Computer Modeling in
Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.069167

ARTICLE

Radial Basis Function Neural Network Adaptive Controller for Wearable Upper-Limb Exoskeleton with Disturbance Observer

Mohammad Soleimani Amiri[1], **Sahbi Boubaker**[2,3,*], **Rizauddin Ramli**[4,*] **and Souad Kamel**[2]

[1]Department of Artificial Intelligence, Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, Melaka, 76100, Malaysia
[2]Department of Computer and Network Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, 21959, Saudi Arabia
[3]King Salman Center for Disability Research, Riyadh, 11614, Saudi Arabia
[4]Department of Mechanical and Manufacturing Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia

*Corresponding Authors: Sahbi Boubaker. Email: sboubaker@uj.edu.sa; Rizauddin Ramli. Email: rizauddin@ukm.edu.my

Received: 16 June 2025; Accepted: 20 August 2025; Published: 30 September 2025

ABSTRACT: Disability is defined as a condition that makes it difficult for a person to perform certain vital activities. In recent years, the integration of the concepts of intelligence in solving various problems for disabled persons has become more frequent. However, controlling an exoskeleton for rehabilitation presents challenges due to their nonlinear characteristics and external disturbances caused by the structure itself or the patient wearing the exoskeleton. To remedy these problems, this paper presents a novel adaptive control strategy for upper-limb rehabilitation exoskeletons, addressing the challenges of nonlinear dynamics and external disturbances. The proposed controller integrated a Radial Basis Function Neural Network (RBFNN) with a disturbance observer and employed a high-dimensional integral Lyapunov function to guarantee system stability and trajectory tracking performance. In the control system, the role of the RBFNN was to estimate uncertain signals in the dynamic model, while the disturbance observer tackled external disturbances during trajectory tracking. Artificially created scenarios for Human-Robot interactive experiments and periodically repeated reference trajectory experiments validated the controller's performance, demonstrating efficient tracking. The proposed controller is found to achieve superior tracking accuracy with Root-Mean-Squared (RMS) errors of 0.022–0.026 rad for all joints, outperforming conventional Proportional-Integral-Derivative (PID) by 73% and Neural-Fuzzy Adaptive Control (NFAC) by 389.47% lower error. These results suggested that the RBFNN adaptive controller, coupled with disturbance compensation, could serve as an effective rehabilitation tool for upper-limb exoskeletons. These results demonstrate the superiority of the proposed method in enhancing rehabilitation accuracy and robustness, offering a promising solution for the control of upper-limb assistive devices. Based on the obtained results and due to their high robustness, the proposed control schemes can be extended to other motor disabilities, including lower limb exoskeletons.

KEYWORDS: Adaptive neural network controller; disturbance observer; upper-limb exoskeleton; rehabilitation robotics; Lyapunov stability; radial basis function network

1 Introduction

Motor function impairments frequently stem from neurological conditions such as spinal cord injuries and strokes [1–3]. In the rehabilitation context, designing appropriate exercises plays a crucial role in helping patients regain motor functions necessary for daily activities [4,5]. However, traditional rehabilitation methods heavily rely on one-on-one manual treatment provided by physiotherapists, who subjectively determine exercise difficulty levels and adaptations based on their expertise [6]. This labor-intensive approach highlights the challenges posed by the subjective nature of therapist perception [7]. Consequently, there is a growing interest in robotic-assisted devices as they have the potential to alleviate muscle atrophy in patients and reduce the physical demands placed on physiotherapists [8]. In addition, those devices can ensure a certain level of rehabilitation automation highly required for special types of disabilities.

Robot-assisted systems which involve a high level of interaction between the device and the patient, have become a significant area of interest in robotics, particularly in the context of rehabilitating patients with neurological disabilities [9–11]. Exoskeleton robots have been utilized in the implementation of rehabilitation programs to support physiotherapists and individuals recovering from post-stroke impairments.

The control strategies employed in rehabilitation robots are essential for achieving optimal performance in various rehabilitation treatments [12–14]. Zhao et al. [15] investigated a back stepping controller based on extended state observer for a hydraulic driven lower limb exoskeleton robot to enhance the functionality of load-bearing capacity and to reduce fatigue. Zhang et al. [16] proposed an estimated time-delay approach using Neural Network (NN) of a model-free proportional-derivative controller for a lower-limb exoskeleton. They validated the efficiency of their control system using a simulated model of an lower-limb exoskeleton. Yang et al. [17] proposed an NN controller for a lower-limb exoskeleton robot to conduct trajectory tracking tasks to compensate for external disturbances and unknown parameters. They validated their control system through simulation in MATLAB. Al-dujaili et al. [18] presented an active disturbance rejection control scheme for knee-joint motion control in an exoskeleton medical robot, utilizing both linear and nonlinear extended state observers for disturbance estimation. Asl et al. [19] also developed an NN feedback trajectory tracking controller for a robotic exoskeleton to compensate the nonlinear dynamics of the exoskeleton. Wu et al. [20] proposed a novel control strategy combining Sliding Mode Controller (SMC) and dynamic movement primitives for a reconfigurable upper-limb rehabilitation exoskeleton. Through experimental validation, they demonstrated that sliding mode control under a combinational reaching law outperforms traditional methods such as Proportional-Integral-Derivative (PID) and power reaching law-based sliding mode control in trajectory tracking. Alwand et al. [21] introduced a hybrid control strategy combining active disturbance rejection control and SMC to enhance the tracking performance of a lower limb exoskeleton for hip and knee rehabilitation. Simulation results demonstrated that their controller outperformed conventional approaches by achieving faster tracking, improved disturbance rejection, reduced chattering, and lower control effort. Motivated by the reliable estimation ability of the NN, this work proposed a NN estimation method to determine the unknown parameters.

Table 1 provides a comparative analysis of several existing studies that have developed controller methods for various applications. The focus of the studies, shown in Table 1, was on the control methods, the application as well as the study highlights including advantages and challenges of those studies. By examining Table 1, it can be observed that the surveyed works were relatively new, ranging from 2018 to 2024. In addition, those studies were concerned with the upper-limb and lower-limb as they were the most common motor disabilities. Challenges such as stability, robustness, uncertainties and disturbances were tackled with different levels of accuracy. In Table 1, controller methods are given by: Finite-time Fractional-order Nonsingular Fast Terminal Sliding Mode Control (FONFTSMC), Neural-Fuzzy Adaptive Controller (NFAC), Model-Free based Neural Network Control with Time-Delay Estimation (TDE-MFNNC), Repetitive Learning Control (RLC), Non-singular Fast Terminal Sliding Mode Control method based on Nonlinear Disturbance Observer (NFTSMC-NDO), Iterative Learning Control (ILC), Fractional-Order Ultra-local Model-based NN Sliding Mode Controller (FO-NNSMC), Adaptive Fuzzy Control (AFC) and Robust Adaptive Radial Basis Function Neural Network (RABFNN) have been used.

Table 1: Comparative analysis of controller methods across various applications

Ref.	Year	Controller method	Application	Highlights
[14]	2022	FONFTSMC	Upper-limb exoskeleton	FONFTSMC ensures finite-time convergence and enhances uncertainty robustness in the trajectory tracking of exoskeleton robots.
[22]	2018	NFAC	Upper-limb exoskeleton	The NFAC exhibited lower position tracking error and improved frequency response characteristics compared to conventional methods.
[16]	2018	TDE-MFNNC	Lower-limb exoskeleton	TDE-MFNNC performed efficiently in trajectory tracking and stability of the control torque.
[17]	2019	RLC & NN	Lower-limb exoskeleton	The RLC with NN controller overcame periodic and non-periodic uncertainties simultaneously.
[23]	2021	NFTSMC-NDO	Upper-limb exoskeleton	NFTSMC-NDO was employed to overcome dynamic model errors and load disturbances during the reconstruction of the modular joints.
[24]	2022	SMC & ILC	Upper-limb exoskeleton	SMC was utilized to handle bounded non-periodic disturbances, with the upper bound estimated using ILC.
[25]	2020	RABFNN	Lower-limb exoskeleton	RABFNN was designed to eliminate time-delay estimation issues arising from delays and discontinuous human-exoskeleton interaction torques.
[26]	2024	FO-NNSMC	Upper-limb exoskeleton	FO-NNSMSwas developed to overcome the model complexity, uncertainties, external disturbances and input deadzone.
[27]	2022	AFC	Lower-limb exoskeleton	AFC was utilized to eliminate the human-robot interaction force using the admittance model of the lower-limb exoskeleton.

From the literature presented in Table 1, the researchers have investigated several controller strategies to increase the tracking performance under external disturbances. Although prior controllers in Table 1 have contributed significantly to the field of rehabilitation robotics, several critical limitations persist. In terms of disturbance handling, some approaches lack explicit disturbance observers, making them less robust against unpredictable human-robot interaction forces. Others rely on the assumption of constant disturbances, which reduces their effectiveness under dynamic patient movements. Regarding stability guarantees, certain methods require prior knowledge of disturbance bounds while others adopt overly conservative designs that compromise responsiveness. Clinical adaptability is also a concern, as some controllers require manual tuning for different rehabilitation modes, and others fail to adjust to a patient's recovery progression over time. These limitations point to the need for a more adaptive, robust, and clinically scalable control strategy.

Motivated by the weakness of the existing works in Table 1, the novelty of this paper lies in the combination of an adaptive controller, Radial Basis Function Neural Network (RBFNN), and disturbance observer to enhance control performance as well as address unknown disturbances and uncertainties in a wearable assistive upper-limb exoskeleton. More explicitly, our paper presented the following contributions:

- An adaptive neural network controller was established based on a high-dimensional integral-type Lyapunov function and associated stability strategies. This design allowed the tracking error to remain within a prescribed boundary while ensuring efficient trajectory tracking.
- A RBFNN was developed to estimate the uncertain parameters of the nonlinear dynamic model. This enhanced the controller's adaptability and improved overall system performance under varying conditions.
- A disturbance observer was integrated into the control framework to compensate for unknown external disturbances. Its implementation increased the robustness of the controller and enabled accurate trajectory tracking in dynamic environments.

These contributions collectively strengthened the capabilities of the adaptive neural network control system, enabling more effective and reliable performance for upper-limb exoskeleton application.

The remainder of this article has been structured as follows: In Section 2, the structure of the upper-limb exoskeleton and the dynamic model have been given. Section 3 addresses the adaptive neural network controller development. Section 4 represents validation of the adaptive neural network controller on an upper-limb exoskeleton prototype. Section 5 concludes this paper.

2 Structure of Upper-Limb Exoskeleton System

The main purpose of upper-limb exoskeleton is to aid patients with muscle injuries in improving their motor capabilities for daily activities. In this section, we have presented a comprehensive overview of the mechanical design and dynamics of the prototype exoskeleton.

2.1 Mechanical Design and Structure

In this study, an upper-limb exoskeleton, which comprised four Degrees of Freedoms (DOFs) that represents four active joints, was considered. Specifically, the shoulder had two DOFs for abduction/adduction and flexion/extension. The elbow had one DOF for flexion/extension, and the wrist had one DOF for ulnar/radial deviation. Fig. 1 shows the Geometric representation of upper-limb exoskeleton.

To simplify notation, the abbreviations SP, SR, EP, and WY have been used to represent shoulder pitch, shoulder roll, elbow pitch, and wrist yaw, respectively.

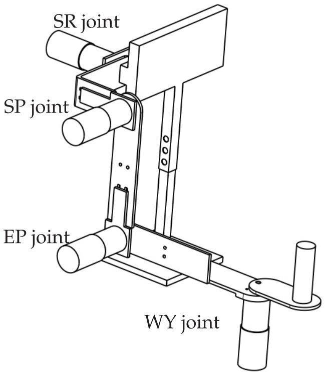


Figure 1: Geometric representation of upper-limb exoskeleton

2.2 Dynamic Model of Upper-Limb Exoskeleton

In the present study, the upper-limb exoskeleton dynamic model is considered as follows:

$$\mathcal{M}(q)\ddot{q} + C(q,\dot{q}) + \mathcal{G}(q) + \mathcal{F}_{\text{ext}}(t) = \tau_c + \tau_h \tag{1}$$

where $\mathcal{M}(q) \in \mathbb{R}^{n\times n}$ represents the mass-inertia matrix, and n is the number of joints. The vector $q \in \mathbb{R}^n$ denotes the joint coordinates. $C(q,\dot{q}) \in \mathbb{R}^n$ represents the Coriolis and centrifugal force effects. The vector $\mathcal{G}(q) \in \mathbb{R}^n$ accounts for gravitational forces. $\mathcal{F}_{\text{ext}}(t) \in \mathbb{R}^n$ denotes external disturbances. $\tau_c \in \mathbb{R}^n$ is the applied control torque, and τ_h represents the interaction forces between the human and the exoskeleton. The inertia matrix $\mathcal{M}(q)$, which captures the distribution of mass and rotational inertia of the links with respect to the joints is represented as follows:

$$\mathcal{M}(q) = \begin{bmatrix} \mathcal{M}_{11}(q) & \mathcal{M}_{12}(q) & \cdots & \mathcal{M}_{1n}(q) \\ \mathcal{M}_{21}(q) & \mathcal{M}_{22}(q) & \cdots & \mathcal{M}_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_{n1}(q) & \mathcal{M}_{n2}(q) & \cdots & \mathcal{M}_{nn}(q) \end{bmatrix} \tag{2}$$

The centripetal and Coriolis matrix $C(q,\dot{q})$ is represented as follows:

$$C(q,\dot{q}) = \begin{bmatrix} C_{11}(q) & C_{12}(q) & \cdots & C_{1n}(q) \\ C_{21}(q) & C_{22}(q) & \cdots & C_{2n}(q) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1}(q) & C_{n2}(q) & \cdots & C_{nn}(q) \end{bmatrix} \tag{3}$$

where C_{ij} is the element of $C(q,\dot{q})$ given as follows:

$$C_{ij} = \sum_{k=1}^{n} \frac{1}{2}\left(\frac{\partial \mathcal{M}_{ij}}{\partial q_k} + \frac{\partial \mathcal{M}_{ik}}{\partial q_j} - \frac{\partial \mathcal{M}_{jk}}{\partial q_i}\right)\dot{q}_k \tag{4}$$

The gravitational forces vector $\mathcal{G}(q)$ is defined as:

$$\mathcal{G}(q) = \begin{bmatrix} \mathcal{G}_1(q) \\ \mathcal{G}_2(q) \\ \vdots \\ \mathcal{G}_n(q) \end{bmatrix} \tag{5}$$

Each element $\mathcal{G}_i(q)$ represents the gravitational forces acting on the i-th joint due to the weight of the links distal to and including link i. It is computed from the potential energy of the system and typically depends on the configuration q of the manipulator.

To obtain the state equations, we define the state vector as $x = \begin{bmatrix} x_1 x_2 \end{bmatrix}^T$, where

$$\begin{cases} x_1 = q \\ x_2 = \dot{x}_1 = \dot{q} \end{cases} \tag{6}$$

If we substitute Eq. (6) into Eq. (1), we obtain the state equation, given as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \mathcal{M}^{-1}(q)[-\mathcal{C}(q,\dot{q})x_2 - \mathcal{G}(q) - \mathcal{F}_{\text{ext}}(t) + \tau_c + \tau_h] \end{cases} \tag{7}$$

The mass-inertia matrix, denoted as $\mathcal{M}(q)$, can be break down into two parts: the known matrix $\mathcal{M}_k(q)$ and the unknown matrix $\Delta_{\mathcal{M}}$, as shown below:

$$\mathcal{M}(q) = \mathcal{M}_k(q) + \Delta_{\mathcal{M}} \tag{8}$$

where $\Delta_{\mathcal{M}} \in \mathbb{R}^{n\times n}$ is the inertia matrix that contains unknown parameters, while $\mathcal{M}_k(q)$ represents a diagonal matrix with non-zero elements ($\mathcal{M}_{kii}(q) \neq 0$).

By substituting Eq. (7) into Eq. (8), we obtain:

$$\mathcal{M}_k(q)\dot{x}_2 = \mathcal{H}(x) + \mathcal{Q} + \tau_c \tag{9}$$

In Eq. (9), $\mathcal{Q} = [(I + \mathcal{M}_k(q)\Delta_{\mathcal{M}}^{-1})(\tau_h - \mathcal{F}_{\text{ext}}(t)) + \mathcal{M}_k(q)\Delta_{\mathcal{M}}^{-1}\tau_c] \in \mathbb{R}^n$ represents disturbance elements, while $\mathcal{H}(x) = [(I + \mathcal{M}_k(q)\Delta_{\mathcal{M}}^{-1})(-\mathcal{G}(q) - \mathcal{C}(q,\dot{q})x_2)] \in \mathbb{R}^n$ denotes unknown and uncertain parameters. It highlights the presence of both disturbance elements and unknown and uncertain parameters within the system dynamics.

Lemma 1: Consider a continuous and differentiable function $\Psi(t)$ [28]:

$$\gamma_1 \leq \|\Psi(t)\| \leq \gamma_2, \quad \forall t \in [t_0, t_1] \tag{10}$$

where γ_1 and γ_2 are positive constant values. Then, its derivative $\dot{\Psi}(t)$ is bounded.

Assumption 1: The external force $\mathcal{F}_{\text{ext}}(t) \in \mathbb{R}^n$ is limited by a positive constant δ as described in [17]. Therefore, we have:

$$\|\mathcal{F}_{\text{ext}}(t)\| < \delta \tag{11}$$

where δ represents a positive constant.

Assumption 2: We assume that $\mathcal{Q} : \mathbb{R}^+ \longrightarrow \mathbb{R}^n$ and $\dot{\mathcal{Q}}$ are symmetric matrices and bounded:

$$\|\mathcal{Q}\| \leq \lambda, \quad \|\dot{\mathcal{Q}}\| \leq \mu, \quad \forall t \in \mathbb{R}^+ \tag{12}$$

where λ and μ are positive constant values.

Remark 1: Referring to Eq. (9), $\mathcal{Q}$ is expressed as follows:

$$\mathcal{Q} = (I + \mathcal{M}_k^{-1}(q))\tau_h - (I + \mathcal{M}_k^{-1}(q))\mathcal{F}_{\text{ext}}(t) + \mathcal{M}_k(q)\Delta_{\mathcal{M}}^{-1}\tau_c \tag{13}$$

In many robotic systems, the actuator inputs are subject to saturation constraints, which impose a bounded condition on the control system output τ_c. Consequently, $\mathcal{M}_k(q)\Delta_{\mathcal{M}}^{-1}\tau_c$ remains bounded. Similarly, in the term $(I + \mathcal{M}_k^{-1}(q))\mathcal{F}_{\text{ext}}(t)$, where $\mathcal{F}_{\text{ext}}(t)$ represents external forces, it is bounded based on Assumption 1. Therefore, $(I + \mathcal{M}_k^{-1}(q))\mathcal{F}_{\text{ext}}(t)$ is also bounded. The variable τ_h represents the forces between the human and the exoskeleton, which is bounded according to [29]. Hence, $(I + \mathcal{M}_k^{-1}(q))\tau_h$ is bounded as well. Consequently, the variable $\mathcal{Q}$ satisfies Assumption 2 by being bounded.

3 Adaptive Neural Network Controller and Stability Analysis

This study presents an adaptive neural network controller using a high-dimensional integral-type Lyapunov function for a wearable assistive upper-limb exoskeleton. The objective of the adaptive controller is the integration between Lyapunov function, RBFNN estimator, and disturbance observer for increasing robustness and consider unknown disturbances and uncertainties. This section presents the development of the adaptive neural network controller along with its stability analysis. Fig. 2 shows the logic diagram of the proposed adaptive neural network controller.

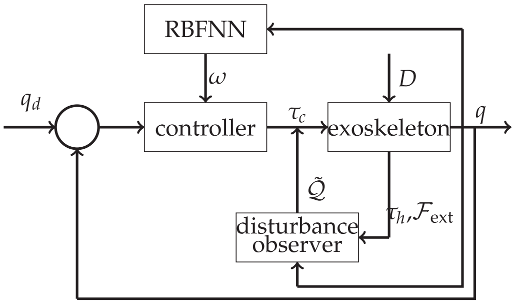


Figure 2: The logic diagram of the controller

The tracking error is defined as:

$$e = q - q_d \tag{14}$$

where q_d represents the desired trajectory and q is the actual trajectory. The tracking error z of the control system is given by:

$$z = \Lambda e + \dot{e} \tag{15}$$

Here, $\Lambda = \text{diag}\begin{bmatrix} \lambda_1, & \lambda_2, & \ldots, & \lambda_3 \end{bmatrix}$ is a positive definite symmetric matrix. According to Eq. (15), several linear differential functions can be defined to ensure that e converges when $z = 0$. Consequently, $\dot{e} \to 0$ as $t \to \infty$. By differentiating Eq. (15), we obtain:

$$\dot{z} = \Lambda\dot{e} + \ddot{e} = \Lambda\dot{e} + \ddot{q} - \ddot{q}_d \tag{16}$$

By substituting Eq. (9) into Eq. (16), the following Eq. (17) is obtained:

$$\dot{z} = M_k^{-1}(\mathcal{H}(x) + \mathcal{Q} + \tau_c) + v \tag{17}$$

where $v = \begin{bmatrix} v_1, & v_2, & \ldots, & v_n \end{bmatrix}^T$ and $z = \begin{bmatrix} z_1, & z_2, & \ldots, & z_n \end{bmatrix}^T$. The elements of v are defined as:

$$v_i = \lambda_i\dot{e}_i - \ddot{q}_d \tag{18}$$

where e_i is the elements of tracking error, $e_i(i = 1, 2, \ldots, n)$. An efficient controller can ensure actual trajectory tracks the desired trajectory and guarantees system stability in the presence of unknown disturbances. To achieve this, the controller's output, τ_c, is designed to converge to zero, i.e., $z \longrightarrow 0$ [27,30].

3.1 Radial Basis Function Neural Network Estimator

In this paper, the RBFNN estimator has been used to estimate the uncertainties parameters. The RBFNN is one type of artificial NN that uses radial basis functions as activation functions which is particularly effective for function approximation, pattern recognition, and nonlinear system estimation. When applied as an estimator, the RBFNN is used to model and predict outputs from given inputs, especially when the system being modeled is complex or nonlinear. In Eq. (9), $\mathcal{H} \in \mathbb{R}^n$ represents an uncertainty term and the RBFNN estimator has been utilized to estimate the uncertain term [31].

Lemma 2: The universal approximation theorem states that any continuous function can be approximated with arbitrary precision using a linearly parameterized estimator [32]. Mathematically, it can be represented as:

$$F(\Gamma) = \psi^T v(\Gamma) + \varepsilon \tag{19}$$

where $F(\Gamma) : \mathbb{R}^n \longrightarrow \mathbb{R}$ is the continuous function, $\Gamma \in \mathbb{R}^m$ represents the input vector, $\psi \in \mathbb{R}^l$ denotes the adjustable weight vector, $v(\Gamma)$ represents a continuous function, and $\varepsilon(\Gamma)$ is a bounded estimation error.

The bounded estimation error $\varepsilon(\Gamma)$ in Eq. (19) satisfies the condition:

$$|\varepsilon(\Gamma)| \leq \bar{\varepsilon}, \quad \forall \Gamma \in \Omega_\Gamma \tag{20}$$

Here, $\bar{\varepsilon}$ represents a positive constant number ($\bar{\varepsilon} > 0$), and Ω_Γ denotes the domain of the input vector Γ. This inequality ensures the estimation error remains within a certain bound for all valid values of Γ. By considering ω^* as the optimal constant weight, Eq. (19) can be rewritten as:

$$F(\Gamma) = \omega^{*T}\Phi(\Gamma) + \rho \tag{21}$$

Here, $\Phi(\Gamma) = \text{diag}\begin{bmatrix} \Phi_1(\Gamma), & \Phi_2(\Gamma), & \ldots, & \Phi_n(\Gamma) \end{bmatrix}^T$ represents the Gaussian function, which is defined as:

$$\Phi_i = \exp\left[-\frac{(\Gamma - c_i)^T(\Gamma - c_i)}{\sigma_i^2}\right], \quad i = 1, 2, \ldots, n \tag{22}$$

In the above equation, $c_i = \begin{bmatrix} c_1, & c_2, & \ldots, & c_i \end{bmatrix}$ represents the receptive field center, and σ_i is the Gaussian function width. Additionally, there exists a positive constant β such that $|\Phi(\Gamma)| \leq \beta$ with $\beta > 0$. ρ represents the smallest possible estimation error of the RBFNN. From the estimation capability of RBFNN, it is known that the estimation error is bounded by the constant ρ^*, i.e., $|\rho| \leq \rho^*$, where ρ^* is a positive constant. Therefore, the estimation of the uncertain term $\mathcal{H}(x)$ in Eq. (9) using the RBFNN can be expressed as:

$$\mathcal{H}(x) = -\omega^{*T}\Phi(x) - \varepsilon \tag{23}$$

Here, $\omega^{*^T} = \text{diag}[\omega_i]$, $i = 1, 2, \ldots, n$ is the RBFNN optimal weight, $\mathcal{H}$ is the hidden-layer output, and ε is a bounded estimated error. We can infer from Eq. (9) that $\mathcal{H}(x)$ as represents an unknown uncertain term that can be estimated by the RBFNN.

Since the exoskeleton has 4 DOFs, the number of neurons in the output layer is set to 4. The number of input variables is also chosen as 4, corresponding to the system's states. The hidden layer is configured with 128 neurons. It is worth noting that, due to the continuity of $\mathcal{H}$ and based on Lemma 1 and Eq. (19), the function is bounded. However, it does not depend on the bounded estimation error ε. Fig. 3 architecture of the RBFNN to estimate $\mathcal{H}$.

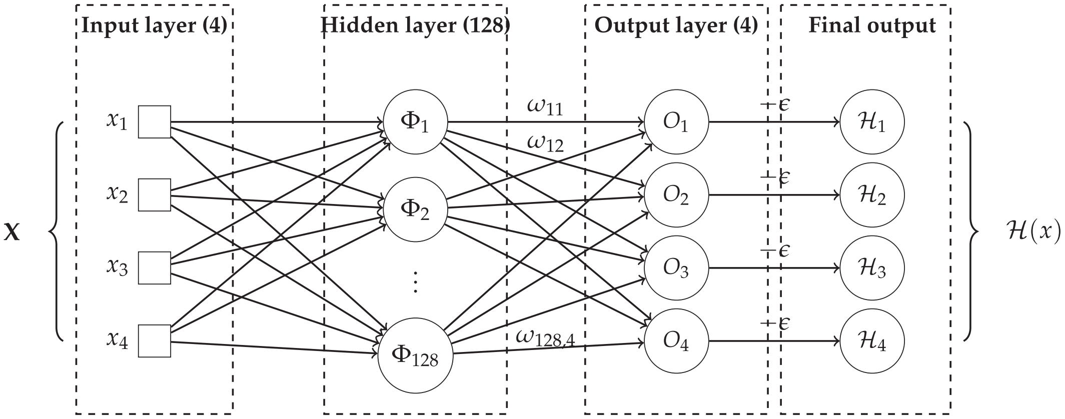


Figure 3: Architecture of the RBFNN with 4 inputs, 128 hidden neurons (Φ), and 4 outputs (H), where $O = -\omega^{*T}\Phi(x)$

3.2 Controller Development with Disturbance Observer

The stability and robustness of the adaptive neural network controller are analyzed by the Lyapunov functions. A candidate Lyapunov function is defined as follows [33]:

$$V_1 = z^T S z \tag{24}$$

where S is given as follows:

$$S = diag[\int_0^1 \mathcal{M}_{k11}(\bar{x}, \theta_{z_1} + v_1)\alpha_{11} d\theta \quad \dots \quad \int_0^1 \mathcal{M}_{knn}(\bar{x}, \theta_{z_n} + v_n)\alpha_{nn} d\theta] \tag{25}$$

For the system stability proof, we define the equation:

$$A_\theta = \int_0^1 \theta A_\alpha d\theta \tag{26}$$

Here, $A_\alpha = \text{diag}\begin{bmatrix} A_{\alpha_{11}}, & A_{\alpha_{22}}, & \dots, & A_{\alpha_{nn}} \end{bmatrix}$, and we make the assumption:

$$A_{\alpha_{ii}}(\bar{x}, \theta_{z_1} + v_i) = M_{kii}(\bar{x}, \theta_{z_1} + v_i)\alpha_{ii} \tag{27}$$

Here, $\alpha_{ii} = \text{diag}\begin{bmatrix} \alpha_{11}, & \alpha_{22}, & \dots, & \alpha_{nn} \end{bmatrix}$ is a positive definite diagonal matrix of size $n \times n$, and $\bar{x} = x_1$. For simplicity, we assume that $\alpha_{11} = \alpha_{22} = \dots = \alpha_{nn}$. Additionally, $v = \dot{q}_d - \zeta$ is defined, where $\zeta = \begin{bmatrix} \zeta_1 & \dots & \zeta_n \end{bmatrix}^T \in \mathbb{R}^n$ with $\zeta = \lambda_i e_i$ and λ_i is a constant eigenvalue. Here, $\dot{q}_d$ and $\theta \in [0, 1]$ represent desired angular velocity and an independent scalar of z, x, and v, respectively [34].

According to Eq. (27), based on the eigenvalues maximum and minimum of A_α ($\lambda_{\max} A_\alpha$ and $\lambda_{\min} A_\alpha$), we have:

$$0 \le \lambda_{\min} A_\alpha z^T z \le z^T A_\alpha z \le \lambda_{\max} A_\alpha z^T z \tag{28}$$

This inequality shows the bounds on the quadratic form $z^T A_\alpha z$ in terms of the eigenvalues of A_α. Substituting Eq. (26) into Eq. (28) yields:

$$0 \le z^T \left(\int_0^1 \theta A_\alpha d\theta \right) z \le \left(\int_0^1 \theta \lambda_{\max}(A_\alpha) d\theta \right) z^T z \tag{29}$$

From this inequality, we can conclude that $V_1 \ge 0$. According to Eqs. (24) and (27), the candidate Lyapunov function can be rewritten as follows:

$$V_1 = \sum_{i=1}^{m} z_i^2 \int_0^1 \theta A_{\alpha_{ii}}(\bar{x}, \theta_{z_1} + v_i) d\theta \tag{30}$$

This representation shows that the candidate Lyapunov function V_1 is a quadratic form of the tracking error z with integration involving the terms $A_{\alpha_{ii}}$. The first derivative of Eq. (24) is given as follows:

$$\dot{V}_1 = 2z^T A\theta \dot{z} + z^T \left(\frac{\partial A_\theta}{\partial z} \dot{z} \right) z + z^T \left(\frac{\partial A_\theta}{\partial x} \dot{x} \right) z + z^T \left(\frac{\partial A_\theta}{\partial v} \dot{v} \right) z \tag{31}$$

Here,

$$\frac{\partial A_\theta}{\partial z} \dot{z} = \operatorname{diag}\left[\int_0^1 \theta \frac{\partial A_{\alpha_{ii}}}{\partial z_i} \dot{z}_i d\theta \right] \tag{32}$$

$$\frac{\partial A_\theta}{\partial x} \dot{x} = \operatorname{diag}\left[\int_0^1 \theta \frac{\partial A_{\alpha_{ii}}}{\partial x_i} \dot{x}_i d\theta \right] \tag{33}$$

$$\frac{\partial A_\theta}{\partial v} \dot{v} = \operatorname{diag}\left[\int_0^1 \theta \frac{\partial A_{\alpha_{ii}}}{\partial v_i} \dot{v}_i d\theta \right] \tag{34}$$

These equations represent the partial derivatives of A_θ with respect to z, x, and v, respectively. By assuming $u = \theta z$, we can determine,

$$\frac{\partial A_\alpha}{\partial z} = \frac{\partial A_\alpha}{\partial u} \frac{\partial u}{\partial z} = \theta \frac{\partial A_\alpha}{\partial u} \tag{35}$$

$$\frac{\partial A_\alpha}{\partial \theta} = \frac{\partial A_\alpha}{\partial u} \frac{\partial u}{\partial \theta} = \frac{\partial A_\alpha}{\partial u} z \tag{36}$$

Substituting these derivatives into the expression for $\frac{\partial A_\theta}{\partial z} z$, we have:

$$\frac{\partial A_\theta}{\partial z} z = diag\left[\int_0^1 \theta \frac{\partial A_{\alpha_{ii}}}{\partial z_i} z_i d\theta \right] = \int_0^1 \theta^2 \frac{\partial A_\alpha}{\partial \theta} d\theta \tag{37}$$

This provides the expression for $\frac{\partial A_\theta}{\partial z} z$ in terms of the derivatives of A_α and the variable z. It can be further derived as follows:

$$z^T \left(\frac{\partial A_\theta}{\partial z} \dot{z} \right) z = z^T \left([\theta^2 A_\alpha]_0^1 - 2 \int_0^1 \theta A_\alpha d\theta \right) \dot{z} = z^T A_\alpha \dot{z} - 2z^T A_\theta \dot{z} \tag{38}$$

Consider that θ and v are independent scalars. Since $u = \theta z$, we can derive $\frac{\partial A_\theta}{\partial v} z = diag\left[\int_0^1 \frac{\partial A_{\alpha_{ii}}}{\partial v} z_i d\theta\right] = \int_0^1 \theta \frac{\partial A_\alpha}{\partial \theta} d\theta$ and $v = -\dot{v}$. Therefore, we have:

$$z^T \left(\frac{\partial A_\alpha}{\partial v} \dot{v}\right) z = z^T \left(-\int_0^1 \theta \frac{\partial A_\alpha}{\partial \theta} d\theta\right) v = z^T \int_0^1 A_\alpha v d\theta - z^T A_\alpha v \tag{39}$$

By substituting Eqs. (38) and (39) into Eq. (31), we have:

$$\dot{V}_1 = z^T (A_\alpha \dot{z} - A_\alpha v + \Psi) \tag{40}$$

where

$$\Psi = \left(\frac{\partial A_\theta}{\partial x} \dot{x}\right) + \int_0^1 A_\alpha v d\theta \tag{41}$$

Since $\mathcal{M}_k(q)$, α, and $\mathcal{M}_k(q)\alpha$ are symmetric matrices, Eq. (17) is substituted into Eq. (40), thus we have:

$$\dot{V}_1 = z^T \left[A_\alpha \mathcal{M}_k^{-1}(q)(\mathcal{H}(x) + \tau_c + \mathcal{Q}) + \Psi\right] \tag{42}$$

Since $\mathcal{M}_k^{-1}$ and A_α are symmetric, we have:

$$\alpha = \mathcal{M}_K^{-1}(q)\alpha\mathcal{M}_k^{-1}(q) = A_\alpha \mathcal{M}_k^{-1}(q) \tag{43}$$

Then, Eq. (42) is rewritten as follows:

$$\dot{V}_1 = z^T \left[\alpha(\mathcal{H}(x) + \tau_c + \mathcal{Q}) + \Psi\right] \tag{44}$$

We can further derive as follows:

$$\dot{V}_1 = z^T [\alpha(\mathcal{H}(x) + \tau_c + \mathcal{Q}) + \Psi_\alpha] \tag{45}$$

where

$$\Psi_\alpha = \int_0^1 \theta \left(\frac{\partial A_\alpha}{\partial x} \dot{x}\right) z d\theta + \int_0^1 A_\alpha v d\theta \tag{46}$$

By considering $A_\alpha = \mathcal{M}_k(q)\alpha$, we have $\Psi_\alpha = \alpha\Omega(x, q)$, where

$$\Omega(x, q) = \left[\int_0^1 \theta \frac{\partial \mathcal{M}_k(q)}{\partial x} + \int_0^1 \mathcal{M}_k(q) v d\theta\right] \tag{47}$$

Thus, Eq. (45) is rewritten as follows:

$$\dot{V}_1 = Z^T \alpha[\mathcal{H}(x) + \tau_c + \mathcal{Q} + \Omega(x, q)] \tag{48}$$

It is acknowledged from Eq. (23), $\mathcal{H}(x)$ is approximated by RBFNN. Therefore we can have:

$$\dot{V}_1 = z^T \alpha[-\omega^{*T}\Phi(x) + \mathcal{Q} + \tau_c + \Omega(x, q)] \tag{49}$$

From Lemma 1 and Assumption 2, it is determined that $\mathcal{Q}$ is bounded $\|\dot{\mathcal{Q}}\| \leqslant \mu$, where μ is an unknown positive constant.

An auxiliary variable D is defined to estimate system disturbance Q by calculating a disturbance observer [35], as follows:

$$D = Q - Kx_2 \tag{50}$$

where $K = K^T > 0$ represents a positive definite symmetric matrix that needs to be specified. Taking into account Eqs. (9) and (50), and assuming $M_k(q)$ is a diagonal matrix, the time derivative of D is given as by:

$$\dot{D} = \dot{Q} - K\dot{x}_2 = \dot{Q} - K\mathcal{M}_k^{-1}(q)[\mathcal{H}(x) + \tau_c + Q] \tag{51}$$

In Eq. (51), the term $H(x)$ is estimated by the RBFNN method, as described in Eq. (23). Thus we can obtain that:

$$\dot{D} = \dot{Q} - K\mathcal{M}_k^{-1}(q)[-\omega^{*T}\Phi(x) + \tau_c + Q] \tag{52}$$

To calculate the system disturbance Q, it is necessary to obtain the intermediate variable D initially. The estimation of Q is denoted as $\hat{Q}$. By utilizing Eq. (52), Assumption 2, and assuming that $\dot{\hat{Q}} = 0$, the update law for $\hat{D}$ can be expressed as follows:

$$\dot{\hat{D}} = -K\mathcal{M}_k^{-1}(q)[-\hat{\omega}^T\Phi(x) + \tau_c + \hat{Q}] \tag{53}$$

From Eq. (50), the approximation of system disturbance Q is represented as follows:

$$\hat{Q} = \hat{D} - Kx_2 \tag{54}$$

Estimation error of Q and D can be obtained as follows:

$$\tilde{Q} = Q - \hat{Q} \tag{55}$$

$$\tilde{D} = D - \hat{D} = \tilde{Q} \tag{56}$$

According to Eqs. (50) and (53), the derivative of $\tilde{D}$ and $\tilde{Q}$ with time t are given as follows:

$$\dot{\tilde{Q}} = \dot{\tilde{D}} = \dot{D} - \dot{\hat{D}} = \dot{Q} - KM_k^{-1}(q)[\tilde{\omega}^T\Phi(x) + \tilde{Q}] \tag{57}$$

where $\tilde{\omega} = \hat{\omega} - \omega^*$, in which $\hat{\omega}$ is the estimation of ω^*. From above analysis, the controller output is defined as follows:

$$\tau_c = \hat{\omega}^T\Phi(x) - \hat{Q} - \Omega(x, q) - K_l\alpha z \tag{58}$$

where $K_l = K_l^T > 0$ is a symmetric, positive definite matrix. The update law of $\hat{\omega}$ is given as follows:

$$\dot{\hat{\omega}}_i = -\Xi_i[P_i(x)\alpha_{ii}z_i + \sigma\hat{\omega}_i] \tag{59}$$

where $\Xi_i \in \mathbb{R}^{n\times n}(i = 1, 2, \ldots, n)$ represents a symmetric positive constant matrix, consisting of positive diagonal elements Ξ_i, and σ is a positive constant.

Considering a Lyapunov function candidate as follows:

$$V_2 = V_1 + \frac{1}{2}\tilde{Q}^T\tilde{Q} + \frac{1}{2}\sum_{i=1}^{n}\tilde{\omega}_i^T\Xi_i^{-1}\tilde{\omega}_i \tag{60}$$

By considering Eq. (49), the derivative of V_2 is given as follows:

$$\dot{V}_2 = z^T\alpha[-\omega^{*T}\Phi(x) + \tau_c + Q + \Omega(x,q)] + \sum_{i=1}^{n}\tilde{\omega}_i\Xi_i^{-1}\dot{\tilde{\omega}}_i + \tilde{Q}^T\dot{\tilde{Q}} \tag{61}$$

Substituting controller output in Eq. (58) into Eq. (61), we have:

$$\dot{V}_2 = z^T\alpha[\tilde{\omega}^T\Phi(x) - \tilde{Q} - K_l\alpha z] + \sum_{i=1}^{n}\tilde{\omega}_i\Xi_i^{-1}\dot{\tilde{\omega}}_i + \tilde{Q}^T\dot{\tilde{Q}} \tag{62}$$

According to Assumption 2, Eqs. (57) and (59), $\|\Phi(x)\| \le \gamma$, and the following expressions,

$$z^T\alpha\tilde{Q} \le \frac{z^T\alpha\alpha z}{2} + \frac{\tilde{Q}^T}{\tilde{Q}} \tag{63}$$

$$\tilde{Q}^T\dot{Q} \le \frac{\tilde{Q}^T\tilde{Q}}{2} + \frac{\|\dot{Q}\|}{2} \tag{64}$$

$$\sum_{i=1}^{n}\tilde{\omega}_i^T P_i(x) z_i \alpha_{ii} = z^T\tilde{\omega}^T\Phi(x) \tag{65}$$

It can be concluded that,

$$\dot{V}_2 \le -z^T\alpha(K_l - 0.5I_{n\times n})\alpha z + \frac{\mu^2}{2} - \tilde{Q}^T(KM_k^{-1}(q) - 1.5I_{n\times n})$$

$$+\frac{\sigma\|\omega^*\|^2}{2} - \frac{\sigma - \|K\mathcal{M}_k^{-1}(q)\|^2\gamma}{2}\sum_{i=1}^{n}\tilde{\omega}_i^T\tilde{\omega}_i \tag{66}$$

where the following expressions are taken into consideration:

$$\tilde{Q}^T K\mathcal{M}_k^{-1}(q)\tilde{\omega}^T\Phi(x) \le \frac{\|\tilde{Q}\|^2}{2} + \frac{\|K\mathcal{M}_k^{-1}(q)\|^2\|\Phi(x)\|^2\|\tilde{\omega}\|^2}{2} \tag{67}$$

$$-\sigma\tilde{\omega}_i\tilde{\omega}_i = -\sigma\|\tilde{\omega}_i\|^2 - \sigma\tilde{\omega}_i\tilde{\omega}_i^* \le -\frac{\sigma\|\tilde{\omega}_i\|^2}{2} + \frac{\sigma\|\tilde{\omega}_i^*\|^2}{2} \tag{68}$$

The positive definite controller parameters K, K_l, and positive constant σ are needed to be chosen such that: $K_l - 0.5I_{n\times n} \ge 0$, $KM_k^{-1}(q) - 1.5I_{n\times n} > 0$, and $\sigma - \|KM_k^{-1}(q)\|^2\gamma > 0$, we establish the following inequality:

$$\dot{V}_2 \le \kappa V_2 + L \tag{69}$$

where

$$\kappa = min\left(\begin{array}{c} 2\lambda_i(K\mathcal{M}^{-1}(q)_k - 1.5I_{n\times n}), \frac{\sigma - \|KM_k^{-1}(q)\|^2\gamma}{\lambda_{max}(\Xi_i^{-1})}, \\ \frac{\lambda_{min}(\alpha(K_l - 0.5I_{n\times n})\alpha)}{\int_0^1\lambda_{max}(A_\alpha)d\theta} \end{array}\right) \tag{70}$$

$$L = \frac{\mu^2}{2} + \frac{\sigma}{2}\|\omega^*\|^2 \tag{71}$$

$\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ represents the minimum and maximum eigenvalues of $(\cdot)$, respectively. The following inequality is established by multiplying $e^{\kappa t}$ and the integrating both sides of Eq. (69) with respect to time:

$$V_2 \leq \left(V_2(0) - \frac{L}{\kappa}\right)e^{\kappa t} + \frac{L}{\kappa} \leq V_{(0)} + \frac{L}{\kappa} \tag{72}$$

Eq. (72) shows that V_2 remains ultimately bounded as $t \to \infty$, it follows that $z, \tilde{Q}$, and $\tilde{\omega}$ are also bounded within a specific range. Hence, we can conclude that the proof is completed and the stability of this system has been proven by using the Lyapunov method [36,37].

4 Results and Discussion

In upper-limb exoskeleton control for rehabilitation, desired trajectories are very important in guiding patient movement and ensuring therapeutic effectiveness. In this context, desired trajectories may vary depending on the patient or the stage of rehabilitation of the same subject. In the case of the same subject, four case scenarios may illustrate the range of control needs. First, passive rehabilitation involves the exoskeleton following a pre-defined trajectory without patient effort, ideal for early-stage recovery or patients with limited motor function. This case-scenario is usually faced during the beginning of the rehabilitation exercises or when the patient power is low. Second, active-assisted control adapts the trajectory based on partial user input, where the exoskeleton supports and guides the limb along a desired path when the patient initiates motion but lacks full strength. Third, in resistive training, the exoskeleton imposes resistance along a target trajectory to strengthen muscles, commonly used in advanced rehabilitation stages. Lastly, task-specific trajectories involve complex, real-world movements (e.g., reaching or grasping), requiring the exoskeleton to follow trajectories that mimic daily activities, promoting neuroplasticity and functional recovery. Each scenario demands precise trajectory planning and real-time adaptability to patient condition, making control design a key challenge in upper-limb rehabilitation robotics.

To test the effectiveness of our proposed controller in trajectory tracking, four case scenarios were created artificially to imitate the above-cited cases. Without loss of generality, the generated trajectories were expected to replicate the upper-limb joint movements of the same subject but at different stages of the rehabilitation or four different subjects. The scenarios were created only for simulation purposes.

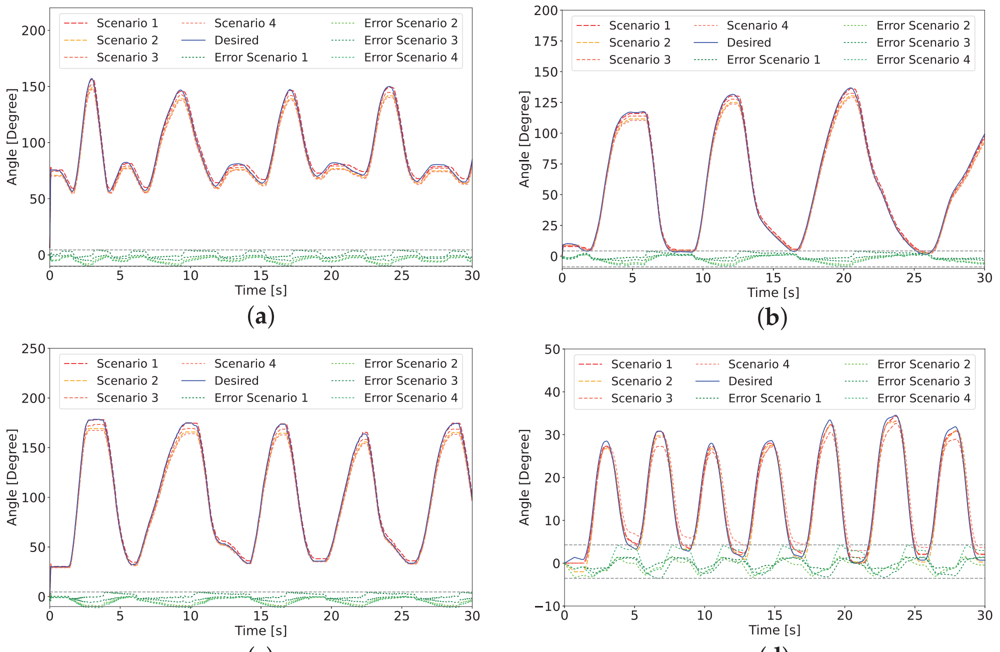
The gains of our adaptive neural network controller, namely Λ, K, K_l, α, σ, and Ξ, were proportional gains that significantly affected the system. For example, selecting larger values for the gains increased the system's convergence rate but also led to larger oscillations. Conversely, choosing smaller values for the gains resulted in slower convergence but reduced oscillation. In this study, the parameters were selected using either a trial-and-error approach or based on prior experience to achieve a balance between convergence rate, oscillation, and maintaining a limited range of tracking error. Considering system response, robustness, and tracking error, the controller parameters were determined as $\Xi = \text{diag}\begin{bmatrix} 1.5 & 1.5 & 1.5 & 1.5 \end{bmatrix}$, $\Lambda = \text{diag}\begin{bmatrix} 200 & 200 & 200 & 200 \end{bmatrix}$, $\alpha = I_{4\times4}$, and $\sigma = 0.55$. For disturbance compensation, the disturbance observer parameters were selected as $K = \text{diag}\begin{bmatrix} 4 & 4 & 4 & 4 \end{bmatrix}$ and $K_l = \begin{bmatrix} 3.2 & 3.2 & 3.2 & 3.2 \end{bmatrix}$. The choice of controller parameters significantly influenced system behavior. For example, higher gain values led to faster convergence but result in greater oscillations, whereas lower gains reduced oscillations at the cost of slower convergence. In this simulation, the parameters were chosen through a trial-and-error process.

The dynamic model of the upper-limb exoskeleton can be described as in Eq. (1). Because it is challenging to obtain the exact mathematical model of the system to simulate precise simulation model of the robot, we assumed that the following external disturbances and the interaction forces between the human and the exoskeleton are present:

$$\tau_h = \begin{bmatrix} 0.4\dot{q}_1 + \sin(2q_1) + 0.3\sin(\dot{q}_1) \\ 1.2\dot{q}_2 - 1.5\sin(q_2 + q_1) + 0.9\sin(\dot{q}_2) \\ -1.6\dot{q}_3 - 2.2\sin(q_3) + 0.2\sin(\dot{q}_3) \\ 0.7\dot{q}_4 - 1.3\sin(q_4 - q_2) + 0.4\sin(\dot{q}_4) \end{bmatrix} \tag{73}$$

$$\mathcal{F}_{\text{ext}}(t) = \begin{bmatrix} 1.5\cos(t) + \sin(2t) \\ 2\cos^2(t) + 1.8\cos(t) \\ 2.5\sin(t) - \cos(t)\sin(2t) \\ 1.2\sin(2t) + 0.5\cos(3t) \end{bmatrix} \tag{74}$$

The artificially generated trajectories q_d have been shown in Fig. 4 for SR, SP, EP, and WY. It should be recalled here that the abbreviations SP, SR, EP, and WY have been used to respectively represent shoulder pitch, shoulder roll, elbow pitch, and wrist yaw. The actual trajectories, represented as q, have been shown in Fig. 4. In what follows, the four cases were considered as scenarios in the context of the conducted simulations.

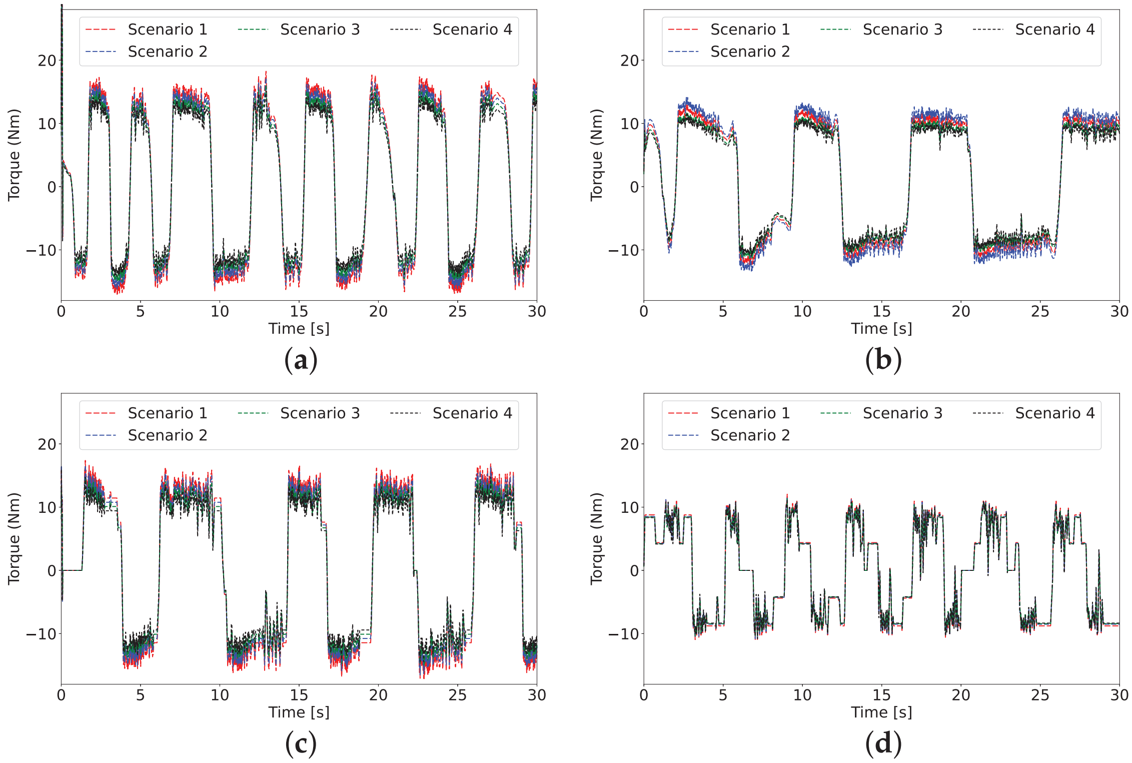


(a) (b) (c) (d)

Figure 4: Angular trajectories and tracking errors of SR, SP, EP, and WY: (**a**) SR (**b**) SP (**c**) EP (**d**) WY

The desired trajectory were generated based on our previous study in [37]. An analysis of Fig. 4 reveals that the tracking error e stayed within a defined range, confirming the effectiveness of the adaptive neural network controller. The output of the controller has been represented in Fig. 5.

Fig. 5 presented the raw output of the controller, where the observed chattering reflects the controller's effort to suppress disturbances affecting the system. Also, the controller output remained within a bounded and stable range. The controller output increased as the generated desired trajectory changed more drastically. For example, as represented in Fig. 4, the trajectory corresponded to a movement of the SR joint more frequently than SP, resulting in the controller applying a higher output to the actuators. This efficient adaptation of the control system can be observed. Furthermore, Fig. 6 illustrates the estimated disturbance of SR, SP, EP, and WY.



(a)

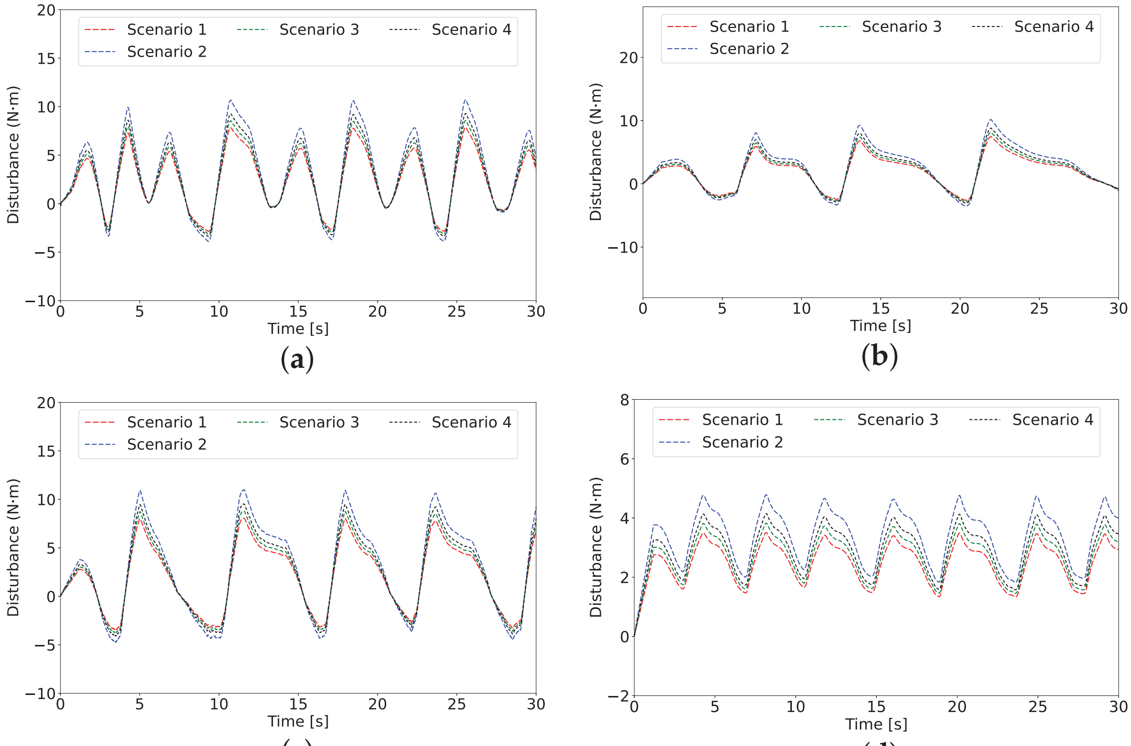
(b)

(c)

(d)

Figure 5: Controller output of SR, SP, EP, and WY: (**a**) SR (**b**) SP (**c**) EP (**d**) WY

The estimated disturbance was calculated based on the measured data from actual trajectory and the measured torque data for each joint. Frequent moves in trajectory introduce heightened disturbances to the joints, elevating angular velocity and external forces on the arm. This creates additional disturbances in the control system. analysis of Figs. 4 and 6 revealed the adaptive neural network controller adeptly observed and responded to these disturbances.



(**a**)

(**b**)

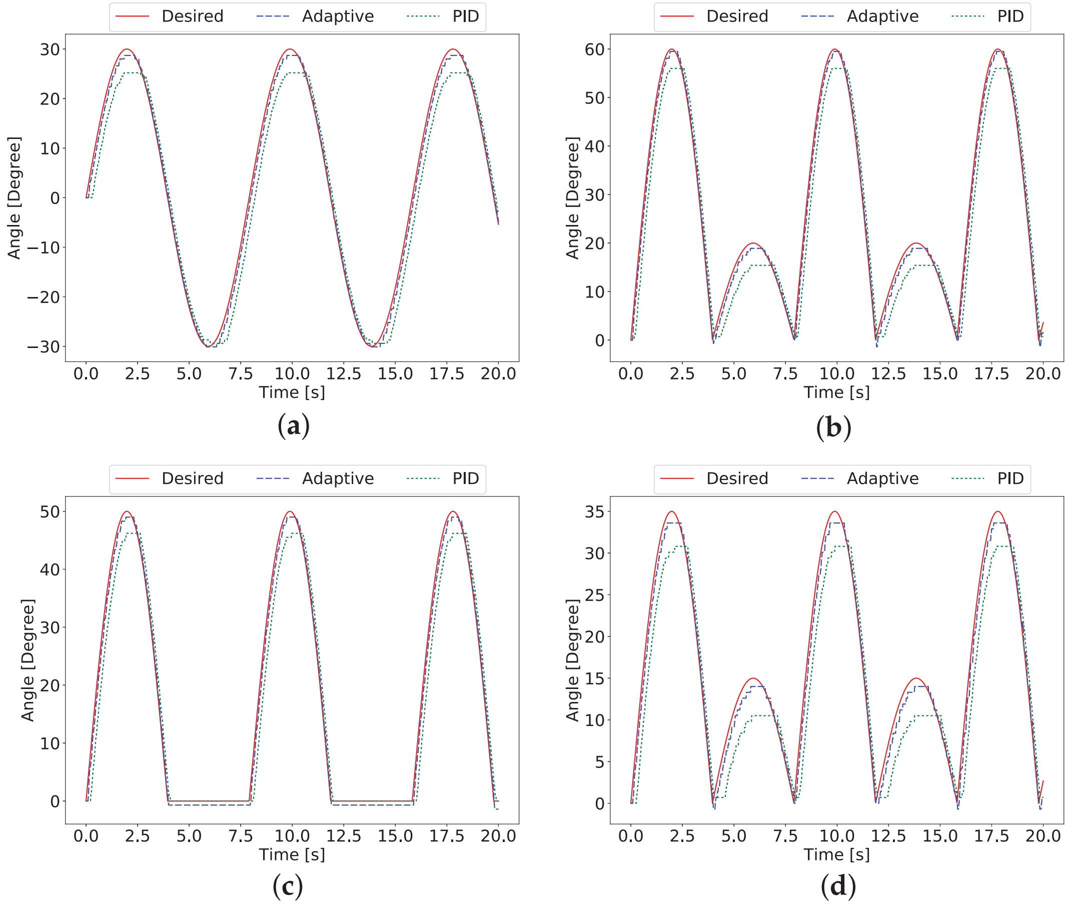
(**c**)

(**d**)

Figure 6: Estimated disturbance of SR, SP, EP, and WY: (**a**) SR (**b**) SP (**c**) EP (**d**) WY

In another experiment, a similar procedure was conducted with predefined periodically desired trajectories. The results were then compared with a conventional PID controller mentioned in [38,39]. The conventional PID controller had the following parameter values: $K_p = diag\begin{bmatrix} 3 & 3 & 3 & 3 \end{bmatrix}$, $K_i = diag\begin{bmatrix} 2.5 & 2.5 & 2.5 & 2.5 \end{bmatrix}$, and $K_d = diag\begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$. Fig. 7 illustrates the comparison of the tracking trajectory between the adaptive neural network controller and the conventional PID controller.

The Root Mean Square (RMS) of the tracking error had been determined as shown in Table 2. This table compares the propsed daptive neural network controller, PID controller from [38], NFAC from [22], as well as AFC from [27].



(**a**)

(**b**)

(**c**)

(**d**)

Figure 7: Comparison of tracking trajectory of our adaptive neural network controller and a conventional PID controller of each joint. (**a**) SR (**b**) SP (**c**) EP (**d**) WY

Table 2: RMS of the tracking error for tracking trajectories

	Joint 1	**Joint 2**	**Joint 3**	**Joint 4**
Controllers	**(SR)**	**(SP)**	**(EP)**	**(WY)**
Adaptive neural network controller	0.022	0.026	0.024	0.019
Conventional PID	0.073	0.084	0.067	0.070
NFAC (Wu et al. [22])	0.037	NA	0.027	0.022
AFC (Huang et al. [27])	0.1209	0.0992	0.1319	0.0929

The adaptive neural network controller exhibited a smaller RMS value compared to the conventional PID controller. The adaptive neural network control scheme, which incorporated a disturbance observer, established a more instinctive interaction between the human and robot than the method mentioned in [22,27]. For instance, the performance of the Adaptive Neural Network Controller for Joint 4 (WY) was more efficient by 15.79%, 389.47%, and 268.42% than the NFAC controller by [22], the AFC controller by [27], and the Conventional PID controller, respectively. Therefore, our adaptive neural network controller method enabled efficient trajectory tracking even in the presence of human interaction forces and unknown disturbances.

Our adaptive neural network controller efficiently tracked defined desired trajectory that was adopted from our previous study in [37]. The disturbance observer effectively mitigated unknown disturbances, ensuring system stability using a high-dimensional integral Lyapunov function. The neural network estimator accurately estimated uncertain parameters of the nonlinear dynamic system. In summary, our adaptive neural network controller constrained the tracking error within a specified range, demonstrating its overall effectiveness. In conclusion, our adaptive control scheme has exhibited satisfactory performance and holds significant potential in the field of upper-limb exoskeletons.

5 Conclusion

This paper presentd a novel adaptive neural network controller designed for a four DoFs wearable assistive upper-limb exoskeleton. The controller ensured system stability using a high-dimensional integral-based Lyapunov function. The RBFNN estimated uncertain parameters, while a disturbance observer handled unknown disturbances. The adaptive neural network controller aimed to accurately track the exoskeleton's desired trajectory for target joints.

Four human-interaction-based scenarios were conducted to assess the controller's effectiveness. Results revealed that the adaptive neural network controller bounded tracking errors within a specific range, surpassing the conventional PID controller in terms of efficiency and performance. The proposed controller achieved a significant improvement in RMS tracking error over several benchmark methods, reducing the error by 73% compared to the conventional PID controller, by 15.8% compared to the NFAC, and by an impressive 389% compared to the AFC method.

Although our adaptive neural network controller exhibited impressive performance, recognizing its limitations is crucial for future enhancements. Future research could focus on refining control parameters to attain a higher level of precision and adaptability. Intelligent algorithms based on human interaction movements could be employed to train disturbance observer parameters. Also, future studies could explore hybridizing our control framework with reinforcement learning for adaptive parameter tuning or Gaussian processes for uncertainty quantification. Incorporating the wearer's estimation strategy into training may further enhance the disturbance observer's efficiency and overall performance.

Acknowledgement: The authors extend their appreciation to the King Salman Center For Disability Research for funding this work through Research Group No. KSRG-2024-468.

Funding Statement: This research was funded by the King Salman Center For Disability Research, through Research Group No. KSRG-2024-468.

Author Contributions: Conceptualization, Mohammad Soleimani Amiri, Sahbi Boubaker, Souad Kamel and Rizauddin Ramli; methodology, Mohammad Soleimani Amiri and Rizauddin Ramli; software, Mohammad Soleimani Amiri; validation, Mohammad Soleimani Amiri, Rizauddin Ramli and Sahbi Boubaker; formal analysis, Mohammad Soleimani Amiri, Rizauddin Ramli and Sahbi Boubaker; investigation, Mohammad Soleimani Amiri, Souad Kamel and Rizauddin Ramli; resources, Mohammad Soleimani Amiri; data curation, Mohammad Soleimani Amiri, Souad Kamel and Rizauddin Ramli; writing—original draft preparation, Mohammad Soleimani Amiri; writing—review and editing, Sahbi Boubaker, Rizauddin Ramli and Souad Kamel; visualization, Mohammad Soleimani Amiri; supervision, Rizauddin Ramli; project administration, Sahbi Boubaker; funding acquisition, Sahbi Boubaker. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Ethics Approval: Not applicable

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Zipser CM, Cragg JJ, Guest JD, Fehlings MG, Jutzeler CR, Anderson AJ, et al. Cell-based and stem-cell-based treatments for spinal cord injury: evidence from clinical trials. Lancet Neurol. 2022;21(7):659–70. doi:10.1016/S1474-4422(21)00464-6.
2. Lin PH, Dong Q, Chew SY. Injectable hydrogels in stroke and spinal cord injury treatment: a review on hydrogel materials, cell-matrix interactions and glial involvement. Mat Adv. 2021;2(8):2561–83. doi:10.1039/d0ma00732c.
3. Wang J, Pei S, Guo J, Bao M, Yao Y. A vector-based motion retargeting approach for exoskeletons with shoulder girdle mechanism. Biomimetics. 2025;10(5):312–27. doi:10.3390/biomimetics10050312.
4. Zhou J, Yang S, Xue Q. Lower limb rehabilitation exoskeleton robot: a review. Adv Mech Eng. 2021;13(4):16878140211011862. doi:10.1177/16878140211011862.
5. Wang T, Chenhao BZL, Liu T, Han Y, Ferreira SWJP, Dong W, et al. A review on the rehabilitation exoskeletons for the lower limbs of the elderly and the disabled. Electronics. 2022;11(3):1–16. doi:10.3390/electronics11030388.
6. Joel JPA, Raj RJS, Muthukumaran N. Review on gait rehabilitation training using human adaptive mechatronics system in biomedical engineering. In: 2022 International Conference on Computer Communication and Informatics (ICCCI); 2022 Jan 25–27; Coimbatore, India. p. 1–5. doi:10.1109/ICCCI54379.2022.9740794.
7. Xia Y, Li J, Yang D, Wei W. Gait phase classification of lower limb exoskeleton based on a compound network model. Symmetry. 2023;15(1):163. doi:10.3390/sym15010163.
8. Cortese M, Cempini M, De Almeida R, Paulo R, Soekadar SR, Carrozza MC, et al. A Mechatronic system for robot-mediated hand telerehabilitation. IEEE/ASME Trans Mechatr. 2015;20(4):1753–64. doi:10.1109/TMECH.2014.2353298.
9. Mohammadi M, Knoche H, Thøgersen M, Bengtson SH, Kobbelgaard FV, Gull MA, et al. Tongue control of a five-DOF upper-limb exoskeleton rehabilitates drinking and eating for individuals with severe disabilities. Int J Hum Comput Stud. 2023;170(10):102962. doi:10.1016/j.ijhcs.2022.102962.
10. Qin L, Wang ZXJ, Lu G, Ji H. Control method in coordinated balance with the human body for lower-limb exoskeleton rehabilitation robots. Biomimetics. 2025;10(5):324–48. doi:10.3390/biomimetics10050324.
11. Song X, Yi B, Chen Q, Zhou Y, Cho H, Hong Y, et al. Machine learning-powered ultrahigh controllable and wearable magnetoelectric piezotronic touching device. Am Chem Soc. 2024;18(26):16648–57. doi:10.1021/acsnano.4c01102.
12. Shi D, Zhang W, Zhang W, Ding X. A review on lower limb rehabilitation exoskeleton robots. Chin J Mech Eng. 2019;32(1):74. (In English). doi:10.1186/s10033-019-0389-8.
13. Rahmani M, Rahman MH. An upper-limb exoskeleton robot control using a novel fast fuzzy sliding mode control. J Intell Fuzzy Syst. 2019;36(3):2581–92. doi:10.3233/JIFS-181558.
14. Razzaghian A. A fuzzy neural network-based fractional-order Lyapunov-based robust control strategy for exoskeleton robots: application in upper-limb rehabilitation. Mathem Comput Simulat. 2022;193(22):567–83. doi:10.1016/j.matcom.2021.10.022.
15. Zhao J, Zhang Y, Hou H, Yue Y, Meng K, Yang Z. Active disturbance rejection control with backstepping for decoupling control of hydraulic driven lower limb exoskeleton robot. IEEE Trans Indust Elect. 2025;72(1):714–23. doi:10.1109/TIE.2024.3413820.
16. Zhang X, Wang H, Tian Y, Peyrodie L, Wang X. Model-free based neural network control with time-delay estimation for lower extremity exoskeleton. Neurocomputing. 2018;272(10):178–88. doi:10.1016/j.neucom.2017.06.055.
17. Yang Y, Huang D, Dong X. Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning. Neurocomputing. 2019;323(4):256–64. doi:10.1016/j.neucom.2018.09.085.
18. AL-Dujaili AQ, Hasan AF, Humaidi AJ, Al-Jodah A. Anti-disturbance control design of Exoskeleton Knee robotic system for rehabilitative care. Heliyon. 2024;10(9):e28911. doi:10.1016/j.heliyon.2024.e28911.
19. Jabbari Asl H, Narikiyo T, Kawanishi M. Neural network-based bounded control of robotic exoskeletons without velocity measurements. Control Eng Pract. 2018;80(12):94–104. doi:10.1016/j.conengprac.2018.08.005.

20. Wu Q, Zheng L, Zhu Y, Xu Z, Zhang Q, Wu H. Development of a reconfigurable 7-DOF upper limb rehabilitation exoskeleton with gravity compensation based on DMP. IEEE Trans Med Robot Bionics. 2025;7(1):303–14. doi:10.1109/TMRB.2024.3517157.
21. Alawad NA, Humaidi AJ, S Alaraji A. Sliding mode-based active disturbance rejection control of assistive exoskeleton device for rehabilitation of disabled lower limbs. An Da Acad Bras De Cienc. 2023;95(2):e20220680. doi:10.1590/0001-3765202320220680.
22. Wu Q, Wagn X, Chen B, Wu H. Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. Mechatronics. 2018;53(3):85–94. doi:10.1016/j.mechatronics.2018.05.014.
23. Liu Q, Liu Y, Zhu C, Guo X, Meng W, Ai Q, et al. Design and control of a reconfigurable upper limb rehabilitation exoskeleton with soft modular joints. IEEE Access. 2021;9:166815–24. doi:10.1109/ACCESS.2021.3136242.
24. Zhang G, Wang J, Yang P, Guo S. A learning control scheme for upper-limb exoskeleton via adaptive sliding mode technique. Mechatronics. 2022;86(2):102832. doi:10.1016/j.mechatronics.2022.102832.
25. Han S, Wang H, Tian Y, Christov N. Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton. ISA Trans. 2020;97(2):171–81. doi:10.1016/j.isatra.2019.07.030.
26. He D, Wang H, Tian Y, Guo Y. A fractional-order ultra-local model-based adaptive neural network sliding mode control of n-DOF upper-limb exoskeleton with input deadzone. IEEE/CAA J Autom Sin. 2024;11(3):760–81. doi:10.1109/JAS.2023.123882.
27. Huang P, Li Z, Zhou M, Li X, Cheng M. Fuzzy enhanced adaptive admittance control of a wearable walking exoskeleton with step trajectory shaping. IEEE Trans Fuzzy Syst. 2022;30(6):1451–552. doi:10.1109/tfuzz.2022.3162700.
28. Amiri MS, Ramli R, Aliman N. Adaptive swarm fuzzy logic controller of multi-joint lower limb assistive robot. Machines. 2022;10(6):425. doi:10.3390/machines10060425.
29. Ruiz-Ruiz FJ, Ventura J, Urdiales C, Gómez-de-Gabriel JM. Compliant gripper with force estimation for physical human-robot interaction. Mech Mach Theory. 2022;178(8):105062. doi:10.1016/j.mechmachtheory.2022.105062.
30. Amiri MS, Ramli R. Utilisation of initialised observation scheme for multi-joint robotic arm in Lyapunov-based adaptive control strategy. Mathematics. 2022;10(17):3126. doi:10.3390/math10173126.
31. Zhang L, Li Z, Yang C. Adaptive neural network based variable stiffness control of uncertain robotic systems using disturbance observer. IEEE Trans Indus Elect. 2017;64(3):2236–45. doi:10.1109/TIE.2016.2624260.
32. Voigtlaender F. The universal approximation theorem for complex-valued neural networks. Appl Comput Harmon Anal. 2023;46(04):33–61. doi:10.1016/j.acha.2022.12.002.
33. Anand A, Seel K, Gjærum V, Håkansson A, Robinson H, Saad A. Safe learning for control using control Lyapunov functions and control barrier functions: a review. Procedia Comput Sci. 2021;192(4-5):3987–97. doi:10.1016/j.procs.2021.09.173.
34. Shi W. Adaptive fuzzy output-feedback control for nonaffine MIMO nonlinear systems with prescribed performance. IEEE Trans Fuzzy Syst. 2021;29(5):1107–20. doi:10.1109/TFUZZ.2020.2969110.
35. Yang Y, Cui Y, Qiao J, Zhu Y. Adaptive periodic-disturbance observer based composite control for SGCMG gimbal servo system with rotor vibration. Control Eng Pract. 2023;132(2):105407. doi:10.1016/j.conengprac.2022.105407.
36. Qiu Z, Duan C, Yao W, Zeng P, Jiang L. Adaptive Lyapunov function method for power system transient stability analysis. IEEE Trans Power Syst. 2023;38(4):3331–44. doi:10.1109/TPWRS.2022.3199448.
37. Amiri MS, Ramli R. Fuzzy adaptive controller of a wearable assistive upper limb exoskeleton using a disturbance observer. IEEE Trans Human-Mach Syst. 2025;55(2):197–206. doi:10.1109/THMS.2025.3529759.
38. Amiri MS, Ramli R, Van M. Swarm-initialized adaptive controller with beetle antenna searching of wearable lower limb exoskeleton for sit-to-stand and walking motions. ISA Trans. 2025;158(5):640–53. doi:10.1016/j.isatra.2025.01.003.
39. Amiri MS, Ramli R. Offline tuning mechanism of joint angular controller for lower-limb exoskeleton with adaptive biogeographical-based optimization. Turk J Elect Eng Comput Sci. 2022;30(4):1654. doi:10.55730/1300-0632.3871.