W}}\| + \frac{\eta_W \sigma_M}{\lambda} \|\tilde{\boldsymbol{x}}\| \|\mathbf{A^{-1}} \rho_1^{-1}\| \|\tilde{\mathbf{W}}\|\end{aligned}$$

By completing the squares for the terms involving $\|\hat{W}\|$, a sufficient condition on $\|x\|$ can be derived that is independent of the neural network weights error and ensures the time derivative of the Lyapunov candidate is negative.

$$\begin{aligned}\dot{L} \le & -\|\tilde{\boldsymbol{x}}\|\|\tilde{\mathbf{W}}\|^2 + k_b\|\tilde{\boldsymbol{x}}\|\|\tilde{\mathbf{W}}\| \\ & -\frac{1}{2}\lambda_{\min}(\mathbf{Q})\|\tilde{\boldsymbol{x}}\|^2 + \|\tilde{\boldsymbol{x}}\|\|\mathbf{P_1}\|w_M - \lambda \boldsymbol{L}_{\text{int}}\end{aligned} \quad (18)$$

where $k_b = \|\mathbf{P_1}\|\sigma_M + W_M + \frac{\eta_W \sigma_M}{\lambda}\|\mathbf{A^{-1}}\rho_1^{-1}\|$. The terms involving $\tilde{\mathbf{W}}$ are of the form $-(\|\tilde{\boldsymbol{x}}\|)\|\tilde{\mathbf{W}}\|^2 + (k_b\|\tilde{\boldsymbol{x}}\|)\|\tilde{\mathbf{W}}\|$. By completing the square, this is bounded above by $\frac{(k_b\|\tilde{\boldsymbol{x}}\|)^2}{4\|\tilde{\boldsymbol{x}}\|} = \frac{k_b^2}{4}\|\tilde{\boldsymbol{x}}\|$. The final inequality for $\dot{L}$ is:

$$\dot{L} \le -\frac{1}{2}\lambda_{min}(\mathbf{Q})\|\tilde{\boldsymbol{x}}\|^2 - \lambda \boldsymbol{L}_{\text{int}} + \left(\|\mathbf{P_1}\|\bar{w} + \frac{k_b^2}{4}\right)\|\tilde{\boldsymbol{x}}\| \quad (19)$$

To find a sufficient condition that guarantees $\dot{L} \le 0$ and subsequently derive the ultimate bound, a simpler upper bound can be analyzed. Since the term $-\lambda L_{int}$ is always non-positive, it can be omitted from the right-hand side while the inequality still holds. The analysis thus proceeds with the remaining terms:

$$\|\tilde{\boldsymbol{x}}\| \ge \frac{2\left(\|\mathbf{P_1}\|\bar{w} + k_b^2\right)}{\lambda_{min}(\mathbf{Q})} = b \quad (20)$$

Thus, the condition on $\|\tilde{\boldsymbol{x}}\|$ ensures the negative semi-definiteness of $\dot{L}$, leading to the ultimate boundedness of $\tilde{\boldsymbol{x}}$ and the output layer weight error $\tilde{\mathbf{W}}$. In fact, $\dot{L}$ is negative definite outside the ball with radius b.

To show the boundedness of the second subsystem, consider (16) which can be rewritten as:

$$\begin{aligned}\dot{\tilde{\mathbf{V}}} &= \eta_2 \hat{\tilde{\boldsymbol{x}}}\left(\boldsymbol{z}^T \boldsymbol{A}^{-1}\hat{\mathbf{W}}(\mathbf{I} - \boldsymbol{\Lambda}(\hat{\mathbf{V}}\hat{\tilde{\boldsymbol{x}}}))\right)^T + \rho_2\|\tilde{\boldsymbol{x}}\|\hat{\mathbf{V}} \\ &= -\alpha\tilde{\mathbf{V}} + \alpha\mathbf{V} + f_2(\boldsymbol{z}, \hat{\mathbf{W}}, \hat{\mathbf{V}}, \hat{\tilde{\boldsymbol{x}}})\end{aligned}$$

where $f_2(\cdot)$ is a function of the system states and parameters, and $\alpha = \rho_2\|\tilde{\boldsymbol{x}}\|$ is a positive constant. It follows that the term $f_2(\cdot)$ is bounded based on the following arguments: First, since the open-loop system is stable and the ideal wieghts are also constant, it follows that $\hat{\mathbf{W}} \in L_\infty$ and $\hat{\boldsymbol{x}} \in L_\infty$. Second, the function $\Lambda(\cdot)$ is bounded due to the boundedness of hyperbolic tangent function. Third, the filtered error $\boldsymbol{z}(t)$ is also bounded since it is the integral of the state error $\tilde{\boldsymbol{x}}(t)$, which is ultimately bounded by (20). Therefore, the boundedness of $\tilde{\mathbf{V}}$ is also ensured.

Thus, the second subsystem is uniformly ultimately bounded as well. The overall conclusion is that all signals in the system are uniformly ultimately bounded, which completes the proof.

■

4. SIMULATION

4.1 Target Model

To demonstrate the effectiveness of the proposed online neural network identifier, a robot manipulator system is considered as a representative example. The n-degree-of-freedom robot manipulator is modeled by a nonlinear state-space equation derived from the Lagrangian formulation, which is well-suited for the proposed identification approach [10].

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau_d} = \boldsymbol{\tau} \quad (21)$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint position vector, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the joint velocity and acceleration vectors, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis/centrifugal matrix, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the control input, and $\boldsymbol{\tau_d}$ is the unknown friction/damping vector.

The system can be generalized for identification of an unknown nonlinear function by considering a port-Hamiltonian system in which the dynamics are partitioned into known and unknown components in state-space form.

$$\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}}\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial q}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix}\boldsymbol{\tau}}_{\text{known}} - \underbrace{\begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix}\boldsymbol{\tau_d}}_{\text{unknown}} \quad (22)$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, $\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$ is the generalized momentum vector, and $H(\boldsymbol{q}, \boldsymbol{p})$ is the Hamiltonian of the system. The first term on the right-hand side represents the intrinsic system dynamics derived from the port-Hamiltonian system, while the second and third terms correspond to the control input and dissipative torque(friction), respectively. The state vector is denoted as $\boldsymbol{x} = [\boldsymbol{q}^T, \boldsymbol{p}^T]^T \in \mathbb{R}^{2n}$, and the control input as $\boldsymbol{u} = \boldsymbol{\tau} \in \mathbb{R}^n$. The system dynamics can then be expressed in the form of (1).

4.2 Simulation Setup

The simulation setup is as follows. The plant is a 2-DOF robot manipulator described by Eq. (21), with physical parameters given in Table 1.

The reference trajectory for each joint is generated using a fifth-order polynomial (quintic) trajectory, which ensures smooth position, velocity, and acceleration profiles. The initial joint positions and velocities are set to zero. Specifically, the trajectory starts at $\boldsymbol{q}_d(0) = [\frac{\pi}{4}, \frac{\pi}{2}]^T$ and ends at $\boldsymbol{q}_d(T) = [\frac{3\pi}{4}, -\frac{\pi}{2}]^T$, where T is the duration of one cycle. After reaching the endpoint, the trajectory reverses and repeats between these two waypoints, resulting in a periodic motion.

A feedback linearization controller is used to track the reference trajectory:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) \quad (23)$$

where $\boldsymbol{v} = \ddot{\boldsymbol{q}}_d - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) - K_p(\boldsymbol{q} - \boldsymbol{q}_d)$, and K_p, K_d are positive definite gain matrices.

The true friction model includes both viscous and Coulomb friction, given by

$$\boldsymbol{\tau}_d = \begin{bmatrix}f_{c,1}\tanh(\frac{\dot{q}_1}{\sigma_1}) + b_1\dot{q}_1 \\ f_{c,2}\tanh(\frac{\dot{q}_2}{\sigma_2}) + b_2\dot{q}_2\end{bmatrix} \quad (24)$$

where $f_{c,i}$ is the Coulomb friction coefficient, b_i is the viscous friction coefficient, and σ_i is a small positive constant to avoid singularities.

The proposed integral error-based adaptive neural identifier is implemented with a single hidden layer, using the hyperbolic tangent activation function. The hidden layer consists of 20 neurons. The proposed method uses gains of $\eta_1 = 5\times10^1$, $\eta_2 = 5\times10^3$, $\rho_1 = 1\times10^{-3}$, $\rho_2 = 1\times10^{-3}$, and a forgetting factor of $\lambda = 2.5$. And conventional method uses same gains except for forgetting factor. The weights of the neural network are initialized using a uniform random distribution in the range $[-0.01, 0.01]$ for reproducibility. For comparison, the conventional instantaneous error-based adaptive neural identifier was also implemented under the same simulation setup.

Table 1. Physical parameters of the 2-DOF robot manipulator used in simulation

	Description	Value
m	Link mass	2.465 kg
l	Link length	0.2 m
l_c	Link CoM position	0.13888 m
I	Link inertia	0.06911 kg·m^2
I_m	Motor inertia	0.008118 kg·m^2
b	Viscous friction	0.5 N·m·s
f_c	Coulomb friction	0.1 N·m
σ	Smoothing parameter	6.67×10^{-4} N·m·s

4.3 Simulation Results

The performance of the proposed integral error-based adaptive neural identifier is evaluated by comparison with the conventional instantaneous error-based method (7)-(9). As shown in Fig. 1, both methods are able to accurately estimate the unknown dynamics, including the friction forces described in (24), and their estimates converge well over time. Notably, the instantaneous function estimation errors of the two methods show no significant difference, indicating that both approaches achieve comparable accuracy in identifying the unknown nonlinearities present in the system. Both methods also successfully approximate the static friction (stick-slip) effects. The observed transient spikes are primarily due to stick-slip phenomena caused by static friction during trajectory reversals of the robot manipulator. After approximately 280 seconds, the estimation errors of both the conventional and proposed methods converge to within a very small residual error ball and remain bounded thereafter. This behavior is consistent with the theoretical result in Eq. (16), which guarantees that the estimation error ultimately stays within a small neighborhood (error ball) determined by the system and adaptation parameters.

Although the estimation error graphs indicate similar performance between the two methods, a closer examination of the weight matrices $\mathbf{W}$ and $\mathbf{V}$ at specific time instances reveals a clear difference in convergence behavior. Fig 2 illustrates the neural network outputs obtained by feeding the same input trajectory $(\boldsymbol{x}, \boldsymbol{u})$ into the neural network, using the weights $\mathbf{W}$ and $\mathbf{V}$ trained up to 280 s, 320 s, and 360 s, respectively.

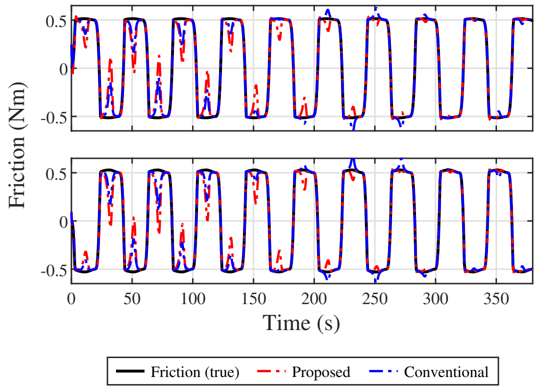

Fig. 1. Estimation performance of the proposed method compared to the conventional method.

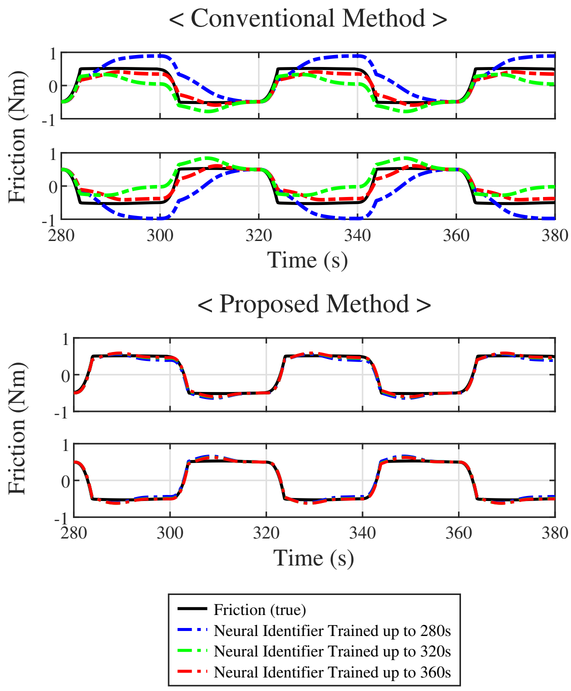

Fig. 2. Feedforward neural network outputs using weights trained up to 280 s, 320 s, and 360 s for the conventional and proposed methods.

The conventional method, which updates weights based solely on instantaneous errors, results in weight matrices that fluctuate significantly over time, as there are many possible weight combinations that can minimize the immediate error at each step. Consequently, the neural network outputs vary at each selected time, indicating a lack of convergence to an ideal set of weights that consistently approximate the true system dynamics.

In contrast, the proposed integral error-based method, which incorporates accumulated past errors, leads to weight matrices that converge more reliably to ideal values. This is reflected in the consistent neural network outputs across the selected time points, demonstrating that the proposed method achieves stable and accurate approximation of the unknown dynamics and friction over the entire time interval. As a result, the proposed method not only reduces instantaneous errors but also ensures overall error minimization, providing improved convergence and robustness in online system identification.

5. CONCLUSION

This paper proposed an integral error-based adaptive law for neural network identifiers, aiming to improve online identification performance for uncertain nonlinear systems. Unlike conventional methods that rely on instantaneous errors, the proposed approach accumulates identification errors over time using a forgetting factor, resulting in more consistent and accurate function approximation. Theoretical analysis based on Lyapunov stability demonstrated the uniform ultimate boundedness of the estimation error. Simulation results on a nonlinear robot manipulator confirmed that the proposed method achieves comparable instantaneous estimation accuracy to conventional approaches, while providing improved convergence of the neural network weights over the entire time interval. Future work includes experimental validation on real robotic systems, extension to broader classes of nonlinear systems, and investigation of computational efficiency for real-time implementation.

REFERENCES

[1] F. Abdollahi, H. A. Talebi, and R. V. Patel, "Stable Identification of Nonlinear Systems Using Neural Networks: Theory and Experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 4, pp. 488–495, Aug. 2006.

[2] R. G. Hart, O. S. Patil, Z. I. Bell, and W. E. Dixon, "System Identification and Control Using Lyapunov-Based Deep Neural Networks without Persistent Excitation: A Concurrent Learning Approach," *arXiv preprint arXiv:2503.21007*, 2025.

[3] M. Indri, I. Lazzero, A. Antoniazza, and A. M. Bottero, "Friction modeling and identification for industrial manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2013, pp. 1–7.

[4] A. L. Bruce, "Learning to Forget: Advanced Online Recursive Identification for Estimation and Adaptive Control," Ph.D. dissertation, Univ. Michigan, 2022.

[5] M. R. Kermani, R. V. Patel, and M. Moallem, "Friction Identification in Robotic Manipulators: Case Studies," in *Proc. IEEE Conf. Control Applications (CCA)*, Aug. 2005, pp. 366–396.

[6] R. H. Nielsen, "Theory of the backpropagation neural network," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 1989, pp. 593–605.

[7] P. Werbos, "Generalization of backpropagation with application to recurrent gas market model," *Neural Networks*, vol. 1, pp. 339–356, 1988.

[8] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[9] Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans, Nonlinear Systems Identification Using Deep Dynamic Neural Networks, *arXiv:1610.01439*, 2016.

[10] M.W. Spong, S. Hutchinson, and M. Vidyasagar, "Robot Modeling and Control," *Wiley*, 2006.

[11] P.A. loannou and J. Sun, "Robust Adaptive Control," *Prentice-Hall*, 1996.