

Preparation of Papers in a Two-Column Format for the International Conference on Control, Automation, and Systems

Donghwa Hong¹ and Kyunghwan Choi^{2*}

¹Department of Mechanical Robotics Engineering, GIST,
Seoul, 13391, Korea (first@hankook.ac.kr)

²Department of Mechanical Engineering, Cho,
Seoul, 13391, Korea (second@hankook.ac.kr) * Corresponding author

Abstract: : This paper presents a novel online system identification method for complex nonlinear systems using neural networks. The proposed method utilizes a cumulative error cost function with a forgetting factor, enabling real-time adaptation to changing system dynamics. The stability of the proposed method is rigorously analyzed using Lyapunov theory, demonstrating Uniform Ultimate Boundedness (UUB) of the identification error. The method is compared with traditional Recursive Least Squares (RLS) and iterative nonlinear parameter estimation methods, highlighting its advantages in terms of adaptability and computational efficiency. Experimental results validate the effectiveness of the proposed approach in identifying unknown nonlinear dynamics in real-time.

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

1. INTRODUCTION

Online system identification is crucial in control applications where system dynamics are unknown or change over time. Real-time adaptability of system dynamics is essential in fields such as smart materials, human-machine interaction, and fault detection. Conventional offline identification methods require large datasets and lack real-time adaptability, making them unsuitable for dynamically changing environments.

Neural networks (NNs) offer powerful capabilities to approximate unknown nonlinearities due to their universal function approximation property. This makes NNs attractive tools for adaptive control and system identification problems where obtaining an accurate model in advance is difficult or impossible.

Most real-world systems are nonlinear, and traditional model-based observers rely on prior knowledge of system nonlinearities, which is rarely satisfied in practice. Neural networks provide an adaptive alternative that can learn the dynamics without such prior knowledge.

Existing research on NN-based adaptive control has mainly focused on tracking error convergence (e.g., minimizing e). However, this study explicitly focuses on identifying the system dynamics themselves (e.g., minimizing $f - \hat{f}$). While NNs have excellent approximation capabilities, their application in control has been primarily centered on optimizing tracking performance. This leaves a gap in the identification of the underlying system dynamics, which this study aims to address. The ability to identify unknown dynamics in real time in dynamically changing environments is a key feature that distinguishes this approach from conventional offline NN training or iterative nonlinear regression methods.

Recursive Least Squares (RLS): RLS achieves faster convergence than LMS (Least Mean Squares) but is mainly applicable to linearly parameterized (LIP) models. It has high computational complexity and generally

requires a persistent excitation (PE) condition for global convergence. Although some variants relax this condition, there are fundamental limitations in directly applying RLS to the nonlinearly parameterized (NIP) structure of deep neural networks, which is the focus of this study.

Iterative Nonlinear Parameter Estimation (e.g., Levenberg-Marquardt): These methods typically rely on iterative procedures to minimize squared residuals offline. They require initial parameter estimates, are susceptible to local minima, and assume accurate model specification. They are not designed to continuously adapt the functional form online, but rather to estimate parameters of a known nonlinear function structure.

RLS and traditional nonlinear regression aim to estimate parameters within a predefined model structure. However, for truly unknown nonlinear functions as considered in this study, the function structure itself is uncertain. Neural networks, with their universal approximation property, can directly approximate functions, bypassing the need for specific parametric models. This provides a higher level of adaptability and applicability to a wide range of complex systems where explicit parameterization is difficult.

This report presents a method for online identification of unknown nonlinear functions using a neural network identifier. The core novelty lies in the cost function that utilizes cumulative error, enabling weight learning via gradient descent using both historical and current data. In particular, a forgetting factor is incorporated to adjust the importance of past data. This approach shares similarities with RLS and traditional nonlinear parameter estimation methods, while also offering distinct advantages.

The main contributions of this study are as follows:

- Development of a robust online neural network identification algorithm for unknown nonlinear dynamics.
- Formal definition and application of a cumulative error cost function with a forgetting factor.

- Rigorous Lyapunov-based stability analysis guaranteeing Uniform Ultimate Boundedness (UUB) of the error.
- Comparative analysis with RLS and traditional nonlinear parameter estimation methods.

This report is organized as follows: introduction, problem definition and neural network identifier structure, proposed online learning algorithm, stability analysis, comparative analysis with related online identification methods, and conclusion and future work.

2. PROBLEM FORMULATION

2.1 Robot Dynamics

The dynamics of an n -degree-of-freedom robot manipulator can be described by the following nonlinear state-space model:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_d = \tau \quad (1)$$

where $q \in \mathbb{R}^n$ is the joint position vector, \dot{q} and \ddot{q} are the joint velocity and acceleration vectors, $M(q) \in \mathbb{R}^{n \times n}$ is the positive definite inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis/centrifugal matrix, $G(q) \in \mathbb{R}^n$ is the gravity vector, and $\tau \in \mathbb{R}^n$ is the control input. To generalize the system for identification of an unknown nonlinear function, we consider a Hamiltonian structure where the dynamics are partitioned into known and unknown components.

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial q} \end{bmatrix}}_{\text{known}} + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix} \tau - \begin{bmatrix} 0 \\ I \end{bmatrix} \tau_d}_{\text{unknown}} \quad (2)$$

where $p = M(q)\dot{q}$ is the generalized momentum, $x = [q^T p^T]^T$ is the state vector composed of joint positions and momenta, and $H(q, p)$ is the Hamiltonian (total energy) of the system. The first term on the right-hand side represents the intrinsic system dynamics derived from the Hamiltonian structure, while the second and third terms correspond to the control input and dissipative torque, respectively.

And so the system can be expressed in the two explicit form:

$$\dot{x}(t) = f(x, u) + h(x, u) \quad (3)$$

where $f(x, u)$ represents the intrinsic system dynamics and control input, while $h(x, u)$ denotes the unknown dissipative torque.

2.2 Neural Network Identifier

To design a identifier, some assumptions are needed as follows :

Assumption 1: The ideal weights W, V are bounded. The disturbance $w(t)$ is bounded by $\|w(t)\| \leq w_M$. The activation function σ and its derivatives are bounded.

Assumption 2 (Open-Loop Stability) The open-loop system (??) is stable, which implies that the state vector $x(t)$

is bounded in L_∞ .

By adding and subtracting Ax from . So, the system is described by:

$$\dot{x}(t) = Ax(t) + g(x, u) + h(x, u) \quad (4)$$

where $A \in \mathbb{R}^{n \times n}$ is an arbitrary Hurwitz matrix, $g(x, u) = f(x, u) - Ax$, which is the known nonlinear function.

Then, the identifier model can be selected as:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + g(x, u) + \hat{h}(\hat{x}) \quad (5)$$

where the NN output is $\hat{h}(\hat{x}, u) = \hat{W}^T \sigma(\hat{V}^T \hat{x})$. Here, $\hat{W} \in \mathbb{R}^{n \times n}$ and $\hat{V} \in \mathbb{R}^{n \times n}$ are the estimated weight matrices. The NN input \hat{x} is constructed from the estimated states \hat{x} and u . Defining the errors $\tilde{x} = x - \hat{x}$, $\tilde{W} = W - \hat{W}$, and $\tilde{V} = V - \hat{V}$, the error dynamics are given by:

$$\dot{\tilde{x}} = A\tilde{x} + \tilde{W}^T \sigma(\hat{V}^T \hat{x}) + w(t) \quad (6)$$

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

3. METHODOLOGY

3.1 Cumulative Error based Gradient Descent

Theorem 1: Consider the plant model (??) and the identifier model (??). Given Assumption 2, if the weights of the NLPNN are updated according to

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

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

where $\eta > 0$ is the learning rate.

$$J = \frac{1}{2} \int_0^t e^{-\lambda(t-\tau)} \tilde{x}(\tau)^T \tilde{x}(\tau) d\tau \quad (9)$$

is the objective function and ρ is a small positive number, then \tilde{x}, \tilde{W} , and $\tilde{V} \in L_\infty$.

Proof: Since the cost functional is of an integral form, we first introduce the filtered error signal $z(t)$ to construct the update laws.

$$z(t) = \int_0^t e^{-\lambda(t-\tau)} \tilde{x}(\tau) d\tau \quad (10)$$

$$\dot{z} = -\lambda z + \tilde{x} \quad (11)$$

Let us define

$$\text{net}_{\hat{V}} = \hat{V} \hat{x} \quad (12)$$

$$\text{net}_{\hat{W}} = \hat{W} \sigma(\hat{V} \hat{x}). \quad (13)$$

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

$$\begin{aligned}\frac{\partial J}{\partial \tilde{\mathbf{W}}} &= \frac{\partial J}{\partial \text{net}_{\tilde{\mathbf{W}}}} \cdot \frac{\partial \text{net}_{\tilde{\mathbf{W}}}}{\partial \tilde{\mathbf{W}}} \\ \frac{\partial J}{\partial \tilde{\mathbf{V}}} &= \frac{\partial J}{\partial \text{net}_{\tilde{\mathbf{V}}}} \cdot \frac{\partial \text{net}_{\tilde{\mathbf{V}}}}{\partial \tilde{\mathbf{V}}},\end{aligned}$$

where

$$\begin{aligned}\frac{\partial J}{\partial \text{net}_{\tilde{\mathbf{W}}}} &= \frac{\partial J}{\partial \tilde{\mathbf{x}}} \frac{\partial \tilde{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{W}}}} = -\mathbf{z}^T \frac{\partial \hat{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{W}}}}, \\ \frac{\partial J}{\partial \text{net}_{\tilde{\mathbf{V}}}} &= \frac{\partial J}{\partial \tilde{\mathbf{x}}} \frac{\partial \tilde{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{V}}}} = -\mathbf{z}^T \frac{\partial \hat{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{V}}}}\end{aligned}\quad (14)$$

and

$$\begin{aligned}\frac{\partial \text{net}_{\tilde{\mathbf{W}}}}{\partial \tilde{\mathbf{W}}} &= \sigma(\hat{\mathbf{V}}\hat{\mathbf{x}}) \\ \frac{\partial \text{net}_{\tilde{\mathbf{V}}}}{\partial \tilde{\mathbf{V}}} &= \hat{\mathbf{x}}.\end{aligned}\quad (15)$$

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

$$\begin{aligned}\frac{\partial \hat{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{W}}}} &\approx -\mathbf{A}^{-1} \\ \frac{\partial \hat{\mathbf{x}}}{\partial \text{net}_{\tilde{\mathbf{V}}}} &\approx -\mathbf{A}^{-1}\tilde{\mathbf{W}}(\mathbf{I} - \mathbf{A}(\hat{\mathbf{V}}\hat{\mathbf{x}})),\end{aligned}\quad (16)$$

where

$$\mathbf{A}(\hat{\mathbf{V}}\hat{\mathbf{x}}) = \text{diag}\{\sigma_i^2(\hat{\mathbf{V}}_i\hat{\mathbf{x}})\}, \quad i = 1, 2, \dots, m. \quad (17)$$

$$\begin{aligned}\dot{\tilde{\mathbf{W}}} &= -\eta_1 (\mathbf{z}^T \mathbf{A}^{-1})^T \sigma(\hat{\mathbf{V}}^T \hat{\mathbf{x}})^T - \rho_1 \|\tilde{\mathbf{x}}\| \tilde{\mathbf{W}} \\ &\quad (18) \\ &\quad (19)\end{aligned}$$

$$\begin{aligned}\dot{\tilde{\mathbf{V}}} &= -\eta_2 \hat{\mathbf{x}} \left(\mathbf{z}^T \mathbf{A}^{-1} \tilde{\mathbf{W}} (\mathbf{I} - \mathbf{A}(\hat{\mathbf{V}}\hat{\mathbf{x}})) \right)^T - \rho_2 \|\tilde{\mathbf{x}}\| \tilde{\mathbf{V}} \\ &\quad (20) \\ &\quad (21)\end{aligned}$$

where $\eta_W, \eta_V, \rho_1, \rho_2 > 0$ are design parameters. ■

3.2 Stability Analysis

To analyze the stability of the system described by (??) with the update laws (??)-(??), we will use Lyapunov's direct method. The goal is to show that the errors $\tilde{\mathbf{x}}$, $\tilde{\mathbf{W}}$, and $\tilde{\mathbf{V}}$ are Uniformly Ultimately Bounded (UUB).

Theorem 2: For the system given by (??) with the update laws (??)-(??), all signals in the system ($\tilde{\mathbf{x}}$, $\tilde{\mathbf{W}}$, $\tilde{\mathbf{V}}$) are Uniformly Ultimately Bounded.

Proof: The stability proof is conducted in two steps using a cascaded system approach.

We first prove the boundedness of the state error $\tilde{\mathbf{x}}$ and the output layer weight error $\tilde{\mathbf{W}}$. This is possible because the term $\sigma_v = \sigma(\hat{\mathbf{V}}^T \hat{\mathbf{x}})$ in the error dynamics (??) is always bounded, regardless of the value of $\hat{\mathbf{V}}$, due to the bounded nature of the activation function σ .

Consider the Lyapunov function candidate for the first subsystem:

$$\begin{aligned}L &= \frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{P}_1 \tilde{\mathbf{x}} + \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}^T \rho^{-1} \tilde{\mathbf{W}}) \\ &\quad + \frac{1}{2} \int_0^t e^{-\lambda(t-\tau)} \tilde{\mathbf{x}}(\tau)^T \mathbf{P}_2 \tilde{\mathbf{x}}(\tau) d\tau\end{aligned}\quad (22)$$

Its time derivative, after substituting the error dynamics, is:

$$\begin{aligned}\dot{L} &= -\frac{1}{2} \tilde{\mathbf{x}}^T (\mathbf{Q}_1 - \mathbf{P}_2) \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T \mathbf{P}_1 (\tilde{\mathbf{W}}^T \sigma_v + \mathbf{w}) \\ &\quad + \text{tr}(\dot{\tilde{\mathbf{W}}}^T \rho^{-1} \tilde{\mathbf{W}}) - \lambda L_{\text{int}}\end{aligned}$$

where $\sigma_v = \sigma(\hat{\mathbf{V}}^T \hat{\mathbf{x}})$, $\mathbf{Q} = \mathbf{Q}_1 - \mathbf{P}_2 > 0$, and $L_{\text{int}} = \frac{1}{2} \int_0^t e^{-\lambda(t-\tau)} \tilde{\mathbf{x}}(\tau)^T \mathbf{P}_2 \tilde{\mathbf{x}}(\tau) d\tau$. We substitute the update law (??) using $\dot{\tilde{\mathbf{W}}} = -\tilde{\mathbf{W}}$.

$$\begin{aligned}\text{tr}(\dot{\tilde{\mathbf{W}}}^T \rho^{-1} \tilde{\mathbf{W}}) &= \text{tr} \left((\eta_W \mathbf{A}^{-T} \mathbf{z} \sigma_v^T + \rho \|\tilde{\mathbf{x}}\| \tilde{\mathbf{W}})^T \rho^{-1} \tilde{\mathbf{W}} \right) \\ &= \eta_W \text{tr}(\sigma_v \mathbf{z}^T l_1 \tilde{\mathbf{W}}) \\ &\quad + \|\tilde{\mathbf{x}}\| \text{tr}(\tilde{\mathbf{W}}^T \tilde{\mathbf{W}})\end{aligned}$$

where $l_1 = \mathbf{A}^{-1} \rho^{-1}$

We expand the leakage term by substituting $\hat{\mathbf{W}} = \mathbf{W} - \tilde{\mathbf{W}}$:

$$\begin{aligned}\|\tilde{\mathbf{x}}\| \text{tr}(\tilde{\mathbf{W}}^T \tilde{\mathbf{W}}) &= \|\tilde{\mathbf{x}}\| \text{tr}((\mathbf{W} - \tilde{\mathbf{W}})^T \tilde{\mathbf{W}}) \\ &= -\|\tilde{\mathbf{x}}\| \|\tilde{\mathbf{W}}\|^2 + \|\tilde{\mathbf{x}}\| \|\mathbf{W}\| \|\tilde{\mathbf{W}}\|\end{aligned}$$

Substituting this back into the \dot{L} expression:

$$\begin{aligned}\dot{L} &\leq -\frac{1}{2} \tilde{\mathbf{x}}^T \mathbf{Q} \tilde{\mathbf{x}} - \|\tilde{\mathbf{x}}\| \|\tilde{\mathbf{W}}\|^2 - \lambda L_{\text{int}} \\ &\quad + \tilde{\mathbf{x}}^T \mathbf{P}_1 (\tilde{\mathbf{W}}^T \sigma_v + \mathbf{w}) + \eta_W \text{tr}(\sigma_v \mathbf{z}^T l_1 \tilde{\mathbf{W}}) \\ &\quad + \|\tilde{\mathbf{x}}\| \|\mathbf{W}\| \|\tilde{\mathbf{W}}\|\end{aligned}$$

Moreover, we have

$$\begin{aligned}|\tilde{\mathbf{x}}^T \mathbf{P}_1 \tilde{\mathbf{W}}^T \sigma_v| &\leq \|\tilde{\mathbf{x}}\| \|\mathbf{P}_1\| (\|\tilde{\mathbf{W}}\| \sigma_M + \bar{\mathbf{w}}) \\ \|\tilde{\mathbf{x}}\| \|\mathbf{W}\| \|\tilde{\mathbf{W}}\| &\leq \|\tilde{\mathbf{x}}\| \|\tilde{\mathbf{W}}\| W_M \\ |\eta_W \text{tr}(\sigma_v \mathbf{z}^T l_1 \tilde{\mathbf{W}})| &\leq \eta_W \|\sigma_v\| \|\mathbf{z}\| l_1 \|\tilde{\mathbf{W}}\| \\ &\leq \eta_W \sigma_M \frac{\sqrt{n}}{\lambda} \|\tilde{\mathbf{x}}\| l_1 \|\tilde{\mathbf{W}}\|.\end{aligned}$$

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

$$\begin{aligned}\|\mathbf{z}(t)\| &= \left\| \int_0^t e^{-\lambda(t-\tau)} \tilde{\mathbf{x}}(\tau) d\tau \right\| \\ &\leq \int_0^t e^{-\lambda(t-\tau)} \|\tilde{\mathbf{x}}(\tau)\| d\tau \\ &\leq \|\tilde{\mathbf{x}}\|_\infty \int_0^t e^{-\lambda(t-\tau)} d\tau \\ &= \frac{1 - e^{-\lambda t}}{\lambda} \|\tilde{\mathbf{x}}\|_\infty \\ &\leq \frac{\sqrt{n}}{\lambda} \|\tilde{\mathbf{x}}\|_\infty \\ &\leq \frac{\sqrt{n}}{\lambda} \|\tilde{\mathbf{x}}\| \quad (\text{since } \|\tilde{\mathbf{x}}\|_\infty \leq \|\tilde{\mathbf{x}}\|).\end{aligned}\quad (23)$$

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

$$\begin{aligned}\dot{L} \leq & -\frac{1}{2}\lambda_{\min}(\mathbf{Q})\|\tilde{\mathbf{x}}\|^2 - \|\tilde{\mathbf{x}}\|\|\tilde{\mathbf{W}}\|^2 - \lambda\mathbf{L}_{\text{int}} \\ & + \|\tilde{\mathbf{x}}\|\|\mathbf{P}_1\|(\|\tilde{\mathbf{W}}\|\sigma_{\mathbf{M}} + \bar{\mathbf{w}}) \\ & + \|\tilde{\mathbf{x}}\|W_{\mathbf{M}}\|\tilde{\mathbf{W}}\| + \eta_{\mathbf{W}}\sigma_{\mathbf{M}}\frac{\sqrt{n}}{\lambda}\|\tilde{\mathbf{x}}\|\|\mathbf{l}_1\|\|\tilde{\mathbf{W}}\|\end{aligned}$$

By completing the squares for the terms involving $\|\tilde{\mathbf{W}}\|$, we look for conditions on $\|\tilde{\mathbf{x}}\|$ which are independent of the neural network weights error and also make the time derivative of the Lyapunov candidate negative.

$$\begin{aligned}\dot{L} \leq & -\|\tilde{\mathbf{x}}\|\|\tilde{\mathbf{W}}\|^2 + k_b\|\tilde{\mathbf{x}}\|\|\tilde{\mathbf{W}}\| \\ & -\frac{1}{2}\lambda_{\min}(\mathbf{Q})\|\tilde{\mathbf{x}}\|^2 + \|\tilde{\mathbf{x}}\|\|\mathbf{P}_1\|\mathbf{w}_{\mathbf{M}} - \lambda\mathbf{L}_{\text{int}}\end{aligned}$$

where $k_b = \|\mathbf{P}_1\|\sigma_{\mathbf{M}} + \mathbf{W}_{\mathbf{M}} + \eta_{\mathbf{W}}\sigma_{\mathbf{M}}\frac{\sqrt{n}}{\lambda}\|\mathbf{A}^{-1}\rho_1^{-1}\|$. The terms involving $\tilde{\mathbf{W}}$ are of the form $-(\|\tilde{\mathbf{x}}\|\|\tilde{\mathbf{W}}\|^2 + (k_b\|\tilde{\mathbf{x}}\|)\|\tilde{\mathbf{W}}\|$. By completing the square, this is bounded above by $\frac{(k_b\|\tilde{\mathbf{x}}\|)^2}{4\|\tilde{\mathbf{x}}\|} = \frac{k_b^2}{4}\|\tilde{\mathbf{x}}\|$. The final inequality for \dot{L} is:

$$\dot{L} \leq -\frac{1}{2}\lambda_{\min}(\mathbf{Q})\|\tilde{\mathbf{x}}\|^2 - \lambda\mathbf{L}_{\text{int}} + \left(\|\mathbf{P}_1\|\bar{\mathbf{w}} + \frac{k_b^2}{4}\right)\|\tilde{\mathbf{x}}\| \quad (24)$$

To find a sufficient condition that guarantees $\dot{L} \leq 0$ and subsequently derive the ultimate bound, we can analyze a simpler upper bound. Since the term $-\lambda\mathbf{L}_{\text{int}}$ 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 :

$$\|\tilde{\mathbf{x}}\| \geq \frac{2(\|\mathbf{P}_1\|\bar{\mathbf{w}} + k_b^2)}{\lambda_{\min}(\mathbf{Q})} = b \quad (25)$$

Furthermore, the above condition on $\|\tilde{\mathbf{x}}\|$ guarantees the negative semi-definiteness of \dot{L} and therefore, ultimate boundedness of $\tilde{\mathbf{x}}$. In fact, \dot{L} is negative definite outside the ball with radius b .

parameter tuning of the learning rates η_1, η_2 and the forgetting factors ρ_1, ρ_2 is crucial for ensuring convergence and stability. The values of these parameters should be chosen based on the specific characteristics of the system being identified, such as the dynamics and noise levels.

4. FIGURES, TABLES, AND EQUATIONS

4.1 Figures and tables

All figures and tables should be placed after their first

mention in the text. Large figures and tables may span across both columns. Scanned images (e.g., line art, photos) can be used if the output resolution is at least 600 dpi.

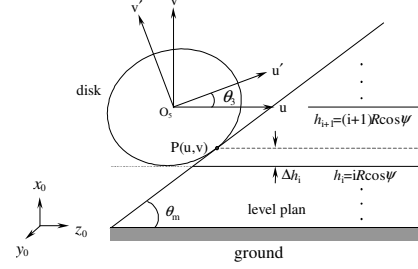


Fig. 1. The caption should be placed after the figure.

Figure captions should be below the figures; table captions should be above the tables. They should be referred to in the text as, for example, Fig. ??, or Figs. 1~3.

	A	B	C
(1)	150%	16.3%	18.2%

Table 1. Your table caption here

4.2 Equations

Equation numbers should be Arabic numerals enclosed in parentheses on the right-hand margin. They should be cited in the text as, for example, Eq. (1), or Eqs. (1)~(3). Equations are located in the middle and equation numbers are located at the end. Punctuate equations with commas or periods when they are part of a sentence. For example,

$$\dot{x} = Ax + Bu, \quad (26)$$

$$y = Cx + Du. \quad (27)$$

4.3 References

References should appear in a separate bibliography at the end of the paper, with items referred to by numerals in square brackets [1, 4-5]. Times New Roman 10pt is used for references.

5. PAGE NUMBERS

Do not put a page number in the manuscript PDF.

REFERENCES

- [1] J. H. Bong, "Controlling the parasite," in *Proc. of International Conference on Control, Automation and Systems*, pp. 0209-0210, 2020
- [2] A. Alice and B. Bob, "Nonlinear unstable systems," *International Journal of Control, Automation, and Systems*, vol. 23, no. 4, pp. 123-145, 1989.
- [3] M. Young, *The Technical Writer's Handbook*, Mill Valley, Seoul, 1989.