International Journal of Electrical and Computer Engineering (IJECE)
Vol. 16, No. 2, April 2026, pp. 717~728
ISSN: 2088-8708, DOI: 10.11591/ijece.v16i2.pp717-728

Adaptive Lyapunov-based control for underactuated nonlinear system using deep neural network

Triya Haiyunnisa, Jony Winaryo Wibowo
Research Center for Smart Mechatronics, National Research and Innovation Agency, Bandung, Indonesia

Article Info

Article history:

Received Sep 30, 2025
Revised Dec 12, 2025
Accepted Jan 15, 2026

Keywords:

Adaptive control
Deep neural network
Lyapunov-based control
Region of attraction
Underactuated nonlinear

ABSTRACT

This paper proposes an adaptive Lyapunov-based control approach using deep neural networks (DNN) for underactuated nonlinear systems, with case studies on the Furuta pendulum and a wheeled path-following system. This approach combines simultaneous learning of the Lyapunov function $V(x)$ to satisfy the positive-definite condition and the control law $u(x)$ to satisfy negative definiteness of $V^{\cdot}(x)$ thus ensuring the asymptotic stability of the system. The proposed model is validated using Python-based simulation. Results show that the proposed method significantly expands the region of attraction (RoA) compared to the linear quadratic regulator (LQR) method. In the Furuta pendulum, the RoA area in the [θ−θ˙] plane increased from 89.04% to 101.14% and in the [α−α˙] plane from 80.28% to 83.79%. Meanwhile, in the wheeled path-following system, the RoA within safety domain increased from 85.28% to 101.69%. Furthermore, robustness tests showed that the controller can maintain tracking performance on a sinusoidal path and reject short disturbances without excessive safety boundary violations. The resulting control signal remained smooth, non-oscillatory, and within the actuator saturation limits, ensuring safe and energy-efficient control. This approach offers a significant contribution by integrating Lyapunov stability theory, deep learning, and online adaptation, resulting a robust and practical for nonlinear underactuated systems.

Corresponding Author:

Triya Haiyunnisa
Research Center for Smart Mechatronics, National Research and Innovation Agency
Bandung 40135, Indonesia
Email: triy016@brin.go.id

1. INTRODUCTION

The control of nonlinear and underactuated systems has long been a fundamental challenge in modern control engineering. Classical approaches such as the linear quadratic regulator (LQR) have been widely used for stabilizing nonlinear systems through local linearization around equilibrium points, as demonstrated in the control of rotary inverted pendulums [1], [2]. While these controllers maintain stability and robustness under small disturbances, they are inherently limited to a narrow operating range due to their dependence on fixed gain matrices and the absence of explicit global Lyapunov stability guarantees [3], [4]. Consequently, controllability and adaptability degrade significantly under large perturbations or parameter uncertainties [5], [6].

To overcome such limitations, several nonlinear control approaches have been proposed. Fuzzy logic-based control [7] has been applied to nonlinear mechatronic systems without requiring internal model knowledge, while backstepping and adaptive control methods [8], [9] improve robustness by recursively stabilizing each subsystem. Similarly, Lyapunov-based control offers a systematic way to guarantee stability and performance in uncertain environments, finding applications in neural networks, adaptive control, and

multi-agent systems [10], [11]. The integration of Lyapunov functions with neural networks increases the prediction effectiveness of control systems under constraints. For instance, control of the angular speed of induction motors using Lyapunov-based neural network model predictive control (NMPC) proved more accurate and robust than conventional approaches [11]. The Lyapunov-based deep neural networks (Lb-DNN) have been developed for nonlinear stochastic systems and accommodate the uncertainties. They adopt an adaptive compensation for drift and diffusion uncertainties, guaranteeing uniformly bounded tracking error in probability [12]. The Lyapunov-based NMPC framework applied to dual-arm manipulators integrates an adaptive mechanism to counteract external disturbances. This method enhances the robustness and stability of the system, overcome challenges under uncertainty [13]. The combination of Lyapunov stability theory with fuzzy logic control systems can adjust control forces real time, which effective in managing complex nonlinear systems. This method has shown the flexibility and effectiveness in controlling various nonlinear systems [14]. In multi-agent systems, Lyapunov-based control method has been done to reduce the effects of false-data-injection attack. By using neural networks for state estimation and stability analysis, these controllers ensure robustness under attack conditions [15]. Recent advances leverage deep neural networks (DNNs) to approximate Lyapunov functions for stability assessment [16], [17]. Rectified linear unit (ReLU) and adaptive weight-update laws guided by Lyapunov theory enable robust tracking in uncertain nonlinear systems [18]–[21].

However, most methods rely on supervised learning with pre-collected data, often neglecting system dynamics during training [22], which can lead to stability violations. While adaptive update laws exist [23], they typically optimize parameters separately rather than jointly learning both Lyapunov functions $V(x)$ and control laws $u(x)$. Moreover, the region of attraction (RoA) is rarely expanded during training, limiting robustness. Furthermore, few studies explicitly address safety-bound constraints or disturbance robustness in adaptive DNN–Lyapunov frameworks, leaving open challenges in achieving reliable stability under real-world uncertainties.

To address these challenges, this paper proposes an Adaptive Lyapunov-based control framework using DNNs for underactuated nonlinear systems. The proposed method employs two neural networks: i) a Lyapunov network that learns a positive-definite function $V(x)$, and ii) an adaptive control network that ensures the negative definiteness of $V(x)$. Both networks are trained jointly using a dynamics-guided feedback process based on the actual nonlinear models of the Furuta pendulum and wheeled path-following systems. A custom composite loss function incorporating RoA expansion and safety-bound constraints is designed to enhance robustness against disturbances while maintaining system stability. The paper is structured as follows: section 2 reviews system dynamics used and Lyapunov theory; section 3 presents the proposed control method; section 4 discusses simulation results and comparisons with LQR; and section 5 concludes the study.

2. BACKGROUND

2.1. System dynamics

An underactuated system is a nonlinear control system with fewer control inputs than degrees of freedom (DOF). Mathematically, it is described by the state-space dynamics:

$$\dot{x}(t) = f(x(t)) + g\big(x(t)u(t)\big) \tag{1}$$

Where $x(t) \in \mathbb{R}^n$ system state vector, $u(t) \in \mathbb{R}^n$ control input vector, $f: \mathbb{R}^n \to \mathbb{R}^n$ drift dynamics, and $g: \mathbb{R}^n \to \mathbb{R}^{n\times m}$ input distribution matrix, with condition $m < n$. This inequality reflects that not all state variable is directly actuated. Their complex dynamics pose challenges for traditional control methods, thus motivating the use of data-driven approaches such as DNNs for learning control laws and Lyapunov functions directly from the system behavior. The Furuta Pendulum and wheeled path-following robot are canonical examples of such systems.

2.1.1. Furuta pendulum

The Furuta pendulum consists of a rotating arm (actuated) and a pendulum (unactuated) mounted on the end of the arm. We used the Furuta pendulum system from Quanser1 to perform simulation [24]. The dynamic equations are expressed as (2), (3):

$$\ddot{\theta} = \phi_1(\psi_1 + u) + \phi_2\psi_2 \tag{2}$$

$$\ddot{\alpha} = \phi_2(\psi_1 + u) + \phi_1\psi_2 \tag{3}$$

With subcomponents:

$$\alpha = J_{arm} + m_p L_r^2 + m_p l_p^2 \sin^2(\alpha) \quad (4)$$

$$b = -m_p l_p L_r \cos(\alpha) \quad (5)$$

$$c = J_p + m_p l_p^2 \quad (6)$$

$$\phi_1 = \frac{c}{ac-b^2}, \phi_2 = \frac{-b}{ac-b^2} \quad (7)$$

$$\psi_1 = -m_p g l_p \sin(\alpha) - m_p l_p^2 \dot{\theta}^2 \sin(\alpha)\cos(\alpha) - B_p \dot{\alpha} \quad (8)$$

$$\psi_2 = -2m_p l_p^2 \dot{\theta}\dot{\alpha} \sin(\alpha)\cos(\alpha) - m_p l_p L_r \dot{\alpha}^2 \sin(\alpha) - B_{arm}\dot{\theta} \quad (9)$$

2.1.2. Wheeled path following

The path following control problem can be modeled using the kinematic bicycle model, which describes the dynamics of a vehicle as it follows a path [25]. The primary goal in path following is to minimize the deviation from a desired trajectory, which is usually defined as a function of the vehicle's position and orientation errors. The dynamics of the vehicle for path following can be expressed as (10), (11):

$$\dot{d_e} = v \sin(\theta_e) \quad (10)$$

$$\dot{\theta_e} = \frac{v}{L}\tan(u) - vk(s)\cos(\theta_e) \quad (11)$$

The goal of the path following control is to design a control law u such that the lateral error d_e and heading error θ_e are minimized, i.e., the vehicle remains close to the desired path with minimal orientation deviation. The equilibrium points relate to the vehicle maintaining a steady trajectory without deviation from the path ($\theta_e = 0, d_e = 0$). All simulations were performed using the nominal parameters listed in Table 1.

Table 1. Model parameter of Furuta pendulum [24]

Nonlinear system	Nomenclature	Value
Furuta Pendulum [24]	m_p - Pendulum mass	0.127 kg
	l_p - Distance from pivot to center of mass	0.156 m
	L_r - Full length of rotary arm	0.216 m
	J_{arm} - Rotary arm moment of inertia about pivot	9.983×10^{-4} kg m^2
	J_p - Pendulum moment of inertia about pivot	0.0012 kg m^2
	B_{arm} - Viscous damping coefficient of arm	0.0024 N m s/rad
	B_p - Viscous damping coefficient of pendulum	0.0024 N m s/rad
	g - Gravity	9.81 m/s^2
Wheeled Path Following [25]	m - Vehicle mass	1.5 kg
	L - Wheelbase	0.165 m
	v - Forward velocity	0.2 m/s
	u – control input (steering angle)	$\pm 30^0$

2.2. Lyapunov stability theory

Lyapunov theory provides a powerful method to certify the stability of nonlinear systems without solving their trajectories.

Lemma 1. Global asymptotic stability via Lyapunov function [25]

Let x = 0 be an equilibrium of the nonlinear system $\dot{x} = f(x)$. If there exists a continuously differentiable function V(x): $\mathbb{R}^n \rightarrow \mathbb{R}$ such that:

$$V(0) = 0, V(x) > 0 \text{ for } x \neq 0$$

$$\dot{V}(x) = \nabla V(x)^T \dot{x} < 0 \text{ for } x \neq 0$$

Then the equilibrium $x = 0$ is globally asymptotically stable.

Definition 1. Input to state stability (ISS) [25]

Consider a nonlinear system in [24], [25]. The system is input-to-state stable (ISS) if there exists a class K$\mathcal{L}$ function $\beta(\cdot,\cdot)$ and a class K function $\gamma(\cdot)$ such that for every initial condition $x(0)$ and bounded input $u(t)$, the solution satisfies:

$$\|x(t)\| \leq \beta(\|x(0)\|, t) + \gamma\left(\sup_{0\leq\tau\leq t}\|u(\tau)\|\right) \tag{12}$$

Where $\beta(\|x(0)\|, t) \to 0$ as $t \to \infty$ is the decay property and $\gamma\left(\sup_{0\leq\tau\leq t}\|u(\tau)\|\right)$ is bounds the influence of input disturbances.

Definition 2. Input-to-state stable (ISS) - Lyapunov function [25]

A function $V(x)$ is an ISS Lyapunov function if there exist class K_∞ functions $\alpha_1, \alpha_2, \alpha_3,$ and σ, such that:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2\|x\|, \forall x \in \mathbb{R}^n \tag{13}$$

$$\dot{V}(x,u) \leq -\alpha_3(\|x\|) + \sigma(\|u\|), \forall x, u \tag{14}$$

This reflects a dissipative energy balance: the state decays unless the input is nonzero. The proposed method minimizes a loss that indirectly enforces $\dot{V}(x,u) < 0$ even in the presence of disturbance $u \approx d(t)$. The design encourages ISS behaviour where the system's stability degrades under perturbation but recovers when the disturbance vanishes.

Definition 3. Safety bounded and region of attraction

The system state $x \in \mathbb{R}^n$. The safety bounded region $S \subset \mathbb{R}^n$ is defined as (15).

$$S = \{x \in \mathbb{R}^n | h_i(x) \leq 0, \forall_i = 1,2,\dots,m\} \tag{15}$$

where each $h_i(x)$ is a differentiable function that represents a safety constraint on a state variable or combination of state variables. The constraints may be upper and lower bounds on each state:

$$x_j{}^{min} \leq x_j \leq x_j{}^{max}, \forall_j \in \{1,\dots,n\}. \tag{16}$$

Region of attraction (RoA), denoted by $\mathscr{R}$ is the subset of state space from which all trajectories converge to the equilibrium under the proposed controller. It is defined as (17).

$$\mathscr{R}: = \{x \in \mathbb{R}^n \mid V(x) < c, \dot{V}(x) < 0\} \tag{17}$$

Where $V(x)$ is positive definite Lypunov function and $c > 0$ denotes a constant sublevel threshold.

3. PROPOSED METHOD

3.1. DNN – based controller design

This work uses 2 DNNs, i.e., LyapunovNN that learns a positive-definite function, and ControlLawNN to ensure the negative definiteness of $\dot{V}(x)$. The LyapunovNN as seen in Figure 1, is a fully connected feed-forward network consisting of two hidden layers (64 and 128 neurons) with SoftPlus activation to ensure that the learned Lyapunov function $V(x)$ remains positive definite for all non-zero states. It maps the four-dimensional state vector $x = [\theta, \dot{\theta}, \alpha, \dot{\alpha}]$ into a single scalar output V(x), representing the system's energy landscape.

The ControlLawNN generates the control signal $u(x)$ according to the nonlinear control law $u(x) = \tanh(k^T x)$ where the initial value of the gain vector k is derived from the optimal gain matrix k obtained by the LQR method. This initialization ensures that the learning process begins from a stabilizing linear policy and then adaptively refines through neural optimization to satisfy Lyapunov stability conditions. The ControlLawNN employs a tanh-based activation to maintain smooth and bounded control outputs, while its parameters are trained jointly with LyapunovNN using a composite loss that penalizes violations of the Lyapunov conditions. Unlike conventional Lyapunov-based controllers, the two DNNs in this framework not only learn the functional relationships of $V(x)$ and $\dot{V}(x)$, but also implicitly determine the RoA using safety bound contrained approach as seen in Figure 2.

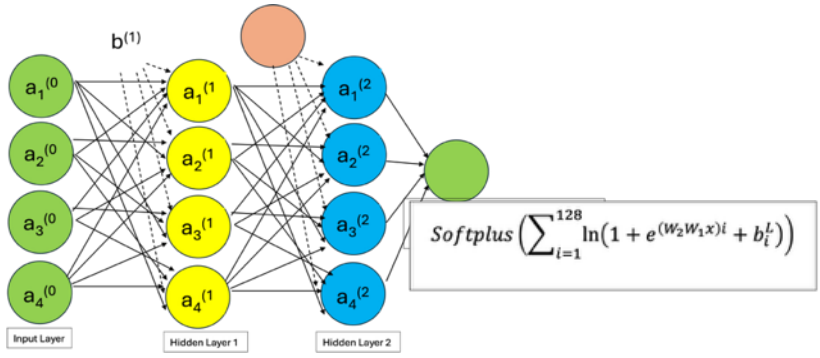


Figure 1. DNN architecture of LyapunovNN

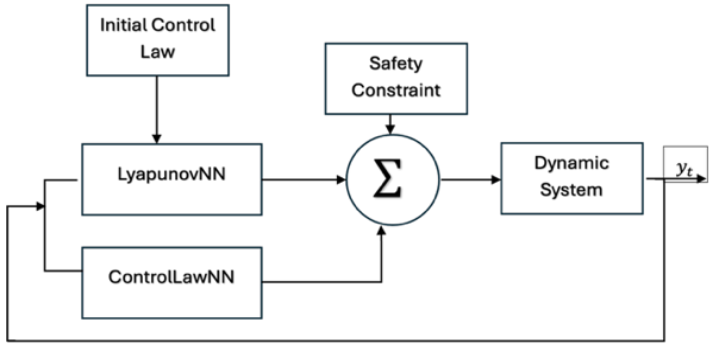


Figure 2. DNN-based controller framework

3.2. Safe and robust region of attraction

In this study, the RoA was further enlarged to approach the safety bound constraint. With the RoA close to the safety bound, the system can handle larger deviations, disturbances, or initialization errors without losing stability, expanding the acceptable operating margin, and reducing the risk of failure even under extreme or unexpected conditions. Robustness is introduced by augmenting the training data with disturbance and parameter variations, allowing the controller to preserve stability under uncertainty. With these following mechanisms, the resulting RoA not only enlarges the stable operating region but also guarantees that all trajectories remain within the predefined safe domain.

3.2.1. Use sublevel set $\{x: V(x) \leq c\}$ and increase c

For positive-definite Lyapunov function $V: \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, the sublevel set $\mathcal{R}(c) = \{x \in \mathbb{R}^n: V(x) \leq c\}$ defines the Region of Attraction (RoA). To enlarge the RoA while maintaining stability ($\dot{V} < 0$), a loss term encourages boundary growth without violating negative definiteness:

$$\mathcal{L}_V = \mathbb{E}_{x\sim\mathcal{D}_c}[\max(0, -V(x) + \epsilon)] + \mathbb{E}_{x\sim\mathcal{D}_c}\left[\max\left(0, \dot{V}(x)\right) + \epsilon_{\dot{V}}\right] \quad (18)$$

where $\mathcal{D}_c$ is sampling distribution near the RoA edge, and $\epsilon, \epsilon_{\dot{V}} > 0$ are small safety margins.

Add scalar $c > 0$ as a trainable parameter, and encourage it to grow.

$$\mathcal{L}_c = -\mu c + \lambda_b \mathbb{E}_{x\sim\partial\mathcal{R}(c)}\left[\max\left(0, \dot{V}(x)\right) + \epsilon_{\dot{V}}\right] \quad (19)$$

Penalty at the boundary $V(x) \approx c$ ensure c value is increased, but $\dot{V}$ stay negative on the edge of RoA. The total Lyapunov loss is $\mathcal{L}_{Lyap} = \mathcal{L}_V + \mathcal{L}_c$

3.2.1. Enclose to safety bounded

To ensure safety near physical limits, the feasible set is defined as $\mathcal{S} = \{x \in \mathbb{R}^n: h_j(x) \geq 0, \forall j\}$, where $h_j(x)$ is barrier function, e.g., $h_\alpha = \alpha_{max} - |\alpha|, h_{\dot{\alpha}}(x) = \dot{\alpha}_{max} - |\dot{\alpha}|$.

A control barrier function (CBF)-based loss so $\mathcal{S}$ forward-invariant:

$$\mathcal{L}_{saftey} = \mathbb{E}_x \sum_j \max\left(0, -\left(\nabla h_j f(x, u(x)) + \alpha_b h_j(x)\right)\right), \tag{20}$$

where $\alpha_b > 0$ and the total loss safe region loss becomes

$$\mathcal{L}_{enclose} = \mathcal{L}_{Lyap} + \mathcal{L}_{safety} \tag{21}$$

This safety bounded approach ensures that all sublevel sets of $V(x)$ remain inside $\mathcal{S}$.

3.2.2. Adaptive control and robustness

The control law $u(x) = \tanh(k^T x)$, initialized from LQR gain K, is adaptively tuned to expand the RoA while ensuring safety. The combined loss is used in the controller:

$$\mathcal{L}_u = \lambda_{track}\, z\left\|y - y_{ref}\right\|^2 + \lambda_{\dot{V}} \mathbb{E}_{x\sim\mathcal{D}_c}\left[\max\left(0, \dot{V}(x) + \epsilon_{\dot{V}}\right)\right] + \lambda_u \mathbb{E}[\|u\|^2] \tag{22}$$

After the adaptive tracking process is complete and the control parameters reach convergence, further testing is carried out by providing external disturbances to the system to evaluate the controller's ability to maintain stability.

$$\mathcal{L}_{rob} = \mathbb{E}_{x\sim\mathcal{D}_c}\left[\max_{d\in\mathcal{D}}\left(\dot{V}(x; d) + \epsilon_{\dot{V}}\right)\right]_+ \tag{23}$$

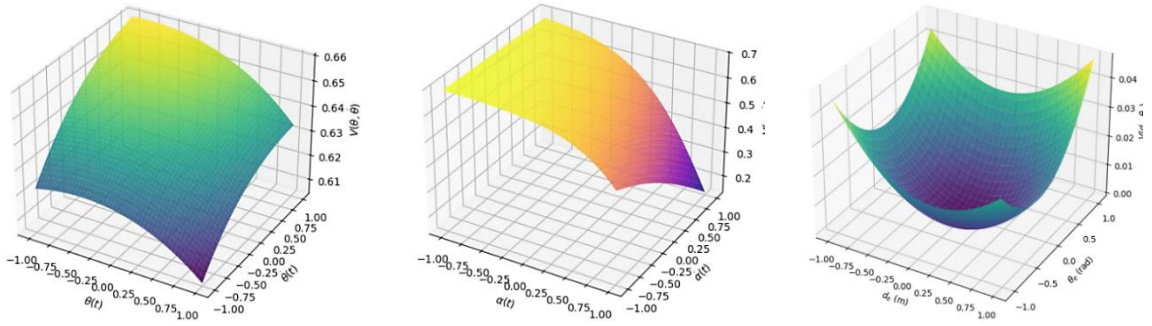
In the end, the total loss function is used so that the RoA resulting from the iteration will increase, be stable, and stick to the safety limit from the inside is $\mathcal{L}_{total} = \mathcal{L}_{enclose} + \lambda_{rob}\mathcal{L}_{rob} + \mathcal{L}_u$.

All simulations and training processes were implemented in Python 3.10 using the PyTorch deep learning framework. The system dynamics of the Furuta pendulum were numerically integrated using NumPy and SciPy libraries, and visualization was performed in Matplotlib.

4. RESULT AND DISCUSSION

4.1. Lyapunov function and control law

In the simulation implementation, LyapunovNN and ControlLawNN are run simultaneously. However, in this discussion, both components are explained separately to provide a deeper understanding of their respective roles. The first section discusses the process of forming the Lyapunov function, which includes the design of the neural network architecture, the training process to obtain a positive definite $V(x)$, and the visualization of the bowl-like shape of the Lyapunov surface to demonstrate global stability. As shown in Figure 3(a)-3(c), the resulting Lyapunov surface exhibits a convex, bowl-like structure with a minimum at the equilibrium point, indicating that the learned Lyapunov function is positive definite over the considered state space.



(a) (b) (c)

Figure 3. Lyapunov contour for (a) rotary arm; (b) inverted pendulum of Furuta pendulum, and (c) Lyapunov contour of the wheeled path-following system

In the Furuta pendulum system, LyapunovNN training is automatically terminated at the 25th epoch using an early stopping mechanism after the Lyapunov criterion is met. Meanwhile, in the wheeled path-

following system, LyapunovNN was trained up to the 29th epoch before automatically stopping. The training results show that the obtained Lyapunov function surface is bowl-shaped with a minimum value at the equilibrium point.

Control law ($u(x)$) is used to minimize the Lyapunov derivative so that negative values of $\dot{V}(x)$ can be achieved. The change in the value of u(x) is obtained from tuning the value of K with $K_{initial}$, which is the result of LQR calculations. As a result, the gain value changes from $K_{initialFP}$=[3.1623, 3.5076, 54.9723, 8.0055] to $K_{finalFP}$=[1.1793, 0.7417, 0.2619, −0.9536] and for Wheeled path following $K_{initialWP}$=[3.1623, 10.1569] to $K_{finalWP}$=[0.0122, −1.9473], which shows that adaptive control softens the response at some states, and adjusts the gain so that Lyapunov stability is not only guaranteed locally as in LQR, but also in a wider nonlinear domain according to the design objectives.

4.2. Region of attraction (RoA) with DNN-based controller

To evaluate the performance of the proposed method, a comparative analysis of RoA area was conducted against conventional control methods based on the LQR. LQR was chosen as a comparison because it is one of the most widely used optimal approaches for stabilizing nonlinear systems around an equilibrium point. The RoA was visualized in 2D slices to facilitate interpretation of the results. The Furuta pendulum has four state variables $(\theta, \dot{\theta}, \alpha, \dot{\alpha})$, the RoA fully resides in 4-dimensional space, which is difficult to visualize directly. Therefore, the RoA is displayed per two state variables—for example, in the planes [θ,θ˙] and [α,α˙]. The RoA comparison results in Figures 4(a) and 4(b) show a significant difference between the conventional LQR-based approach and the proposed DNN controller-based method with adaptive feedback learning, where the RoA with the DNN-based controller produces a RoA 2 times wider, compared to the RoA produced by the LQR method. A similar trend is seen in the wheeled path following system in Figure 4(c). Although the resulting RoA expansion is not as large as in the Furuta pendulum, the proposed method still produces a larger RoA. Therefore, it can be concluded that the DNN approach not only expands the RoA but also improves the system's ability to achieve stability from various initial conditions, providing better advantages over the classical LQR approach, especially in nonlinear systems and wider domains.

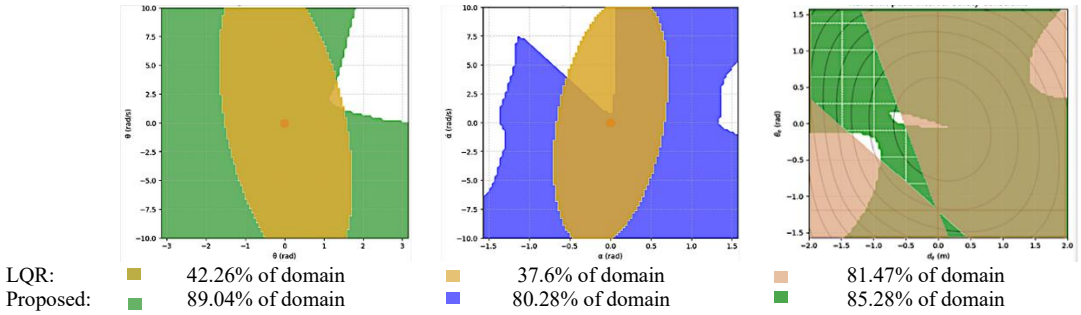


Figure 4. Comparison of RoA: Furuta pendulum (a) [θ–θ˙], (b) [α-α˙] plane; and (c) wheeled path following

4.3. Robustness test

4.3.1. Online adaptive tracking with feedback-based learning

In this study, an online adaptive tracking with feedback-based learning approach is used to ensure that the Furuta pendulum can follow the reference signal $\theta_{ref}(t)$ with high precision while maintaining system safety. Simulation results show that the proposed adaptive controller can track the angle θ according to the given sinusoidal reference, while keeping the system within the safe limits. Based on the Figure 5 shows that the average tracking error is very small and controlled over a 20-second time horizon, with the error contribution decreasing over time (evidenced by the low mean squared error - MSE value). Furthermore, the penalty mechanism applied to the loss function is proven effective in keeping the angle α (pendulum position) within the operating range of $|\alpha| \leq 1$ radian, thus preventing the pendulum from moving outside the stable region. The angular velocity limit parameters $\dot{\theta}$ and $\dot{\alpha}$ are also respected, with oscillations decreasing over time so that the system operates safely without exceeding ±3 rad/s.

In addition to the Furuta pendulum, tracking for sinusoidal paths was also performed on wheeled path following. The main objective was to minimize the lateral error de and heading error θe, while maintaining the vehicle state within the safety bounded. The adaptive DNN controller significantly reduced the error after the first few seconds, as seen in Figure 6, although there were still safety bound violations that needed to be mitigated. The resulting MSE was quite small for the sinusoidal path scenario. This indicates that DNN-based adaptive control is effective in addressing model uncertainty in path-following systems.

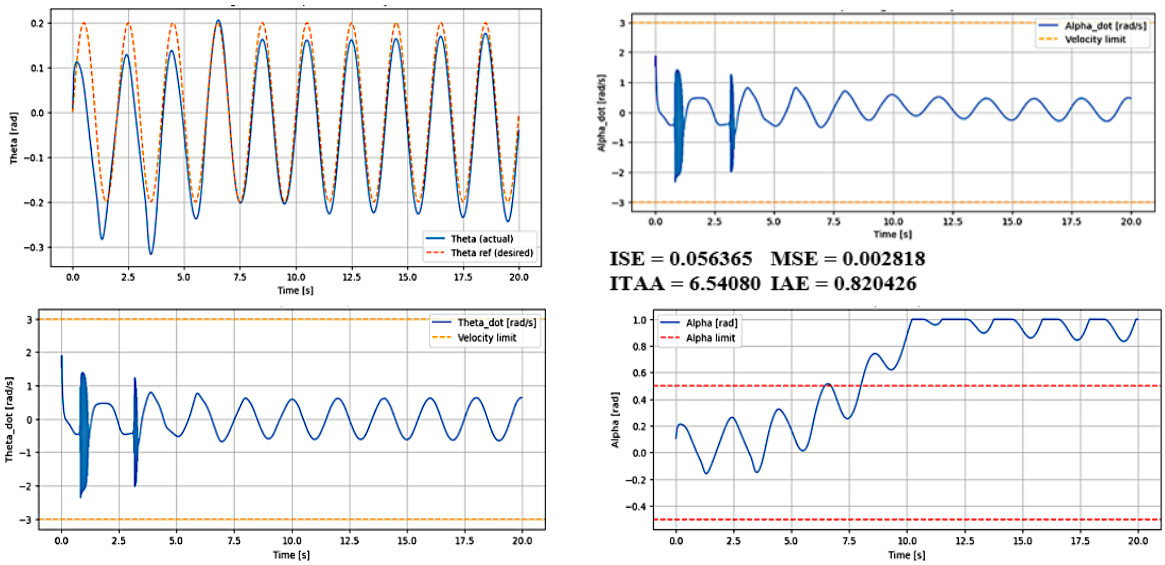


Figure 5. Theta tracking in Furuta pendulum

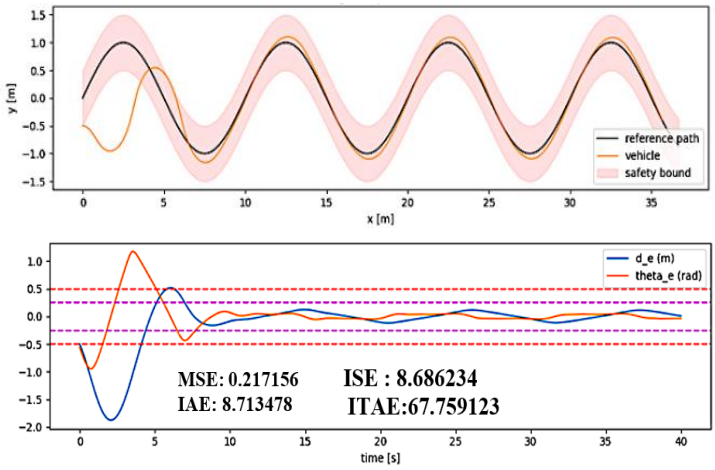


Figure 6. Path tracking in wheeled path following

4.3.2. Disturbance rejection

The next test involves applying an external disturbance to the system to evaluate the controller's ability to maintain stability. Figure 7 shows the response of the Furuta pendulum system to two impulse disturbances applied at approximately t = 5 s and t = 14 s (marked by the pink area). Figure 7(a) presents the reponse of the rotary arm angle θ, showing a sudden deviation immediately after each disturbance followed by a gradual convergence back to the reference trajectory. Quantitatively, system performance during disturbances can be measured by the MSE value. For the first disturbance (t=5.0–6.0 s), the MSE was recorded at 0.116251 with a recovery time to the reference trajectory of approximately 7.07 seconds. For the second disturbance (t=14.0–15.0 s), the MSE was slightly higher at 0.762733, indicating a larger deviation before the system returned to stability. This difference indicates that even though the second disturbance was more severe, the controller still managed to direct the system back to the reference trajectory.

In addition to ensuring that the system can restore stability when a disturbance is applied, safety aspects of other state variables must also be considered is presented in Figure 7(b). Based on the state variable response graph, it appears that $\dot{\theta}$ and $\dot{\alpha}$ move within the safety bound, while α moves outside the safety bound. Although the value is not significantly greater than the safety bound, safety aspects still need to be considered.

The test was also conducted on wheeled path following. The graph clearly shows two shaded areas (grey) at x=15–16 m and 20–21 m (disturbance), shown in Figure 8. The wheeled path following in Figure 8(a)

shows a sinusoidal reference path, where a significant deviation from the reference can be observed during each disturbance region before the controller successfully drives the vehicle back within the safety bounds. Figure 8(b) presents the corresponding lateral tracking error and heading angle over time, showing pronounced peaks during disturbance injection followed by a gradual attenuation as the system recovers.

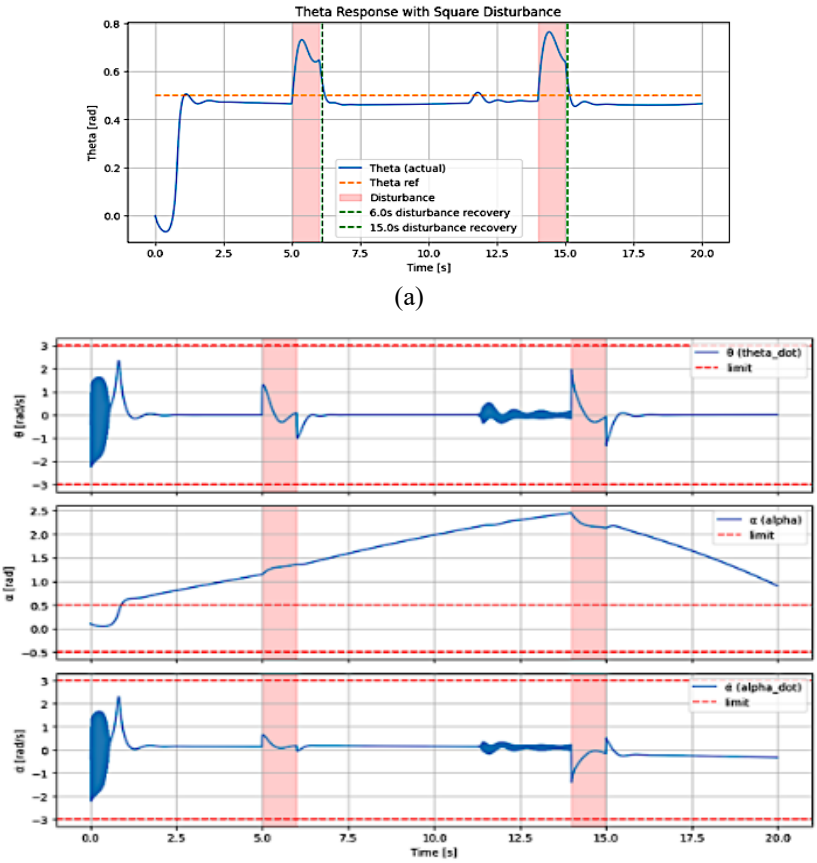


Figure 7. Response to disturbance (a) theta variable and (b) other variable state of Furuta pendulum

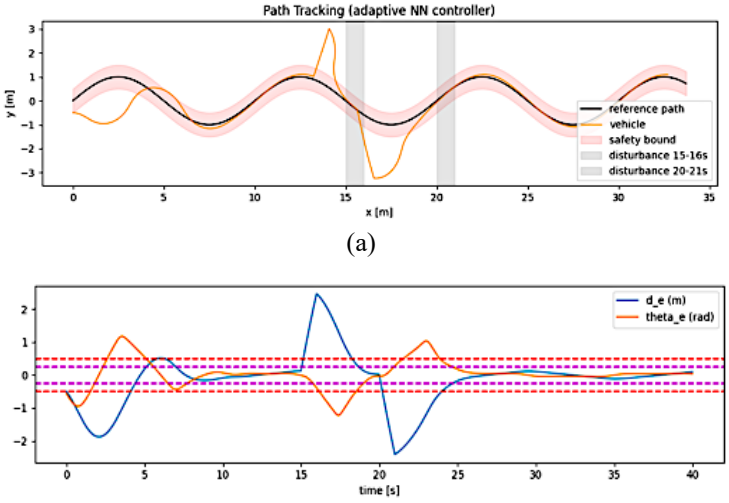


Figure 8. Response to disturbance wheeled path following (a) trajectory following and (b) corresponding lateral tracking error and heading angle

4.4. Enlargement RoA

The robustness test results through two main scenarios—namely online adaptive tracking with feedback-based learning and disturbance rejection—show a significant increase in the RoA compared to the initial RoA obtained only from the results of determining the Lyapunov function based on a static model is presented in Figure 9. As can be seen in Table 2, in the Furuta pendulum system, the RoA in the [θ–θ̇] plane is presented in Figure 9(a) increased from 89.04% to 101.14%, while in the [α–α̇] plane is presented in Figure 9(b) increased from 80.28% to 83.79%, with a constant value of c=5.02 indicating an expansion of the stability region against disturbances. Meanwhile, for the wheeled path following system is presented in Figure 9(c), the RoA increased from 85.28% to 101.69% with c=3.02. An RoA value exceeding 100% indicates that with the feedback learning-based adaptive control, the system can achieve stability in areas previously outside the static RoA domain.

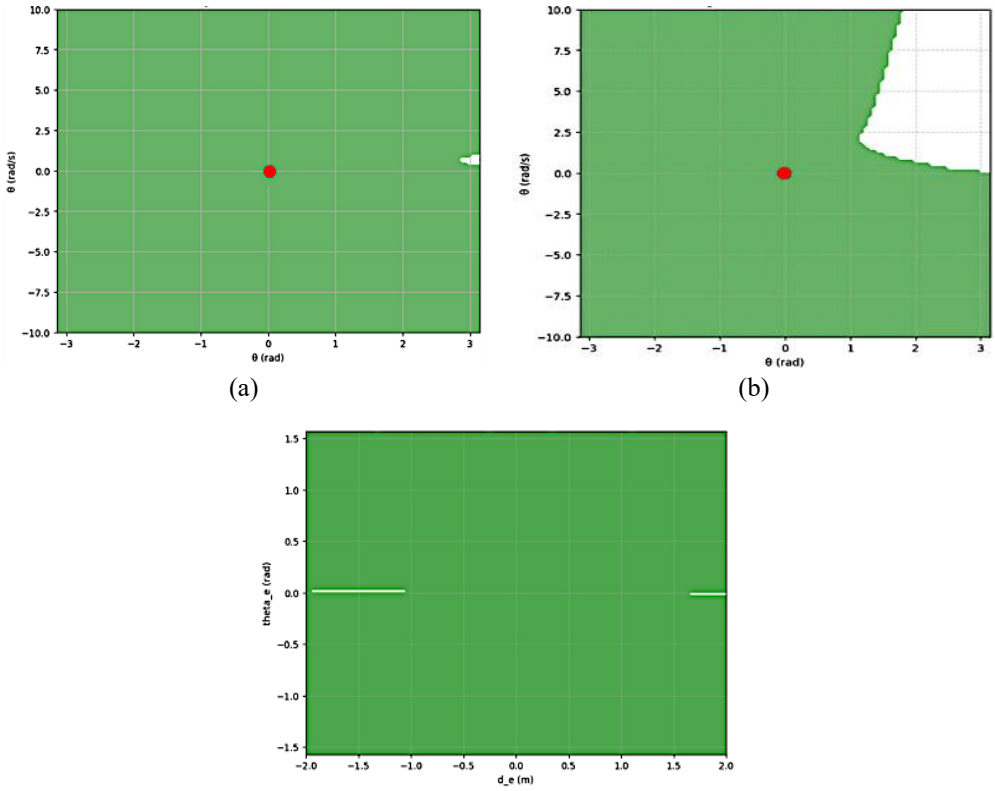


Figure 9. Enlargement RoA of Furuta pendulum (a) [θ–$\dot{\theta}$], (b) [$\alpha - \dot{\alpha}$] plane; and (c) RoA of wheeled path following

Table 2. Quantitative comparison of RoA

	LQR controller (%)	Initial DNN Lyapunov controller (%)	Adaptive DNN Lyapunov - after robustness training (%)
Furuta pendulum	42.26; 37.6	89.04; 80.28	101.14; 83.79
Wheeled path	81.47	85.28	101.69

5. CONCLUSION

This research successfully proposes a DNN-based adaptive Lyapunov-based control approach for underactuated nonlinear systems, specifically the Furuta pendulum and wheeled path following. The proposed framework jointly learns a Lyapunov function V(x) and an adaptive control law u(x), starting from LQR-initialized gains and iteratively tuning them to guarantee $V(x) > 0$ and $\dot{V}(x) < 0$, across the operating domain. As summarized in Table 2, the RoA was significantly expanded compared to the baseline LQR solution-with the Furuta pendulum achieving over 100% coverage of the training domain in [θ–$\dot{\theta}$] and the wheeled path-following system reaching 85.45% of the safety-constrained domain.

Through online adaptation and dynamics-guided feedback, the controller demonstrated strong robustness against disturbances and improved trajectory tracking. The control inputs remain smooth and within saturation limits, indicating energy-efficient actuation and practical implementability. These results highlight the effectiveness of combining Lyapunov stability theory with deep learning to achieve both theoretical guarantees and enhanced performance in nonlinear underactuated systems. Future work will focus on investigating real-time hardware implementation for experimental validation.

ACKNOWLEDGMENTS

This section should acknowledge individuals who provided personal assistance to the work but do not meet the criteria for authorship, detailing their contributions. It is imperative to obtain consent from all individuals listed in the acknowledgments.

FUNDING INFORMATION

This research received no external funding and was conducted as part of the corresponding author's ongoing research activities.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Triya Haiyunnisa		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Jony Winaryo Wibowo	✓									✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

There are no conflicts of interest

INFORMED CONSENT

This research did not involve human subjects or identifiable personal data.

ETHICAL APPROVAL

This research did not involve human participants or animals; therefore, ethical approval was not required.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] G. Gao, L. Xu, T. Huang, X. Zhao, and L. Huang, "Reduced-order observer-based LQR controller design for rotary inverted pendulum," *CMES - Computer Modeling in Engineering and Sciences*, vol. 140, no. 1, pp. 305–323, 2024, doi: 10.32604/cmes.2024.047899.

[2] O. Saleem, "An enhanced adaptive-LQR procedure for under-actuated systems using relative-rate feedback to dynamically reconfigure the state-weighting-factors," *JVC/Journal of Vibration and Control*, vol. 29, no. 9–10, pp. 2316–2331, 2023, doi: 10.1177/10775463221078654.

[3] S. D. A. Sanjeewa and M. Parnichkun, "Control of rotary double inverted pendulum system using LQR sliding surface based sliding mode controller," *Journal of Control and Decision*, vol. 9, no. 1, pp. 89–101, 2022, doi: 10.1080/23307706.2021.1914758.

[4] S. Singh and A. Swarup, "Control of rotary double inverted pendulum using sliding mode controller," in *2021 International Conference on Intelligent Technologies, CONIT 2021*, 2021, vol. 9, no. 1. doi: 10.1109/CONIT51480.2021.9498525.

[5] W. D. Compton, I. D. J. Rodriguez, N. Csomay-Shanklin, Y. Yue, and A. D. Ames, "Constructive nonlinear control of underactuated systems via zero dynamics policies," *Proceedings of the IEEE Conference on Decision and Control*. pp. 8350–8357, 2024. doi: 10.1109/CDC56724.2024.10886411.
[6] A. Shahzad and A. Altalbe, "Computing of LQR technique for nonlinear system using local approximation," *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 853–871, 2023, doi: 10.32604/csse.2023.035575.
[7] D. Perdukova, P. Fedor, M. Sobek, and J. Bacik, "A simple linearization method of nonlinear systems based on fuzzy logic," *IEEE Access*, vol. 12, pp. 165441–165457, 2024, doi: 10.1109/ACCESS.2024.3493251.
[8] Y. Chang and X. Zhang, "Backstepping-based adaptive fuzzy tracking control for pure-feedback nonlinear multi-agent systems," *International Journal of Systems Science*, vol. 55, no. 8, 2024.
[9] X. Gong, W. Fu, X. Bian, and J. Fei, "Adaptive backstepping terminal sliding mode control of nonlinear system using fuzzy neural structure," *Mathematics*, vol. 11, no. 5, p. 1094, 2023, doi: 10.3390/math11051094.
[10] W. Ben Hassen, T. Abdelkrim, N. Abdelkrim, and A. Tellili, "Control of an uncertain nonlinear system," in *2023 IEEE International Workshop on Mechatronics Systems Supervision,* 2023. doi: 10.1109/IW_MSS59200.2023.10369415.
[11] C. Stiti *et al.*, "Lyapunov-based neural network model predictive control using metaheuristic optimization approach," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-69365-9.
[12] S. Akbari, C. F. Nino, O. S. Patil, and W. E. Dixon, "Lyapunov-based deep neural networks for adaptive control of stochastic nonlinear systems," *arXiv:2412.21095*, 2024.
[13] V. C. Nguyen, H. L. Thi, and T. L. Nguyen, "A Lyapunov-based model predictive control strategy with a disturbances compensation mechanism for dual-arm manipulators," *European Journal of Control*, vol. 75, p. 100913, 2024, doi: 10.1016/j.ejcon.2023.100913.
[14] S. Y. Li, L. M. Tam, H. K. Chen, and C. S. Chen, "A novel-designed fuzzy logic control structure for control of distinct chaotic systems," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 10, pp. 2391–2406, 2020, doi: 10.1007/s13042-020-01125-3.
[15] A. Sargolzaei, B. C. Allen, C. D. Crane, and W. E. Dixon, "Lyapunov-based control of a nonlinear multiagent system with a time-varying input delay under false-data-injection attacks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2693–2703, 2022, doi: 10.1109/TII.2021.3106009.
[16] Y. Chow, O. Nachum, A. Faust, E. Duenez-Guzman, and M. Ghavamzadeh, "Lyapunov-based safe policy optimization for continuous control." 2019. doi: 10.48550/arXiv.1901.10031.
[17] C. Uyulan, "Efficient computation of Lyapunov functions using deep neural networks for the assessment of stability in controller design," *Preprint*, Dec. 2023, doi: 10.21203/rs.3.rs-3698604/v1.
[18] O. S. Patil, D. M. Le, E. J. Griffis, and W. E. Dixon, "Lyapunov-based deep residual neural network (ResNet) adaptive control," *arXiv:2404.07385*, Feb. 2025.
[19] R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Systems Letters*, vol. 6, no. 1, pp. 193–198, 2022, doi: 10.1109/LCSYS.2021.3055454.
[20] O. S. Patil, D. M. Le, M. L. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Systems Letters*, vol. 6, pp. 1855–1860, 2022, doi: 10.1109/LCSYS.2021.3134914.
[21] M. Barreau and N. Bastianello, "(Un)supervised learning of maximal Lyapunov functions," *arXiv:2408.17246v2*, May 2025.
[22] L. Grune, "Computing Lyapunov functions using deep neural networks," *Journal of Computational Dynamics*, vol. 8, no. 2, pp. 131–152, 2021, doi: 10.3934/jcd.2021006.
[23] Q. Inc., "Rotary pendulum workbook," Markham, Ontario, Canada, 2011.
[24] "Line follower sensor V1.0 datasheet," *DFRobot*, Sep. 2015. https://www.robotshop.com/en/dfrobotshop-rover-tracked-robot-basic-kit.html
[25] H. A. Koenig, *Nonlinear systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2019. doi: 10.1201/9780203746738-3.

BIOGRAPHIES OF AUTHORS

Triya Haiyunnisa received the graduated degree in electrical engineering from Sriwijaya University, Indonesia, in 2014, specializing in control and computer systems. She then pursued the master's degree in electrical engineering at Institut Teknologi Bandung, Indonesia, in 2024, with a focus on intelligent control. Currently, she works as a first researcher at the Research Center for Smart Mechatronics, National Research and Innovation Agency (BRIN), Indonesia. She can be contacted at email: triy016@brin.go.id. Her research profile is available at https://www.researchgate.net/profile/Triya-Haiyunnisa.

Jony Winaryo Wibowo completed his bachelor (S.Si.) degree in computer science from Padjajaran University of Bandung (UNPAD), Indonesia, in 2007. He received his master of informatics engineering (M.T.) degree from Institut Teknologi Bandung (ITB), Indonesia, in 2010. He is a researcher at the Research Center for Smart Mechatronics, part of the National Research and Innovation Agency of the Republic of Indonesia. He is an active member of the Intelligent Instrumentation Research Group, focusing his research interests on technology instrumentation and control systems. He has published papers in various national and international journals, as well as in conference proceedings related to technology instrumentation and control systems conferences. He can be contacted at email: jony001@brin.go.id.