Active Neural Network Control for a Wearable Upper Limb Rehabilitation Exoskeleton Robot Driven by Pneumatic Artificial Muscles

Haoqi Zhang, *Graduate Student Member, IEEE*, Jiade Fan, Yanding Qin, *Member, IEEE*, Mengqiang Tian, and Jianda Han, *Member, IEEE*

***Abstract*—Pneumatic artificial muscle (PAM) has been widely used in rehabilitation and other fields as a flexible and safe actuator. In this paper, a PAM-actuated wearable exoskeleton robot is developed for upper limb rehabilitation. However, accurate modeling and control of the PAM are difficult due to complex hysteresis. To solve this problem, this paper proposes an active neural network method for hysteresis compensation, where a neural network (NN) is utilized as the hysteresis compensator and unscented Kalman filtering is used to estimate the weights and approximation error of the NN in real time. Compared with other inversion-based methods, the NN is directly used as the hysteresis compensator without needing inversion. Additionally, the proposed method does not require pre-training of the NN since the weights can be dynamically updated. To verify the effectiveness and robustness of the proposed method, a series of experiments have been conducted on the self-built exoskeleton robot. Compared with other popular control methods, the proposed method can track the desired trajectory faster, and tracking accuracy is gradually improved through iterative learning and updating.**

***Index Terms*—Upper limb rehabilitation, pneumatic artificial muscle, unscented Kalman filter, multilayer feedforward neural network, hysteresis compensation.**

I. Introduction

ROTATOR cuff injury is a condition characterized by damage or rupture of the tendons around the shoulder joint and associated muscle groups. In the treatment and recovery process, it is often essential to engage in repetitive task-oriented movements to activate the affected limb. This is beneficial to the restoration of muscle strength and joint function. Generally, the rehabilitation process often demands the involvement of multiple medical staffs to assist patients during training. This not only necessitates additional manpower but may also place a significant economic burden on both patients and society. Many studies have demonstrated that rehabilitation robots are effective in treating patients with muscle injuries. This provides a promising solution to patients who are unable to complete independent rehabilitation training due to neurological or motor function disorders [1], [2], [3], [4], [5].

Currently, motors are commonly employed as actuators for exoskeleton robots, despite their limitations in terms of flexibility [6]. As a new type of flexible actuator, pneumatic artificial muscle (PAM) emulates the structure and functioning principles of muscles. The inherent flexibility of PAM allows for superior adaptation to rehabilitation tasks, providing natural and smooth actuation [7], [8], [9], [10]. However, due to its special material properties, friction and inertia forces generate obvious hysteresis during actuation. Moreover, the hysteresis of PAM is asymmetric and rate-dependent [11], [12], increasing the complexities and difficulties in modeling and control.

Researchers have conducted extensive researches on the modeling and control methods of PAMs, which can be briefly divided into phenomenon models, physical models, and neural network (NN) models. Among them, physical models are based on the principles of aerodynamics and mechanics to describe the behavior of PAMs, such as the three-element model and the fluid structure interaction model. Phenomenological models are usually simple in structure and easy to use, such as Preisach model, Maxwell model, Bouc Wen model. NN models use artificial neural networks to learn and simulate the behavior of PAMs. They can automatically learn the nonlinear relationships through a large number of training data and can carry out online adaptive learning and control. Multilayer feedforward neural network (MFNN) has shown great applicability in nonlinear system modeling. W. Liu proposes an inversion-free predictive controller, which is based on a dynamic linearized MFNN model [13]. Firstly, it was verified that MFNN has high modeling accuracy at

Manuscript received 16 January 2024; revised 7 June 2024; accepted 8 July 2024. Date of publication 16 July 2024; date of current version 22 July 2024. This work was supported in part by Shenzhen Science and Technology Program under Grant KQTD20210811090143060, in part by the Natural Science Foundation of Tianjin under Grant 21JCZDJC00090, and in part by the Sustainable Development Science and Technology Special Project of Shenzhen under Grant KCXFZ20230731100900002. *(Haoqi Zhang and Jiade Fan are co-first authors.) (Corresponding author: Yanding Qin.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethical Committee of Nankai University under Application No. NKUIRB2024007.

Haoqi Zhang, Jiade Fan, Yanding Qin, and Jianda Han are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China, and also with Shenzhen Research Institute, Nankai University, Shenzhen 518083, China (e-mail: 1120230255@mail.nankai.edu.cn; 2120220471@mail.nankai.edu.cn; qinyd@nankai.edu.cn; hanjianda@nankai.edu.cn).

Mengqiang Tian is with the Department of Joint and Sport Medicine, Tianjin Union Medical Center, Tianjin 300121, China (e-mail: tmqjoint@126.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2024.3429206, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2024.3429206

different frequencies. Then, trajectory tracking experiments showed that the proposed method can effectively improve the tracking accuracy of the system.

Inversion-based hysteresis compensation approaches have been widely adopted, where the inverse hysteresis model is obtained and used as the hysteresis compensator. However, inverting a hysteresis model is always challenging. Furthermore, in some instances, inversion may not even be possible. As a result, several inversion-free methods have been proposed. For instance, M. Rakotondrabe proposed an inverse multiplication scheme to avoid inversion calculations [14]. Y. Qin proposed a direct inverse modeling approach to identify the inverse hysteresis model directly from measurement data [15].

Although scholars have proposed various hysteresis compensation methods for PAM, due to the complex nonlinearity of PAMs, most models are only approximations of the real system. The significant modeling errors affect the precision of hysteresis compensation. Therefore, to further improve trajectory tracking accuracy, closed-loop control is generally implemented as it can make real-time corrections to the control variables based on tracking errors. For instance, Y. Zhang proposed a feedforward and feedback combined compensation method based on an MFNN inverse model [16]. This method requires the prior identification of inverse models from experimental data. This offline identification might pose another challenge for inexperienced users.

Kalman filtering, commonly used as a state estimator, produces more accurate state estimation over time compared to a single measurement [17]. J. Escareno proposed using a discrete extended-state linear Kalman filter to estimate disturbances and velocity [18]. For nonlinear systems, the most commonly used state estimator is the extended Kalman filter (EKF). However, the EKF also exhibits obvious shortcomings, including the requirement for sufficient differentiability of state dynamics and susceptibility of state estimates to bias and divergence. On the other hand, the unscented Kalman filter (UKF) directly uses the nonlinear model instead of linearizing it, making it more applicable [19], [20], [21].

Based on the above analyses, this paper proposes an active neural network (ANN) method for hysteresis compensation in a PAM-actuated exoskeleton robot, where an NN serves as the hysteresis compensator and its weights and approximation error are dynamically estimated using UKF. Without loss of generality, MFNN is adopted. The proposed method is experimentally verified on a self-built upper limb rehabilitation exoskeleton robot. For this multi-degree-of-freedom (DOF) system, the complex dynamic characteristics introduce additional challenges to system control. Thanks to the nonlinear approximation ability of MFNN and the estimation accuracy of UKF, the proposed method efficiently compensates for hysteresis and improves trajectory tracking performance. The contributions of the proposed ANN can be summarized as follows:

1) A neural network is employed to directly formulate the hysteresis compensator for PAM-actuated systems. Real-time parameter identification is accomplished via UKF, completely eliminating the need for inversion calculation and offline training of the neural network.

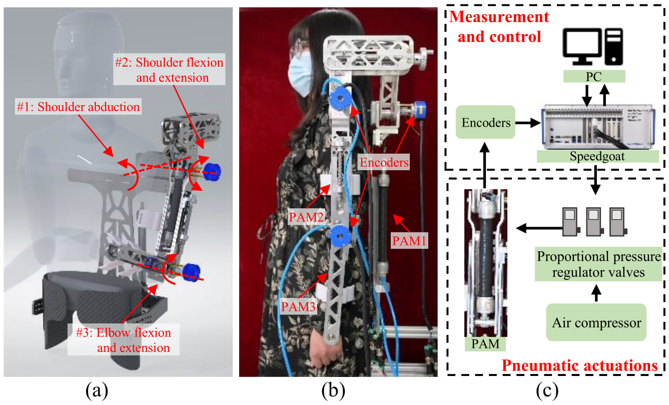


Fig. 1. The self-built upper limb rehabilitation exoskeleton robot: (a) 3D model showing the DOFs, (b) photograph of the robot worn on a participant, and (c) schematic diagram of the control system setup.

2) In UKF, in addition to the weights of the neural network, the approximation error of the neural network is also estimated as an augmented state. The estimated approximation error is then integrated into the control input. The proposed ANN achieves rapid convergence within the first two or three iterations, demonstrating robust iterative learning capability.

3) The developed PAMs actuated exoskeleton robot, together with the proposed ANN, can provide robust and safe assistance in completing iterative motions involved in upper limb rehabilitation trainings.

The rest of the paper is organized as follows: Section II describes the self-built upper limb rehabilitation exoskeleton robot system. Section III introduces the architecture of the proposed active neural network method. Section IV shows the experimental results in tracking different trajectories. Section V summarizes this paper.

II. Prototype Development of the Upper Limb Rehabilitation Exoskeleton Robot

To assist patients with rotator cuff injuries who are unable to complete rehabilitation training independently, a wearable upper limb rehabilitation exoskeleton robot prototype is developed. As shown in Fig. 1, the developed exoskeleton robot can provide 3-DOF assistances for shoulder joint abduction, shoulder joint flexion and extension, and elbow joint flexion and extension, respectively. Three PAMs (DMSP-20 series, Festo) are used to provide flexible and safe actuation. Three proportional pressure regulator valves (VPPM-6L series, Festo) are used to regulate the pressure of the PAMs within 0-0.6 MPa. The rotations of the axes are measured using encoders (OIH48-2500P8-L6, Tamagawa). A real-time target machine (Performance, Speedgoat) is used to compile and execute the control algorithm at a sampling rate of 1 kHz.

Strong hysteretic nonlinearities exist in this exoskeleton robot. Without loss of generality, the elbow flexion and extension motion are selected as an example. A 0-5 V sinusoidal control voltage is applied to the PAM in the axis. The hysteresis loops at 0.05 Hz, 0.1 Hz and 0.2 Hz are recorded and shown in Fig. 2. It can be observed that the shape of the

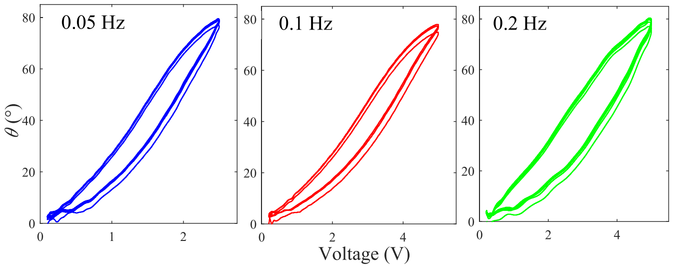


Fig. 2. Hysteresis loops of the elbow's flexion and extension module at different frequencies.

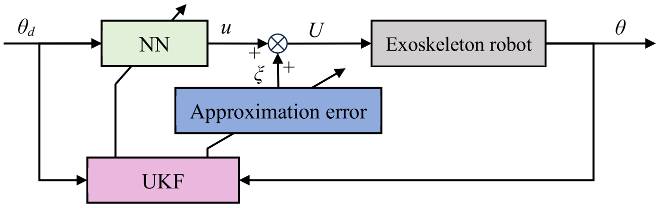


Fig. 3. Schematic diagram of the proposed active neural network control.

hysteresis loop slightly varies at different frequencies, showing obvious rate dependence and asymmetry. The hysteresis loops in the other two axes are similar, and thus they will not be shown for the conciseness of the paper.

III. Active Neural Network Control Using UKF

As stated in Introduction, the hysteresis nonlinearity of PAM exhibits strong rate dependence and asymmetry. NN has been widely used in hysteresis compensation of PAMs. Generally, for conventional NN-based methods, offline data acquisition and parameter identification are required to obtain a precise model. However, for the exoskeleton robot, the dynamic characteristics of the robot will be mixed with the hysteresis nonlinearity. In addition, in rehabilitation training, PAMs will inevitably be affected by unknown external disturbances and load changes. These uncertainties will degrade the accuracy of the NN. In this paper, online parameter identification is achieved with the help of UKF, a powerful state estimator. Further, UKF is also used to estimate the approximation error of the NN, which is then integrated into the control signal. The schematic diagram of the proposed method is illustrated in Fig. 3. The nonlinear approximation capability of NN and the powerful estimation capability of UKF help to improve the hysteresis compensation performance of the exoskeleton robot.

A. NN-Based Hysteresis Compensator

In each axis, the hysteresis can be regarded as the mapping from the control input to the rotation angles. In this paper, NN is adopted to construct the inverse hysteresis model for each axis. Without loss of generality, MFNN is adopted herein. MFNN includes an input layer, a hidden layer and an output layer, as schematically shown in Fig. 4. The hysteresis of the PAM is not only influenced by the current state of the system but also the historical states of the system. Therefore, in addition to the current desired trajectory θ_d, the input layer neuron of the MFNN also incorporates the previous desired trajectory θ_{d_pre}, to enhance the local memory of the network.

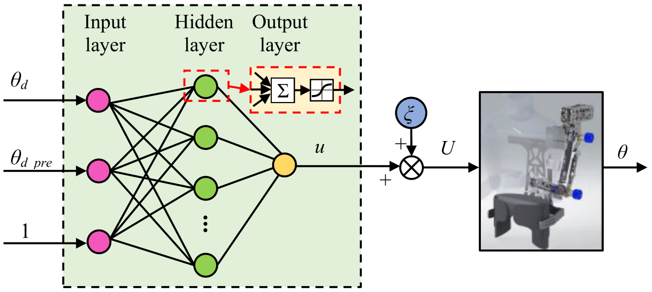


Fig. 4. MFNN-based feedforward hysteresis compensation.

In addition, 1 in the input layer neuron is used to increase the generalization ability of the MFNN [22]. The activation function of the hidden layer is adopted to be:

$$h(x) = (e^x - e^{-x})\big/(e^x + e^{-x}) \tag{1}$$

The output of MFNN is defined as follows:

$$u = f_{NN}(\mathbf{x}, \mathbf{A}, \mathbf{B}) = \sum\nolimits_{j=1}^{7} \mathbf{B}_j h_j(\Sigma_{i=1}^{3}\mathbf{x}_i\mathbf{A}_{ij}) \tag{2}$$

where $\mathbf{x}$ is the input layer state vector defined as $\mathbf{x}$=[x_1 x_2 x_3] =[θ_d θ_{d_pre} 1], $\mathbf{A}$=[a_{11} a_{12} $\ldots a_{17}$ a_{21} a_{22} $\ldots a_{27}$ a_{31} a_{32} $\ldots a_{37}$] is the weight vector from the input layer to the hidden layer, among them, $\mathbf{A}_{ij}$ is the weight value from the i-th input layer neuron to the j-th hidden layer neuron. $h_j(\Sigma_{i=1}^{3}\mathbf{x}_i\mathbf{A}_{ij})$ is the output of the j-th hidden layer neuron. $\mathbf{B}$=[b_1 b_2 $\ldots b_7$] is the weight vector from the hidden layer to the output layer, and u is the output of the neural network.

B. Active Neural Network Controller for Each Axis

Although the NN has strong nonlinear approximation capabilities, approximation error is inevitable, especially in the very beginning of motion. In this paper, the UKF is utilized to dynamically identify the weights of the MFNN, eliminating the off-line parameter identification. Further, UKF is also used to estimate the approximation error and add it to the NN's output, forming the final control input to the PAM, as shown in Fig. 4.

UKF uses the nonlinear model directly through the unscented transform (UT). UT approximates the state distribution with a limited set of points, i.e., the sigma points calculated from the prior mean and covariance. In this paper, the extended state is defined as the weights and approximation error of the MFNN, as illustrated below:

$$\begin{aligned}\mathbf{X} &= [\mathbf{A}\,\mathbf{B}\ \xi]^T \\ &= [a_{11}\ a_{12}\cdots a_{17}\ \ a_{21}\ a_{22}\cdots a_{37}\ b_1\ b_2\cdots b_7\ \xi]^T\end{aligned} \tag{3}$$

where ξ is the approximation error of the MFNN.

Subsequently, the UKF is employed to predict and update the extended state. UKF consists of the following two steps:

The first step is to calculate the sampling points based on the estimated mean and covariance:

$$\begin{aligned}\mathbf{\Psi}_{k-1} &= \left[\mathbf{X}_{k-1}\cdots\mathbf{X}_{k-1}\right]_{n\times(2n+1)} \\ &\quad - \left[\mathbf{0}_{n\times 1} - \sqrt{(n+\kappa)\mathbf{P}_{k-1|k-1}}\sqrt{(n+\kappa)\mathbf{P}_{k-1|k-1}}\right] \\ &= [\mathbf{\Psi}_{k-1}(1)\cdots\mathbf{\Psi}_{k-1}(2n+1)]_{1\times(2n+1)}\end{aligned} \tag{4}$$

where n is the state dimension, κ is a constant, $\mathbf{\Psi}_{k-1}$ is the matrix after sampling expansion, calculated from the estimated mean and covariance calculated on the state vector $\mathbf{X}_{k-1}$. $\mathbf{\Psi}_{k-1}(m)$, $m = 1,2\ldots 2n+1$, is the m-th column vector of matrix $\mathbf{\Psi}_{k-1}$, and $\mathbf{\Psi}_{k-1}(m) = [\mathbf{A}_{k-1}(m)\ \mathbf{B}_{k-1}(m)\ \xi(m)]^{\mathrm{T}}$.

The second step is the prediction:

$$\begin{aligned}
&\mathbf{\Psi}_{k|k-1} = F(\mathbf{\Psi}_{k-1})\\
&\mathbf{P}_{X(k|k-1)} = \sum_{i=1}^{2n+1}\Bigg[\mathbf{W}_c(i)\Bigg(\mathbf{\Psi}_{k|k-1}(i) - \sum_{j=1}^{2n+1}\mathbf{W}_m(j)\mathbf{\Psi}_{k|k-1}(j)\Bigg)\\
&\qquad\Bigg(\mathbf{\Psi}_{k|k-1}(i) - \sum_{j=1}^{2n+1}\mathbf{W}_m(j)\mathbf{\Psi}_{k|k-1}(j)\Bigg)^{\mathrm{T}}\Bigg] + \mathbf{Q}_k
\end{aligned} \tag{5}$$

where $\mathbf{\Psi}_{k|k-1}$ is obtained by nonlinear conversion of sampling points $\mathbf{\Psi}_{k-1}$, $\mathbf{Q}_k$ is the covariance matrix of process noise, the process noise is considered in this step to obtain the weighted covariance matrix in the state space $\mathbf{P}_{X(k|k-1)}$. The state update function F can be set to 1. In this case, the previous state observation $\hat{w}_{k-1}$ is directed inherited in the prediction of $\hat{w}_k$.

The third step is the update:

$$\begin{aligned}
\Upsilon_{\psi(k|k-1)} &= [f_{NN}(\mathbf{x}_{k-1}, \mathbf{A}_{k|k-1}(1), \mathbf{B}_{k|k-1}(1)) + \xi(1)\\
&\quad \cdots f_{NN}(\mathbf{x}_{k-1}, \mathrm{A}_{k|k-1}(2n+1), \mathbf{B}_{k|k-1}(2n+1))\\
&\quad + \xi(2n+1)]\\
\mathbf{P}_{\Upsilon(k|k-1)} &= \sum_{i=0}^{2n}\Bigg[\mathbf{W}_c(i)(\Upsilon_{\psi(k|k-1)}(i)\\
&\quad - \sum_{j=0}^{2n}\mathbf{W}_m(j)\Upsilon_{\psi(k|k-1)}(j))\\
&\quad (\Upsilon_{\psi(k|k-1)}(i) - \sum_{j=0}^{2n}\mathbf{W}_m(j)\Upsilon_{\chi(k|k-1)}(j))^{\mathrm{T}}\Bigg] + \mathbf{R}_k\\
\mathbf{P}_{X\Upsilon(k|k-1)} &= \sum_{i=0}^{2n}\Bigg[\mathbf{W}_c(i)(\mathbf{\Psi}_{k|k-1}(i) - \sum_{j=0}^{2n}\mathbf{W}_m(j)\mathbf{\Psi}_{k|k-1}(j))\\
&\quad (\Upsilon_{\chi(k|k-1)}(i) - \sum_{j=0}^{2n}\mathbf{W}_m(j)\Upsilon_{\chi(k|k-1)}(j))^{\mathrm{T}}\Bigg]\\
\mathbf{K}_k &= \mathbf{P}_{X\Upsilon(k|k-1)}\mathbf{P}_{\Upsilon(k|k-1)}^{-1}\\
\mathbf{X}_{k|k} &= \mathbf{W}_m\mathbf{\Psi}_{k|k-1} + \mathbf{K}_k\Theta\\
\mathbf{P}_{k|k} &= \mathbf{P}_{X(k|k-1)} - \mathbf{K}_k\mathbf{P}_{X\Upsilon(k|k-1)}^{\mathrm{T}}
\end{aligned} \tag{6}$$

where $\mathbf{R}_k$ is the covariance matrix of the measurement noise, $\Upsilon_{\psi(k|k-1)}$ is the measurement prediction of $\mathbf{\Psi}_{k-1}$ from time k-1 to time k, $\mathbf{K}_k$ is the Kalman gain, Θ is the innovation, $\mathbf{X}_{k|k}$ is the estimation of state vector based on confidence field $\mathbf{\Psi}_{k|k-1}$, $\mathbf{P}_{k|k}$ is the update of confidence matrix $\mathbf{P}_{k-1|k-1}$. In this paper, the measurement function f_{NN} employs the MFNN structure described in (2), $\mathbf{x}_{k-1}$ is the sampling value of $\mathbf{x}$ at the k-1 moment. It is worthing noting that, $\mathbf{x}_{k-1}$ can be obtained from the desired trajectory, $\mathbf{A}$, $\mathbf{B}$ and ξ are update at each sample time.

In the UKF, the difference between the estimated and actual states needs to be used as the innovation to update the state at the next moment. In this paper, the difference between the desired joint angle θ_d and the actual measured joint angle θ is directly used as the innovation, as expressed below:

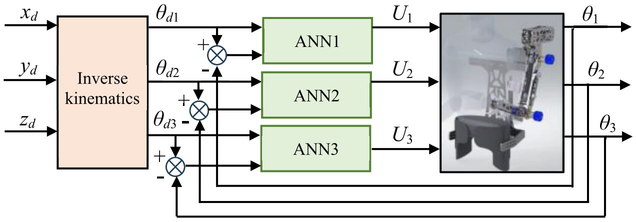


Fig. 5. Schematic diagram of the closed-loop controller.

$$\Theta = \theta_d - \theta \tag{7}$$

Further, if the input and output are on different orders of magnitudes, in order to map the difference in the joint angle to the difference in the input voltage, the innovation is scaled by a constant coefficient of $\eta = u_{max}/\theta_{max}$, where u_{max} and θ_{max} are the maximum input voltage and output angle of the system, respectively. Because η is only used to align the orders of magnitude, exact values of u_{max} and θ_{max} are not required. Status updates are redefined as:

$$\mathbf{X}_{k|k} = \mathbf{W}_m\mathbf{\Psi}_{k|k-1} + \mathbf{K}_k \cdot \eta \cdot (\theta_d - \theta) \tag{8}$$

Finally, the control voltage of the proposed controller can be obtained as:

$$U_k = f_{NN}(\mathbf{x}_{k-1}, \mathbf{A}_{k|k}, \mathbf{B}_{k|k}) + \xi_{k|k} \tag{9}$$

C. Closed-Loop Control of the Overall System

For the exoskeleton robot, in addition to the trajectory tracking accuracy in joint space, the accuracy of trajectory tracking in task space is also crucial. The forward kinematics of the exoskeleton robot was derived in our previous work [23]. In this paper, for a given 3-DOF trajectory in the workspace, the inverse kinematics is employed to obtain the desired trajectories of each axis, as illustrated in Fig. 5.

IV. Experimental Verifications

Experiments are firstly conducted on a human phantom to evaluate the performance of the exoskeleton robot. In this case, the repeatability of the human phantom establishes an equitable foundation for comparing the performance of the proposed control method to other widely-used controllers. Moreover, the exoskeleton robot is worn by a healthy participant to assess its real-world performance in practical implementations. Detailed experimental results and video recordings are provided in this section and in the supplemented video, respectively.

A. Trajectory Tracking Performance in the Joint Space

The trajectory tracking performance is firstly tested in the joint space of the exoskeleton robot. Parameters of the proposed method are set as follows: The number of neurons in the hidden layer is set to seven, and the weights of MFNN are initialized to 0. The parameters in the UKF are set to: $\mathbf{Q}$=0.001×diag(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 10 10 10 10 10 10 10 100) and $\mathbf{R}$=1, u_{max} is set to 10 V, and θ_{max} is set to 80 °, i.e., $\eta = 0.125$.

Remark 1: A careful trial-and-error process is suggested to fine tune the parameters of MFNN and UKF. For MFNN, too few neurons in the hidden layer will cause inability to fit complex nonlinearities, and decrease the tracking accuracy. Conversely, too many neurons in the hidden layer will lead to overfitting and significantly increase the computational load of the algorithm. Balancing between trajectory tracking accuracy and complexity, we ultimately set the number of neurons in the hidden layer to seven. Successful tuning of **Q** and **R** matrices in UKF requires some prior knowledge of the system dynamics and measurement characteristics. Initially, these parameters can be roughly adjusted, and fine-tuning can be performed once the system stabilizes.

Proportional-integral-derivative (PID) controller is chosen as a comparative method, with the following parameters set: $k_p = 0.01$, $k_i = 0.2$, and $k_d = 0$. In addition, the adaptive projection (AP) algorithm is also chosen as the second comparative method. Detailed information on this AP algorithm can be found in our previous work [23]. In this paper, the parameter γ of the AP algorithm is tuned to 0.0016.

Remark 2: In comparison, when determining the parameters of the PID controller, we first use low-frequency trajectories for tuning, balancing its transient control performance and steady-state tracking error. We determine the final parameters through iterative trial and error, and then apply this set of parameters to other trajectory tracking experiments. The AP algorithm requires simpler parameter tuning, involving only one parameter, γ. A larger γ leads to an increase in the steady-state error, while a smaller γ may lead to greater overshoot and oscillation. The algorithm's performance under different parameter settings can be found in [23].

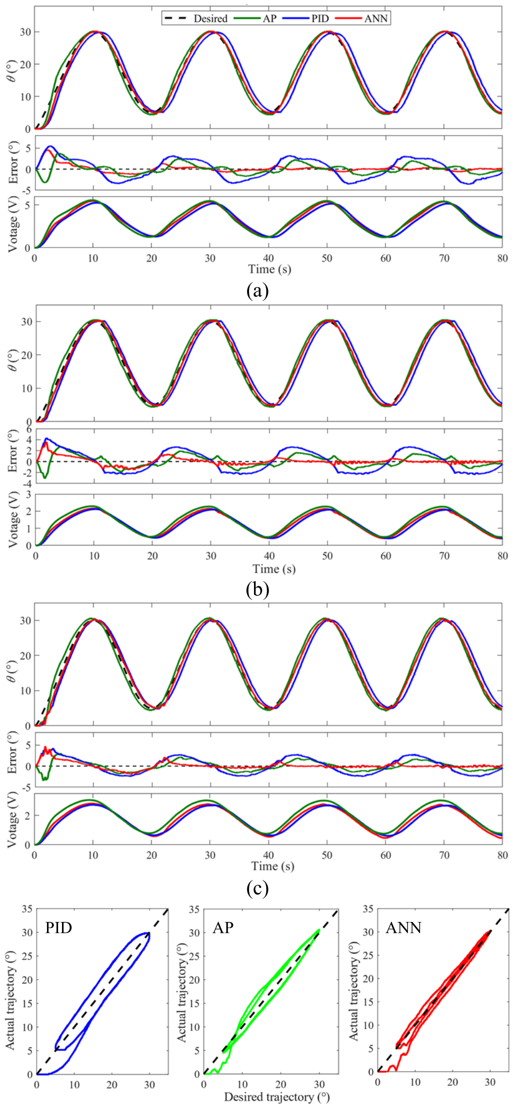
Group 1: Tracking of Slow Trajectories: First, the desired trajectories are set to be 0.05 Hz sinusoidal and triangular trajectories for all axes, corresponding to the slow motions in the beginning of rehabilitation trainings. The amplitude and offset of the sinusoidal trajectory are set to be 12.5° and 17.5°, respectively. For the triangular trajectory, the amplitude and offset are set to be 10° and 15°.

The sinusoidal trajectory tracking results are shown in Fig. 6. From the experimental results, it can be seen that the proposed ANN can quickly follow the desired trajectories, and after the second iteration, the tracking error gradually converges. After the third iteration, the steady-state error of ANN stays close to zero. In contrast, the PID controller converges slower without iterative learning capability, and the steady-state error is larger. The AP controller can converge to the desired trajectory fast, whereas the transient performance is poor. The steady-state error is between ANN and PID. Taking the elbow flexion-extension module as an example, Fig. 6 (d) displays the hysteresis compensation effects of three methods. It can be observed that the proposed method can better compensate for the asymmetry of the system hysteresis, and the hysteresis loop is closest to the 45° line, indicating a good hysteresis compensation performance.

The tracking results for the triangular trajectory are shown in Fig. 7. Similarly, ANN can quickly converge the steady-state error towards 0 after one iteration, showing a strong learning ability. In order to quantitatively compare the tracking errors, the following root-mean-square error (RMSE) is used:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[\theta(k) - \theta_d(k)]^2} \tag{10}$$

The statistics on the tracking errors are listed in TABLE I. It can be seen that ANN achieves the minimum RMSE values among the three methods when tracking slow sinusoidal and



(a)

(b)

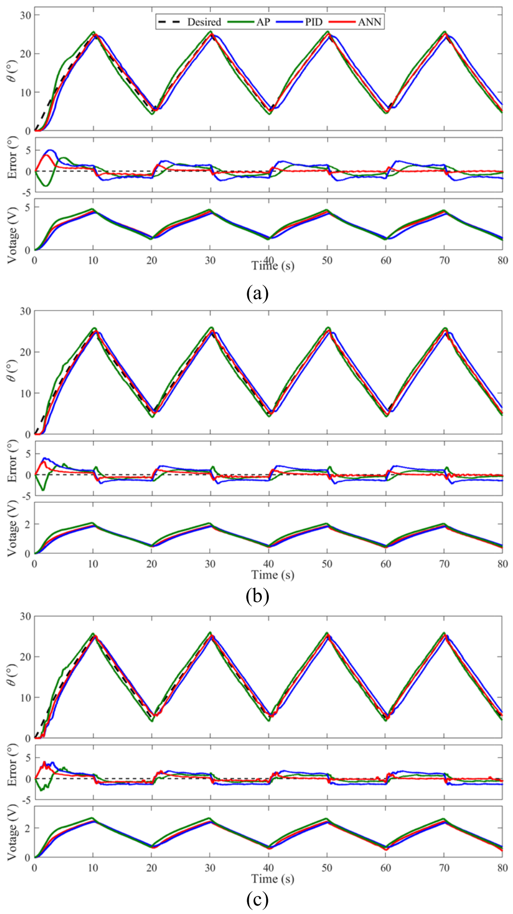
(c)

(d)

Fig. 6. 0.05 Hz sinusoidal trajectory tracking: (a)-(c) axis 1-3, (d) hysteresis plot.

TABLE I
RMSEs OF THE TRACKING ERRORS OF 0.05 Hz SLOW TRAJECTORIES

		ANN	AP	PID
Sinusoidal (°)	Axis 1	**0.6380**	1.1428	2.3219
	Axis 2	**0.5033**	0.9931	1.8651
	Axis 3	**0.6070**	1.1321	1.8077
Triangular (°)	Axis 1	**0.5235**	0.9566	1.6657
	Axis 2	**0.4571**	0.7885	1.4362
	Axis 3	**0.5295**	0.6910	1.3763



(a)

(b)

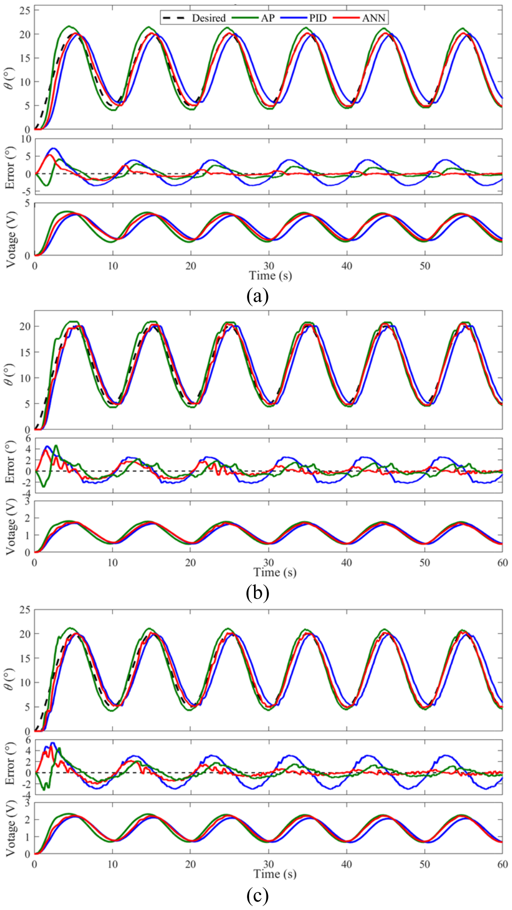
(c)

Fig. 7. 0.05 Hz triangular wave trajectory tracking: (a)-(c) axis 1-3.

triangular trajectories. According to the above results, the proposed ANN achieved better tracking performance.

Group 2: Tracking of Fast Trajectories: The hysteresis of the PAM is rate-dependent, i.e., the hysteresis loop widens as the frequency increases. Therefore, it is necessary to evaluate the tracking performance on fast trajectories. In this section, the frequency of the sinusoidal trajectory is increased to 0.1 Hz. The experimental results are shown in Fig. 8. For this faster trajectory, the proposed ANN converges faster than the other two comparative methods, and the steady-state error gradually converges towards zero. On the contrary, the PID controller exhibits periodic oscillations in the steady-state error. The performance of the AP controller degrades for faster trajectories. Although the tracking error decreases iteratively, the tracking performance is still not comparable to ANN. The RMSE values of the tracking errors are also calculated and listed in TABLE II. Statistics show that when tracking a faster trajectory, ANN continue to demonstrate good tracking performance, whereas AP and PID require more time to converge.

Further, three different trajectories are adopted as the desired trajectories to the axes of the exoskeleton robot to evaluate the performance of the proposed ANN in generating more complex rehabilitation motions. The desired trajectories include a 0.05 Hz sinusoidal trajectory for shoulder abduction, a 0.05-0.1 Hz swept sinusoidal trajectory for shoulder flexion and extension, and a 0.05 Hz sinusoidal trajectory with



(a)

(b)

(c)

Fig. 8. 0.1 Hz sinusoidal trajectory tracking: (a)-(c) axis 1-3.

TABLE II

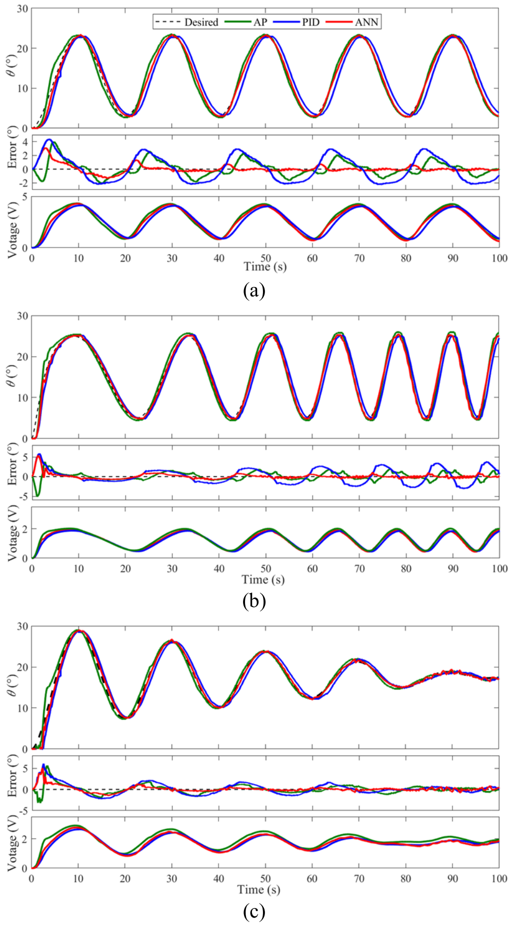
RMSEs of the Tracking Errors For Fast and Varying Trajectories

		ANN	AP	PID
0.1 Hz sinusoidal (°)	Axis 1	**0.6664**	1.2062	2.7123
	Axis 2	**0.5903**	0.9433	1.7719
	Axis 3	**0.6203**	1.9325	2.1736
Varying trajectory (°)	Axis 1	**0.5978**	1.0883	1.8435
	Axis 2	**0.6824**	0.9973	1.8503
	Axis 3	**0.6633**	0.9533	1.2543

descending amplitude for elbow flexion and extension. The experimental results are shown in Fig. 9, and the statistics on the tracking errors are listed in TABLE II. It can be observed that the proposed method still exhibits good transient and steady-state performances when tracking such complex trajectories. Specifically, when tracking the swept sinusoidal trajectory, the tracking errors of both AP and PID controllers increase with the increment of frequency and can go beyond ±1.5°. In contrast, the tracking errors of ANN gradually decreasing and ensuring they remain within ±1°, showing better learning capabilities.

B. Robustness Against External Disturbances

During rehabilitation trainings, the exoskeleton robot will encounter external disturbances, such the unknown disturbances from the patient. For this reason, it is necessary to test the robustness of the exoskeleton robot against external disturbances. To be concise, robustness is tested during elbow flexion and extension motions. The desired trajectory is set to



(a)

(b)

(c)

Fig. 9. Tracking results for varying trajectories: (a)-(c) axis 1-3.

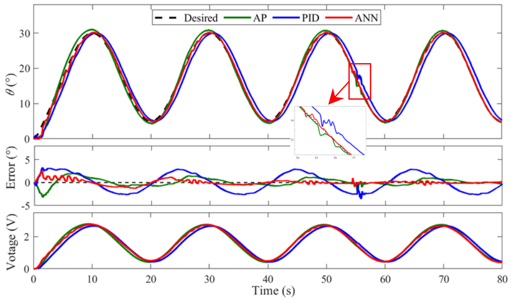


Fig. 10. Robustness against external disturbances.

be a 0.05 Hz sinusoidal trajectory with an amplitude of 12.5° and an offset of 17.5°, and an external disturbance is applied around the 55th s.

The control performances of the controllers before and after the disturbances are recorded and shown in Fig. 10. For the proposed ANN, it can be seen that a small oscillation of about 0.99° occur after the disturbance. For the AP controller, the amplitude of oscillation reaches 1.62°. For the PID controller, more oscillations are observed and the oscillation amplitude can increase to 2.21°. This demonstrates that the proposed ANN is able to quickly converge after the disturbance, exhibiting high robustness against external disturbances.

Remark 3: To assess the time-efficiency of the proposed method, the computation time is recorded on the real-time target machine. The mean and maximum values of the computation time are measured to be 0.113 ms and 0.117 ms, respectively, which are well below the sampling period of the closed-loop system, i.e., 1 ms. Therefore, the execution of the neural networks and UKF is fast within the current system step.

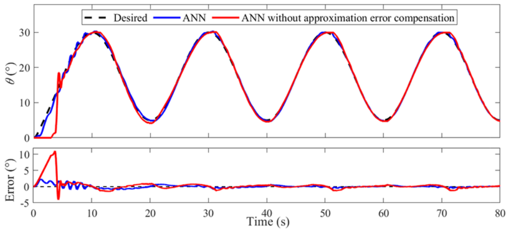


Fig. 11. Trajectory tracking performance of ANNs with and without approximation error compensation.

C. Approximation Error and Algorithm Convergence Verification

In section III, the approximation error of the neural network, defined as an extended state, is estimated and compensated in real-time. Typically, the training and convergence process of neural network weights often lead to a decline in transient tracking performance. However, the real-time estimation and compensation of the approximation error effectively address this limitation. To validate the role of neural network approximation error, the shoulder joint flexion and extension module is selected as an example to compare the performance of algorithms with and without approximation error. The desired trajectory is set to be a 0.05 Hz sinusoidal trajectory with an amplitude of 12.5° and an offset of 17.5°, The experimental results are shown in Fig. 11. From these results, it can be observed that the proposed method is able to respond rapidly during trajectory tracking transients. However, without approximation error compensation, the relatively slow convergence of neural network weights during the transient leads to a decline in tracking performance. In addition, it can be observed that without approximation error compensation, significant overshoot and oscillations can be observed. In upper limb rehabilitation trainings, such oscillations are dangerous and unacceptable. This further verifies the necessity of neural network approximation error compensation.

In order to show the stability of the proposed method, the convergence of the estimated weights and approximation error of the MFNN are analyzed. Because theses parameters are on different magnitudes, without loss of generality, only a_{11}, a_{31}, b_1 and ξ in Fig. 11(a) are selected and presented in Fig. 12. After the first iteration, the weights gradually converge and show a trend of small periodic oscillations around the steady-state values. Simultaneously, in the initial iteration when the weights of the MFNN don't converge, the estimated approximation error of the MFNN inclraweases to compensate for the modeling error. As the weights gradually converge, the estimated approximation error gradually decreases and oscillate around 0.5.

Finally, to further illustrate the convergence of the UKF in the proposed method, we observed the convergence of the Kalman gains, as depicted in Fig. 13. The figure displays the Kalman gains corresponding to weights a_{11}, a_{31}, b_1 and ξ. Similar to the convergence of weights, the Kalman gains

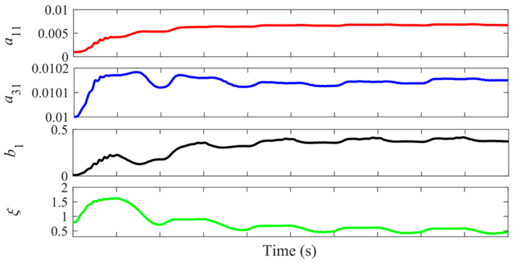


Fig. 12. Weights convergence of the UKF.

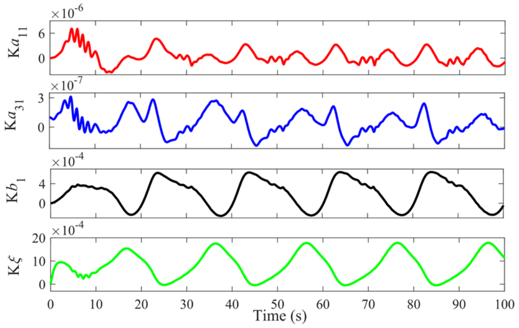


Fig. 13. Kalman gains of UKF.

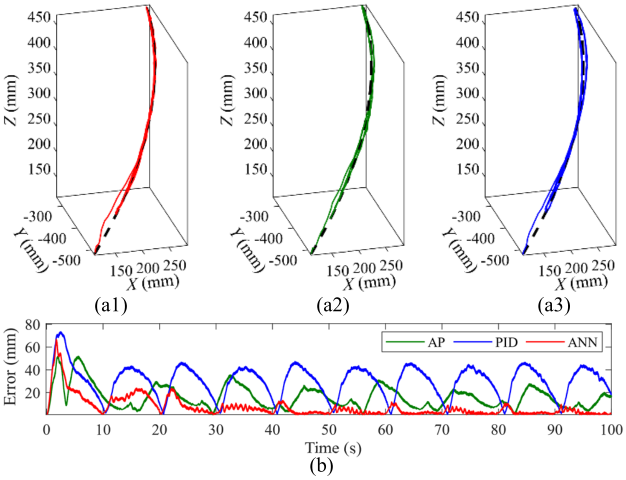


Fig. 14. Tracking of a 3-DOF spatial trajectory: (a1)∼(a3) the 3D plot of the trajectory of the robot using ANN, AP, and PID controllers, respectively, and (b) the tracking errors.

gradually converge to around zero after a brief adjustment period, displaying periodic oscillations, which further demonstrates the convergence of the UKF.

D. Trajectory Tracking Performance in the Workspace

The above experiments illustrate the effectiveness of the proposed method by observing the rotation angles of each axis. Furthermore, with the help of the kinematics of the robot, 3-DOF spatial trajectory tracking can be implemented within the robot's workspace. In this section, a spatial circular trajectory is adopted as the desired trajectory for the robot. The trajectory tracking performance is depicted in Fig. 14(a1) ∼(a3).

In order to quantitively compare the performances of the controllers, the following tracking error is adopted:

$$Error = \sqrt{(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2} \tag{11}$$

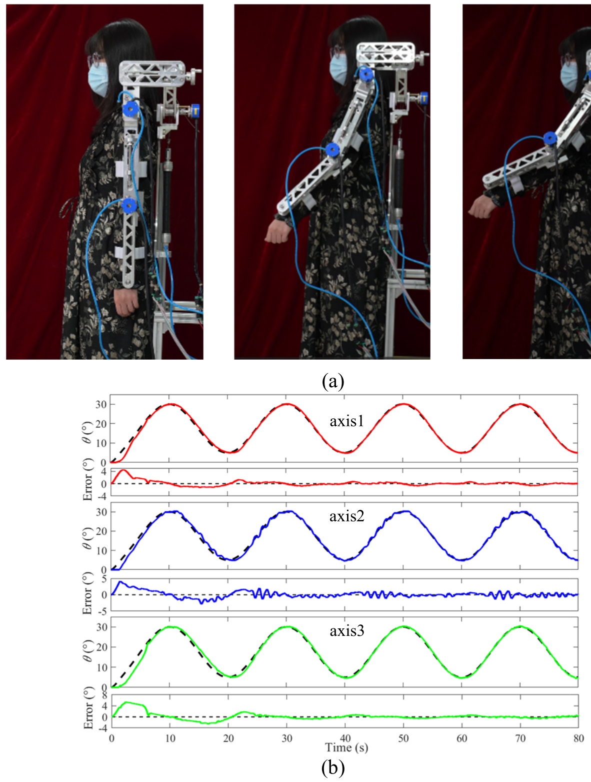


Fig. 15. Human participant trial result: (a) snapshots of the experiment, and (b) the trajectory tracking performance.

where, x, y, and z are the coordinates of the actual trajectories on the X, Y, and Z axes, respectively, and x_d, y_d, and z_d are the coordinates of the desired trajectories on the X, Y, and Z axes, respectively.

The tracking errors of the controllers are shown in Fig. 14(b). It can be found that the proposed method achieves the fastest convergence and smallest tracking error among the controllers.

E. Human Participant Trial

After validating the trajectory tracking performance and robustness of the proposed method, it is important to assess the real-world performance of the exoskeleton robot when worn by subjects. In this section, a healthy participant wears the exoskeleton robot, with the subject's left arm secured using elastic straps and the exoskeleton robot fixed to the base, as shown in Fig. 15(a). Prior to the execution of the experimental protocol, the subject received detailed explanations about the potential risks of the experiment and signed informed consent forms before participating in this study. Additionally, the subject has authorized us to use her personal information and experimental results.

We use the proposed method to drive three axes to track a 0.05 Hz sinusoidal trajectory. The amplitude and offset of the sinusoidal trajectory are set at 12.5° and 17.5°, respectively. The experimental results are shown in Fig. 15(b), and a video showcasing the actual performance of the rehabilitation exoskeleton worn by healthy participants is included in supplementary Movie 1. Compared with the experimental results on human phantom, it is observed that during the motion of the subject's arm driven by the exoskeleton robot, the tracking error is slightly larger in the first 10 seconds. No overshooting phenomenon is observed, ensuring the safety of the startup process. This is reasonable because the uncertainties of the subject introduce extra disturbances to the system. After 1 iteration, the error of the proposed method gradually decreases, showing excellent iterative learning capability and steady-state performance. This experimental result further demonstrates the effectiveness and applicability of the exoskeleton robot.

V. Conclusion

In this paper, a PAM-actuated upper limb rehabilitation exoskeleton robot is developed to assist patients with rotator cuff injuries. An active neural network hysteresis compensation method is proposed for the exoskeleton robot. In this method, a NN is utilized to compensate for the hysteresis nonlinearity of the system, and the weights of the NN are estimated and updated in real time using UKF. Additionally, the approximation error of the NN is considered and jointly estimated as an extended state in UKF, which significantly improves compensation accuracy. The output of the NN and the estimated approximation error are integrated as the control voltage applied to the exoskeleton robot. This approach completely eliminates the need for model inversion and offline parameter identification, thus enhancing ease of use.

Different sets of experiments have been carried out on the exoskeleton robot. Experimental results show that the proposed method can effectively compensate for the PAM's hysteresis and improve tracking performance on both slow and fast trajectories. Regarding external disturbances, the proposed method achieves the best robustness among the comparative methods. Spatial 3D trajectory tracking was also conducted to verify the capability of the developed exoskeleton robot in providing complex rehabilitation motions. In all the aforementioned experimental verifications, the tracking accuracy of the proposed method can be gradually improved through iterative learning and updating. This capability is particularly valuable for the iterative motions involved in rehabilitation training. Future work will focus on the clinical applications and testing of the developed exoskeleton robot.

References

[1] T. Ahmed, M. R. Islam, B. Brahmi, and M. H. Rahman, "Robustness and tracking performance evaluation of PID motion control of 7 DoF anthropomorphic exoskeleton robot assisted upper limb rehabilitation," *Sensors*, vol. 22, no. 10, p. 3747, May 2022.

[2] S. H. Chen et al., "Assistive control system for upper limb rehabilitation robot," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 11, pp. 1199–1209, Nov. 2016.

[3] B. O. Mushage, J. C. Chedjou, and K. Kyamakya, "Fuzzy neural network and observer-based fault-tolerant adaptive nonlinear control of uncertain 5-DOF upper-limb exoskeleton robot for passive rehabilitation," *Nonlinear Dyn.*, vol. 87, no. 3, pp. 2021–2037, Feb. 2017.

[4] A. Razzaghian, "A fuzzy neural network-based fractional-order Lyapunov-based robust control strategy for exoskeleton robots: Application in upper-limb rehabilitation," *Math. Comput. Simul.*, vol. 193, pp. 567–583, Mar. 2022.

[5] A. Riani, T. Madani, A. Benallegue, and K. Djouani, "Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton," *Control Eng. Pract.*, vol. 75, pp. 108–117, Jun. 2018.

[6] P. Beyl, M. Van Damme, R. Van Ham, B. Vanderborght, and D. Lefeber, "Pleated pneumatic artificial muscle-based actuator system as a torque source for compliant lower limb exoskeletons," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 3, pp. 1046–1056, Jun. 2014.

[7] G. Andrikopoulos, G. Nikolakopoulos, I. Arvanitakis, and S. Manesis, "Piecewise affine modeling and constrained optimal control for a pneumatic artificial muscle," *IEEE Trans. Ind. Electron.*, vol. 61, no. 2, pp. 904–916, Feb. 2014.

[8] D. Liang, N. Sun, Y. Wu, G. Liu, and Y. Fang, "Fuzzy-sliding mode control for humanoid arm robots actuated by pneumatic artificial muscles with unidirectional inputs, saturations, and dead zones," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3011–3021, May 2022.

[9] T. Nuchkrua and T. Leephakpreeda, "Novel compliant control of a pneumatic artificial muscle driven by hydrogen pressure under a varying environment," *IEEE Trans. Ind. Electron.*, vol. 69, no. 7, pp. 7120–7129, Jul. 2022.

[10] R. M. Robinson, C. S. Kothera, R. M. Sanner, and N. M. Wereley, "Nonlinear control of robotic manipulators driven by pneumatic artificial muscles," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 1, pp. 55–68, Feb. 2016.

[11] Q. Wang, T. Yang, G. Liu, Y. Qin, Y. Fang, and N. Sun, "Adaptive compensation tracking control for parallel robots actuated by pneumatic artificial muscles with error constraints," *IEEE Trans. Ind. Informat.*, vol. 20, no. 2, pp. 1585–1595, Feb. 2024.

[12] Y. Qin, H. Zhang, X. Wang, N. Sun, and J. Han, "Adaptive set-membership filter based discrete sliding mode control for pneumatic artificial muscle systems with hardware experiments," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1682–1694, Apr. 2004.

[13] W. Liu, L. Cheng, Z.-G. Hou, J. Yu, and M. Tan, "An inversion-free predictive controller for piezoelectric actuators based on a dynamic linearized neural network model," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 1, pp. 214–226, Feb. 2016.

[14] M. Rakotondrabe, "Bouc-Wen modeling and inverse multiplicative structure to compensate hysteresis nonlinearity in piezoelectric actuators," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 428–431, Apr. 2011.

[15] Y. Qin, Y. Tian, D. Zhang, B. Shirinzadeh, and S. Fatikow, "A novel direct inverse modeling approach for hysteresis compensation of piezoelectric actuator in feedforward applications," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 3, pp. 981–989, Jun. 2013.

[16] Y. Qin, Y. Zhang, H. Duan, and J. Han, "High-bandwidth hysteresis compensation of piezoelectric actuators via multilayer feedforward neural network based inverse hysteresis modeling," *Micromachines*, vol. 12, no. 11, p. 1325, Oct. 2021.

[17] D. Zhang, X. Zhao, and J. Han, "Active model-based control for pneumatic artificial muscle," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1686–1695, Feb. 2017.

[18] J. Escareno, J. Abadie, E. Piat, and M. Rakotondrabe, "Robust micro-positionnig control of a 2DOF piezocantilever based on an extended-state LKF," *Mechatronics*, vol. 58, pp. 82–92, Apr. 2019.

[19] Z. Deng, H. Wang, and R. Wang, "An improved unscented Kalman filter for interrupted and drift sensor faults of aircrafts," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[20] L. Bergmann et al., "Lower limb exoskeleton with compliant actuators: Design, modeling, and human torque estimation," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 2, pp. 758–769, Apr. 2023.

[21] J. Kim, D. Lee, B. Kiss, and D. Kim, "An adaptive unscented Kalman filter with selective scaling (AUKF-SS) for overhead cranes," *IEEE Trans. Ind. Electron.*, vol. 68, no. 7, pp. 6131–6140, Jul. 2021.

[22] Y. Chen, N. Sun, D. Liang, Y. Qin, and Y. Fang, "A neuroadaptive control method for pneumatic artificial muscle systems with hardware experiments," *Mech. Syst. Signal Process.*, vol. 146, Jan. 2021, Art. no. 106976.

[23] Y. Qing, J. Fan, H. Zhang, and J. Han, "Design and control of a pneumatic artificial muscle actuated exoskeleton robot for upper limb rehabilitation," *J. Mech. Eng.*, vol. 60, no. 11, Jun. 2024.