Received: 13 March 2025 | Revised: 29 April 2025 | Accepted: 19 May 2025

DOI: 10.1002/asjc.3741

RESEARCH ARTICLE

WILEY

Wearable exoskeleton robot control using radial basis function-based fixed-time terminal sliding mode with prescribed performance

Mahmoud Abdallah[1] | **Yassine Kali**[2] | **Maarouf Saad**[1] | **Raouf Fareh**[3] | **Maamar Bettayeb**[3]

[1]Department of Electrical Engineering, École de technologie supérieur, Montreal, Quebec, Canada

[2]School of Engineering, Université du Québec en Abitibi-Témiscamingue, Rouyn-Noranda, Quebec, Canada

[3]Department of Electrical Engineering, University of Sharjah, Sharjah, United Arab Emirates

Correspondence
Mahmoud Abdallah, Department of Electrical Engineering, École de technologie supérieur, Montreal, QC, Canada.
Email: mahmoud-a-y.abdallah.1@ens.etsmtl.ca

Abstract

This paper tackles the problem of robust and accurate fixed-time tracking in human–robot interaction and deals with uncertainties. This work introduces a control approach for a wearable exoskeleton designed specifically for rehabilitation tasks. The approach combines prescribed performance control-based fixed-time terminal sliding mode with a neural network. Its main objectives are to achieve trajectory tracking, reduce chattering, ensure fixed-time stability, and maintain robustness against uncertainties. The controller includes a radial basis function neural network to estimate unknown dynamics and incorporates prescribed performance criteria. This enables precise joint space trajectory tracking, even in the presence of uncertain dynamics and disturbances. The prescribed performance ensures that the trajectory tracking error evolves within prescribed limits. The combined neural network and fixed-time terminal sliding mode technique are proposed to ensure robustness and fixed-time convergence. The closed-loop stability is analyzed using the Lyapunov theory, and a new fixed-time convergence is provided. Numerical simulations demonstrate a reduction tracking compared to another advanced SMC technique, while experimental results on a 7-DoF ETS-MARSE exoskeleton show better tracking with control torques free of chattering compared with two advanced SMC techniques.

KEYWORDS

exoskeleton, prescribed performance, RBF, rehabilitation, sliding mode control, tracking control

1 | INTRODUCTION

Rehabilitation treatments are now widely recognized as essential post-surgery exercises for individuals recovering from neurological injuries such as strokes, trauma, or paralysis [1]. Physiotherapists typically oversee these physical rehabilitation tasks for patients with disabilities. However, the global rise in limb impairments requires external support. Robotic systems have gained traction in the medical field, with exoskeleton robots being integrated into physiotherapy as intelligent assistive solutions for enhanced rehabilitation [2]. Wearable exoskeleton robots are advanced robotic systems designed to be worn by individuals to assist in movement and physical rehabilitation [3]. These robots are engineered to support and augment the movements of the upper limb of the user, providing

precise control and support that mimics natural functions of the limb such as in [4]. The NEUROExos study in [5] demonstrates real-world validation through human trials, assessing the effectiveness of an elbow exoskeleton for physical rehabilitation. Wearable exoskeletons, such as the ETS-MARSE system discussed in this paper, are particularly valuable in rehabilitation, as they help patients regain mobility and strength through guided exercises. The wearable nature of these exoskeletons ensures that patients can comfortably use them, allowing for prolonged and effective therapy sessions. The design requirements of an exoskeleton—such as range of motion, compactness, comfort, and safety—directly influence the selection of control algorithms. For instance, achieving a natural range of motion necessitates control strategies that can accurately replicate complex movements, while ensuring user comfort and safety may require adaptive algorithms capable of responding to real-time biomechanical feedback [6].

In recent years, extensive research has been conducted on the development and application of various control methods to track the control of exoskeleton robot systems. The key motivation behind these studies is to improve the performance and robustness of exoskeletons, ensuring precise and reliable support during rehabilitation. However, many advanced control strategies face the challenge of high computational cost, which hinders their real-time applicability, especially in complex systems with high degrees of freedom. One widely studied control method is proportional–derivative (PD) control, which has been effectively used in several applications due to its simplicity and ease of implementation [7]. However, PD control often struggles with nonlinearities and uncertainties in exoskeleton dynamics, prompting the exploration of more advanced strategies. Adaptive control methods have also gained traction, as they can adjust control parameters in real time to accommodate variations in system dynamics [8, 9]. For example, adaptive control has been successfully implemented to improve the accuracy and adaptability of exoskeletons under different operating conditions [10]. Sliding mode control (SMC) is another popular approach, known for its robustness against disturbances and uncertainties. SMC has been applied to exoskeleton robots to achieve precise trajectory tracking, despite the inherent nonlinearities of the system [11–13]. SMC variations, such as the terminal SMC (TSMC), have been developed to improve the convergence speed and control precision [14, 15]. Fuzzy logic control (FLC) and neural network control (NNC) are intelligent control methods that have shown promise in handling the complexities of exoskeleton dynamics. These methods leverage the ability of fuzzy systems and neural networks to approximate nonlinear functions and adapt to changing conditions [9, 16,17]. Model predictive control (MPC) has also been explored for exoskeleton trajectory tracking, offering the advantage of optimizing control actions over a finite-time horizon while considering system constraints [18, 19]. MPC has effectively achieved smooth and accurate movement trajectories for exoskeleton robots [19]. Additionally, robust control techniques, such as H-infinity control, have been applied to ensure the stability and performance of exoskeleton systems under uncertain conditions [20]. These methods provide a systematic way to handle model uncertainties and external disturbances, ensuring reliable trajectory tracking. Recent studies have also investigated the use of hybrid control approaches, combining different control strategies to leverage their respective strengths. For instance, combining SMC with adaptive control or fuzzy logic has shown improved performance in trajectory tracking tasks [16, 17].

Among these strategies, SMC stands out as one of the most prevalent for uncertain robots [21–23]. The fundamental objective of SMC is to guide the dynamics of the nonlinear system to slide along a surface that eventually converges to an equilibrium point. This characteristic contributes to SMC's ability to reduce the system's complexity [24, 25]. Nevertheless, SMC is not without its challenges, notably the well-known issue of chattering phenomena.

The chattering phenomenon can destabilize the system, elevate the input signal frequency, and potentially damage the hardware [26]. To address the chattering problem effectively, multiple enhancements to SMC have been introduced, including second-order SMC [27], adaptive SMC [28], fractional-order SMC (FOSMC) [29], exponential-reaching SMC [30], and TSMC [31]. Second-order SMC improves upon traditional SMC by adding second-order dynamics to the control law [32]. Instead of applying a discontinuous control directly, it employs continuous control with these dynamics to guide the system's state. This reduces chattering and results in smoother control. However, second-order SMC requires some unavailable information for measurements and suffers from complexity in designing and tuning, high computational time, and sensitivity to model mismatches and internal uncertainties [33]. Adaptive SMC also suffers from high tuning and computational costs [21]. FOSMC demonstrates its capability to mitigate chattering in complex systems by incorporating fractional calculus within the formulation of the sliding manifold. Nevertheless, the utilization of fractional calculus introduces intricacies in the system's stability analysis, resulting in a more complex and nuanced process when it comes to design and tuning [34]. In [30], the solution of using an exponential-reaching SMC has been introduced to reduce chattering and maintain the quality of the tracking performance of nonlinear systems. The work uses an exponential-reaching law function that adapts to dynamic changes in the reaching

phase. However, the exponential-reaching-based SMC has increased the number of tuning parameters with a certain condition to meet, which leads to an increase in the computational cost.

Over the past few years, fast TSMC [35] has emerged as a more advanced version of the previous SMC techniques. Fast TSMC aims to guarantee finite-time stability, rapid convergence, rejection of the model uncertainties and disturbances, and precise tracking [36]. Fast TSMC may have a singularity issue, but this can be resolved by using a non-singular terminal sliding mode (NTSMC). NTSMC solves the singularity problem that arises in the terminal sliding mode while still ensuring the finite-time convergence [37]. Fast TSMC has been applied to various robotic systems, including exoskeletons and serial-link manipulators [38–40]. The simulation results of the studies mentioned above demonstrate the effectiveness of fast TSMC. However, the heavy dynamic model of a high-degree-of-freedom robotic exoskeleton restricts the applicability of fast TSMC to simulation only. Therefore, addressing this issue and finding ways to overcome it is crucial.

An approach to enhance the effectiveness of the traditional SMC is to incorporate prescribed performance functions that modify the tracking error of the system. The goal of these functions is to reformulate the dynamic error of the system to guarantee convergence and enhance transient behaviors and steady-state accuracy [41]. Many previous works utilized different prescribed performance expressions to improve SMC in manipulator tracking [42–44].

In order to tackle the issue of high computational cost and complexity in dynamic models, various types of neural networks can be utilized to approximate their nonlinear behavior. Among these, radial basis function (RBF) networks [45] are particularly useful in controlling robotic manipulators. RBF networks are a type of neural network that excels in function approximation and interpolation. They are widely used in tasks such as trajectory generation [46], inverse kinematics [47], and adaptive control [48], due to their ability to model complex nonlinear dynamics. By improving the precision and efficiency of robotic manipulators, RBF networks enable them to accomplish complex tasks with greater accuracy and agility [49–51]. Therefore, RBF networks play a crucial role in advancing robotics, allowing for the development of more sophisticated and capable robotic systems.

The focus of this paper is to introduce an RBF neural network-based fixed-time SMC (FTSMC) that is designed to provide prescribed performance for uncertain robotic systems for upper-limb rehabilitation. The FTSMC offers several advantages in the context of human–robot interaction for a seven-degree-of-freedom (7-DoF) exoskeleton. It ensures fixed-time stability within a predefined time regardless of initial conditions. Additionally, the FTSMC provides robustness against uncertainties and external disturbances, making it well suited for dynamic environments. The integration of an RBF neural network helps reduce chattering, leading to smoother control actions that enhance user comfort and prolong hardware lifespan. Furthermore, the FTSMC achieves high tracking accuracy, keeping errors within strict bounds for precise and stable movements. Importantly, it maintains a balance between robustness and computational efficiency, ensuring real-time feasibility for several robotics applications.

Indeed, the proposed method will be tested on the 7-DoF ETS-MARSE in real time. The motivation behind developing this controller is driven by the need to tackle various challenges associated with existing SMC techniques. These challenges include unknown dynamics, external disturbances, uncertainties, and the substantial complexities arising from the heavy dynamics of a 7-DoF ETS-MARSE model. These issues, combined with the limitations of earlier approaches, underscore the necessity for a more robust solution. The proposed method aims to overcome these challenges by utilizing an RBF neural network to estimate the ETS-MARSE dynamic model efficiently. The first step in designing the approach is to define the prescribed performance function, which is responsible for reforming the tracking error of each joint of the robot, ensuring that it remains within a narrow band around the equilibrium point. The sliding surface then will be a function of the newly formed tracking error. Then, an RBF model will estimate the nonlinearities of the robot, including centrifugal and Coriolis forces, gravity force, motor frictions, and uncertainties. The RBF model takes the robot's states as input and outputs an estimated function that is added to the mathematical expression to better model nonlinearities. Having the optimal prescribed performance and the estimated function of nonlinearities will form the control scheme of the fixed-time TSMC. Lyapunov stability analysis will be used to ensure the stability and the fixed-time convergence of the proposed performance taking into consideration the limits of the tuning parameters and the hyperparameters of the RBF.

The contributions of this paper could be summarized as follows:

- Propose a switching terminal sliding surface with prescribed performance that ensures a new fixed-time convergence.
- Inspiring from the proposed surface, a new fixed-time reaching law is proposed to drive the system's trajectories to the designed sliding surface.
- Combine the proposed fixed-time technique with the RBF that estimates the dynamics to reject the effect of

the uncertainties and to reduce the complexity computations due to the high number of DoF.
- Implement the proposed control scheme on the 7-DoF ETS-MARSE experimentally and use it for the application of rehabilitation. To the best of the authors' knowledge, this has not been introduced in the literature.

The following sections of this paper are outlined after this introductory section: Section 2 presents the wearable robotic exoskeleton model and formulates the problem. Section 3 discusses the proposed controller design procedure and the stability analysis. Section 4 presents case studies where the controller is compared to two other advanced techniques in simulation and is implemented in real time on a rehabilitation prototype robot, and Section 5 concludes the paper.

2 | DYNAMIC MODEL

ETS-MARSE has three modes of operation: passive, active, and assistive rehabilitation. In the passive mode, the robot functions like a standard manipulator, with the human upper limb being treated as an external disturbance. In active rehabilitation, the patient's arm generates a force that is applied to the end-effector of the ETS-MARSE. The assistive mode is the same as the active mode, but the robot assists the patient by predicting his intention. In all modes, the dynamic model of the ETS-MARSE is:

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + F(\dot{\theta}) - \tau_p = \tau, \tag{1}$$

where $\theta \in \mathbb{R}^7$ denotes the joint space position vector; $D(\theta) \in \mathbb{R}^{7\times7}$ denotes the bounded and invertible inertia matrix; $C(\theta,\dot{\theta})\dot{\theta} \in \mathbb{R}^7$ is the Coriolis and centrifugal forces vector; $G(\theta)$ and $F(\dot{\theta})$ are, respectively, the (7×1) gravitational and friction vectors; $\tau_p \in \mathbb{R}^7$ is the vector of matched disturbances and unknown patient–robot interaction forces; and $\tau \in \mathbb{R}^7$ is the vector of torque inputs. In this paper, we assume that τ_p is bounded such as

$$\|\tau_p\| \leq \tau_p^{max}, \tag{2}$$

where τ_p^{max} is a known positive constant.

TABLE 1 Modified DH parameters of the studied robot.

Joint (i)	a_{i-1} (m)	d_i (m)	α_{i-1} (°)	θ_i (°)
1	0	d_s	0	θ_1
2	0	0	−90	θ_2
3	0	d_e	90	θ_3
4	0	0	−90	θ_4
5	0	d_w	90	θ_5
6	0	0	−90	$\theta_6 - 90$
7	0	0	−90	θ_7

Furthermore, the robot dynamics in (1) can be rewritten as

$$\ddot{\theta} = D^{-1}(\theta)\left[N(\theta,\dot{\theta}) + \tau_p + \tau\right], \tag{3}$$

where $N(\theta,\dot{\theta}) = -C(\theta,\dot{\theta})\dot{\theta} - G(\theta) - F(\dot{\theta})$.

Equation (1) represents the dynamic model of the ETS-MARSE exoskeleton, formulated using the standard Lagrangian approach to account for inertia, Coriolis and centrifugal forces, gravity, friction, and external disturbances. This model serves as the foundation for controller design, enabling the proposed FTSMC to achieve precise trajectory tracking and robustness against uncertainties. Additionally, it explicitly incorporates unknown human–robot interaction forces, which are essential in rehabilitation robotics, where human effort varies dynamically. Because certain dynamic terms, such as friction and patient–robot interactions, are challenging to model accurately, an RBF neural network is employed in Section 3 to approximate these unknown dynamics, enhancing real-time implementation. Moreover, the model plays a crucial role in ensuring stability and fixed-time convergence, as the Lyapunov-based stability proof and prescribed performance function are formulated directly from it, ensuring theoretical rigor in the control approach.

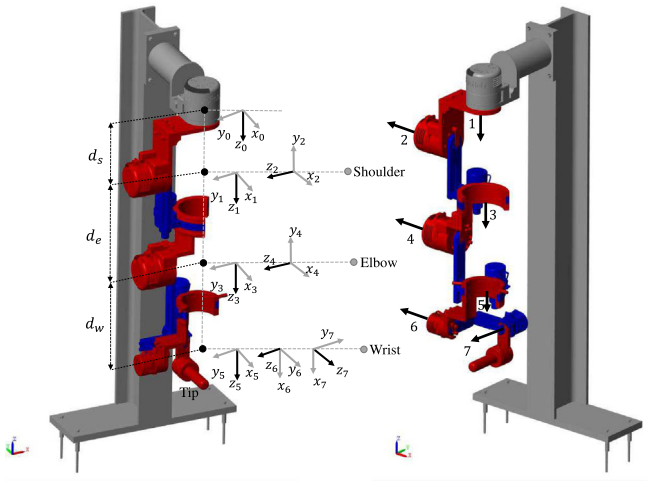


FIGURE 1 ETS-MARSE links frame. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Key specifications of the ETS-MARSE exoskeleton robot.

Degrees of freedom (DoF)	7
Actuation type	Electric motors (Maxon brushless)
Range of motion (per joint)	± 45° to ± 120° (varies by joint)
Maximum torque output	9 to 40 Nm (varies by joint)
Weight	10.542 kg
Sensor type	Encoders, force sensors

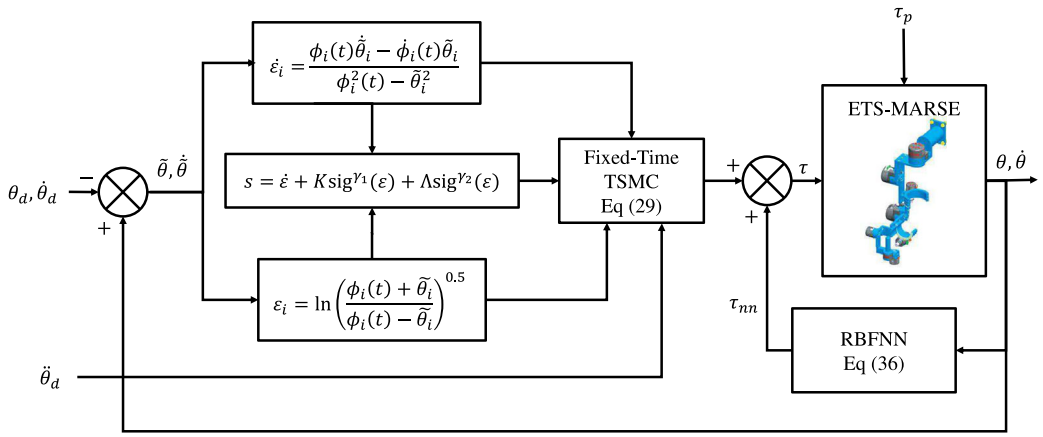


FIGURE 2 Control scheme block diagram. [Color figure can be viewed at wileyonlinelibrary.com]

Table 1 shows the DH parameters of the ETS-MARSE, Figure 1 shows the ETS-MARSE links frame Simscape model using MATLAB Simscape toolbox, and Table 2 summarizes the key specifications of the ETS-MARSE exoskeleton, detailing its degrees of freedom, actuation type, range of motion, and other essential parameters.

3 | THE CONTROL SCHEME

The control scheme is developed into two stages. The first stage is to develop the prescribed performance fixed-time TSMC. The second stage is to exploit the RBF neural network to approximate the uncertain nonlinearities of the studied robot. The control scheme block diagram is depicted in Figure 2.

The following key assumptions were made in the design of the proposed controller:

1. The robot dynamics are modeled accurately using Equation (3), and the inertia matrix $D(\theta)$ is bounded and invertible.
2. The external disturbances and uncertainties, represented by τ_p, are bounded as stated in Equation (2).
3. The tracking error is initially within the bounds defined by the prescribed performance function, ensuring that the sliding mode surface can be reached.
4. The RBF neural network can approximate the nonlinear dynamics of the system with negligible estimation error under sufficient training.

3.1 | Prescribed performance FTSMC

This section proposes a switching manifold that guarantees a fixed-time convergence and a good tracking performance using the following prescribed performance function [52]. The function is illustrated in Figure 3 and is

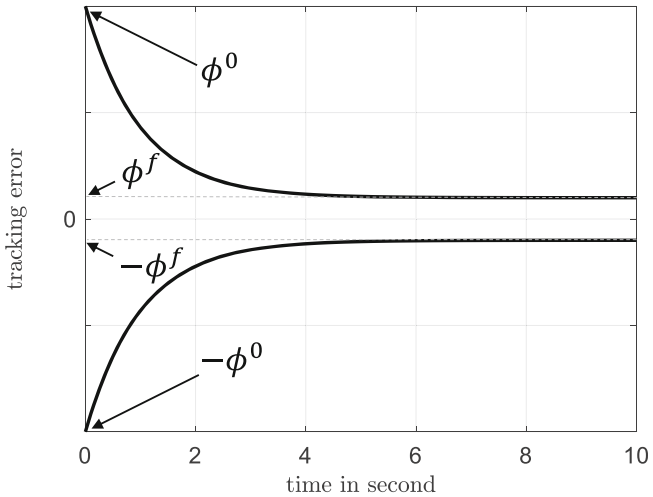


FIGURE 3 Prescribed function.

defined for $i = 1, \cdots, 7$ by

$$\phi_i(t) = \left(\phi_i^0 - \phi_i^f\right)\exp(-\alpha_i t) + \phi_i^f, \tag{4}$$

where $\phi_i^0 > \phi_i^f > 0$ and $\alpha_i > 0$. Because $\phi_i(t)$ is a positive-definite function while its first-time derivative is a negative definite one $\forall t > 0$, $\phi_i(t)$ is a decreasing one. Moreover, it can be observed that $\phi_i(t)$ converges to ϕ_i^f when the time t converges to ∞. The above function is used to limit the tracking error as follows:

$$-\phi_i(t) < \tilde{\theta}_i = \theta_i - \theta_i^d < \phi_i(t). \tag{5}$$

Let us introduce a new variable, denoted as ψ_i:

$$\tilde{\theta}_i = \phi_i(t)\psi_i, \tag{6}$$

where

$$\psi_i = \frac{1 - \exp(-2\varepsilon_i)}{1 + \exp(-2\varepsilon_i)}, \tag{7}$$

where ε_i is the tracking error transformation:

$$\varepsilon_i = 0.5\left(\ln\left(\tilde{\theta}_i + \phi_i(t)\right) - \ln\left(\phi_i(t) - \tilde{\theta}_i\right)\right). \tag{8}$$

The first-time derivative of (8) gives

$$\dot{\varepsilon}_i = \frac{\phi_i(t)\dot{\tilde{\theta}}_i - \dot{\phi}_i(t)\tilde{\theta}_i}{\phi_i^2(t) - \tilde{\theta}_i^2}. \tag{9}$$

Based on (8) and (9), the following fixed-time terminal sliding surface is proposed [53]:

$$s = \dot{\varepsilon} + K\text{sig}^{\gamma_1}(\varepsilon) + \Lambda\text{sig}^{\gamma_2}(\varepsilon), \tag{10}$$

where K and Λ are two diagonal positive-definite matrices, $\text{sig}^{\gamma_j}(\varepsilon) = \left[|\varepsilon_1|^{\gamma_{j1}}\text{sign}(\varepsilon_1), \ldots, |\varepsilon_7|^{\gamma_{j7}}\text{sign}(\varepsilon_7)\right]^T$ with γ_{1i} and γ_{2i} are defined for $i = 1, \ldots, 7$ by

$$\gamma_{1i} = \begin{cases} \gamma_{1i}^*, & \text{if } |\varepsilon_i| > 1, \\ 1, & \text{if } |\varepsilon_i| \le 1, \end{cases} \tag{11}$$

$$\gamma_{2i} = \begin{cases} 1, & \text{if } |\varepsilon_i| > 1, \\ \gamma_{2i}^*, & \text{if } |\varepsilon_i| \le 1, \end{cases} \tag{12}$$

where $\gamma_{1i}^* > 1, 0 < \gamma_{2i}^* < 1$. These parameters are tuned in the real-time experiment section based on these limitations. Also,

$$\text{sign}(\varepsilon_i) = \begin{cases} -1, & \text{if } \varepsilon_i < 0, \\ 0, & \text{if } \varepsilon_i = 0, \\ 1, & \text{if } \varepsilon_i > 0, \end{cases} \tag{13}$$

It is worth mentioning that the proposed surface, even when designed piecewise, ensures continuity at the switching point [53].

Theorem 1. *Consider the proposed nonlinear sliding surface with prescribed performance in (10), the exoskeleton robot (1) is practical fixed-time stable at $\tilde{\theta} = 0$. For each joint, the maximal convergence time is given by*

$$\begin{aligned} T_i =& T_{1i} + T_{2i}, \\ T_{2i} =& \frac{1}{\underline{k}_i}\ln\left(1 + \frac{k_i}{\lambda_i}\right) + \frac{1}{\lambda_i(\gamma_{1i}^* - 1)}\ln\left(1 + \frac{\lambda_i}{k_i}\right), \end{aligned} \tag{14}$$

where T_{1i} is the required time to the system trajectory to converge to the designed fixed-time switching manifold s_i such as $s_i(T_{1i}) = 0$ and $\underline{k}_i = (1 - \gamma_{2i}^*)k_i$.

Proof. Let us take a look at the following Lyapunov function and its first-time derivative:

$$\begin{aligned} V_1 =& 0.5\varepsilon^T\varepsilon, \\ \dot{V}_1 =& \varepsilon^T\dot{\varepsilon}. \end{aligned} \tag{15}$$

Once the system's trajectories reach the sliding surface, then

$$\dot{\varepsilon} = -K\text{sig}^{\gamma_1}(\varepsilon) - \Lambda\text{sig}^{\gamma_2}(\varepsilon). \tag{16}$$

Substituting (16) into $\dot{V}_1$ gives

$$\begin{aligned} \dot{V}_1 &= \varepsilon^T\left(-K\text{sig}^{\gamma_1}(\varepsilon) - \Lambda\text{sig}^{\gamma_2}(\varepsilon)\right) \\ &= -\sum_{i=1}^{7}\left(k_i|\varepsilon_i|^{1+\gamma_{1i}} + \lambda_i|\varepsilon_i|^{1+\gamma_{2i}}\right) \le 0 \\ &= -\sum_{i=1}^{7} k_i(2V_{1i})^{0.5(\gamma_{1i}+1)} + \lambda_i(2V_{1i})^{0.5(\gamma_{2i}+1)} \le 0. \end{aligned} \tag{17}$$

Moreover, let us rewrite (17) for $i = 1, \ldots, 7$ as follows:

$$\dot{V}_{1i} = \begin{cases} -k_i(2V_{1i})^{0.5(\gamma_{1i}^*+1)} - 2\lambda_i V_{1i}, & \text{if } V_{1i} > 1/2, \\ -2k_i V_{1i} - \lambda_i(2V_{1i})^{0.5(\gamma_{2i}^*+1)}, & \text{if } V_{1i} \le 1/2. \end{cases} \tag{18}$$

Introducing a new variable η such as $\eta = \ln(2V_{1i}) + 1$ if $2V_{1i} > 1$ and $\eta = (2V_{1i})^{0.5(1-\gamma_{2i}^*)}$ if $2V_{1i} \le 1$ such as (18) can be rewritten as

$$\dot{\eta} = \begin{cases} -2k_i\exp(0.5(\eta - 1)(\gamma_{1i}^* - 1)) - 2\lambda_i, & \text{if } \eta > 1, \\ -\underline{k}_i\eta - \underline{\lambda}_i, & \text{if } 0 < \eta \le 1, \end{cases} \tag{19}$$

where $\underline{k}_i = (1 - \gamma_{2i}^*)k_i$ and $\underline{\lambda}_i = (1 - \gamma_{2i}^*)\lambda_i$. Hence, the maximal convergence time is computed by solving the above equation as follows:

$$T_i = T_{1i} + T_{2i}, \tag{20}$$

where T_{2i} is the solution of (19) when $|\varepsilon_i| > 1$ and $|\varepsilon_i| \le 1$ that are computed as

$$\begin{aligned} T_{2i} =& \frac{1}{\underline{k}_i}\int_0^1 \frac{1}{\eta + \frac{\underline{\lambda}_i}{\underline{k}_i}}d\eta \\ &+ \frac{1}{2k_i}\int_1^\infty \frac{1}{\exp(0.5(\eta - 1)(\gamma_{1i}^* - 1)) + \frac{\lambda_i}{k_i}}d\eta \\ =& \frac{1}{\underline{k}_i}\ln\left(1 + \frac{\underline{k}_i}{\underline{\lambda}_i}\right) + \frac{1}{\lambda_i(\gamma_{1i}^* - 1)}\ln\left(1 + \frac{\lambda_i}{k_i}\right). \end{aligned} \tag{21}$$

Hence, the proposed sliding surface ensures fixed-time stability.

This completes the proof. □

Now, let us compute the derivative of s using (9), (10), and the robot's known dynamics in (3):

$$\begin{aligned}\dot{s} =&\ddot{\varepsilon} + K\gamma_1 \lfloor \varepsilon \rceil^{\gamma_1 - 1}\dot{\varepsilon} + \Lambda\gamma_2 \lfloor \varepsilon \rceil^{\gamma_2 - 1}\dot{\varepsilon} \\ =&R\Phi\ddot{\tilde{\theta}} - R\ddot{\Phi}\tilde{\theta} + \dot{R}\left(\Phi\dot{\tilde{\theta}} - \dot{\Phi}\tilde{\theta}\right) \\ &+ K\gamma_1 \lfloor \varepsilon \rceil^{\gamma_1 - 1}\dot{\varepsilon} + \Lambda\gamma_2 \lfloor \varepsilon \rceil^{\gamma_2 - 1}\dot{\varepsilon} \\ =&R\Phi D^{-1}(\theta)\tau - R\Phi\ddot{\theta}^d + \dot{R}\left(\Phi\dot{\tilde{\theta}} - \dot{\Phi}\tilde{\theta}\right) \\ &- R\ddot{\Phi}\tilde{\theta} + K\gamma_1 \lfloor \varepsilon \rceil^{\gamma_1 - 1}\dot{\varepsilon} + \Lambda\gamma_2 \lfloor \varepsilon \rceil^{\gamma_2 - 1}\dot{\varepsilon},\end{aligned} \tag{22}$$

where

- $R = \text{diag}\left(\frac{1}{\phi_i^2(t) - \tilde{\theta}_i^2}\right)$, for $i = 1, \ldots, 7$;
- $\dot{R}$ is the first derivative of R;
- $\Phi = \text{diag}(\phi_1, \cdots, \phi_7)$, $\dot{\Phi} = \text{diag}(\dot{\phi}_1, \cdots, \dot{\phi}_7)$, $\ddot{\Phi} = \text{diag}(\ddot{\phi}_1, \cdots, \ddot{\phi}_7)$ with $\dot{\phi}_i$ and $\ddot{\phi}_i$ are, respectively, the first and second time derivatives of ϕ_i;
- $\ddot{\theta}^d = \left[\ddot{\theta}_1^d, \cdots, \ddot{\theta}_7^d\right]^T$ is the desired acceleration vector;
- $\lfloor \varepsilon \rceil^{\gamma_j - I} = \text{diag}\left(|\varepsilon_1|^{\gamma_{j1}-1}, |\varepsilon_2|^{\gamma_{j2}-1}, \cdots, |\varepsilon_7|^{\gamma_{j7}-1}\right)$ for $j = 1, 2$ with I is the identity matrix;
- $\gamma_j = \text{diag}(\gamma_{j1}, \cdots, \gamma_{j7})$ for $j = 1, 2$.

The equation mentioned above is used to calculate the equivalent control by solving for $\dot{s}$ when it is equal to zero.

$$\tau_{eq} = D(\theta)\ddot{\theta}^d - D(\theta)\Phi^{-1}R^{-1}\tau_0, \tag{23}$$

where:

$$\tau_0 = \dot{R}\left(\Phi\dot{\tilde{\theta}} - \dot{\Phi}\tilde{\theta}\right) - R\ddot{\Phi}\tilde{\theta} + K\gamma_1 \lfloor \varepsilon \rceil^{\gamma_1 - I}\dot{\varepsilon} + \Lambda\gamma_2 \lfloor \varepsilon \rceil^{\gamma_2 - I}\dot{\varepsilon}. \tag{24}$$

Remark 1. During real-time implementation, the term $|\varepsilon_i|^{\gamma_{2i}-1}\dot{\varepsilon}_i$ may become unbounded as $\dot{\varepsilon}_i \neq 0$ and $\varepsilon_i \to 0$, due to the exponent $\gamma_{2i} - 1 < 0$. To limit its amplitude, the saturation function (sat) is used, and the term $\lfloor \varepsilon \rceil^{\gamma_2 - I}\dot{\varepsilon}$ is substituted by $\text{sat}\left(\lfloor \varepsilon \rceil^{\gamma_2 - I}\dot{\varepsilon}, \mu\right) = \left[\text{sat}\left(|\varepsilon_1|^{\gamma_{21}-1}\dot{\varepsilon}_1, \mu_1\right), \cdots, \text{sat}\left(|\varepsilon_7|^{\gamma_{27}-1}\dot{\varepsilon}_7, \mu_7\right)\right]^T$ such as

$$\text{sat}(\bullet_i, \mu_i) = \begin{cases} \mu_i \text{sign}(\bullet_i), & \text{if } |\bullet_i| \geq \mu_i, \\ \bullet_i, & \text{if } |\bullet_i| < \mu_i, \end{cases} \tag{25}$$

with $\bullet_i = |\varepsilon_i|^{\gamma_{2i}-1}\dot{\varepsilon}_i$ for $i = 1, \cdots, 7$ and $0 < \mu_i < 1$. It is clear that the function used (sat) will limit the amplitude to $\mu_i \text{sign}(\bullet_i)$.

Otherwise, to reject the effect of the perturbations and the unknown user–robot interaction force and to ensure fixed-time convergence during the reaching phase, the following bi-power reaching law is added to the equivalent control:

$$\tau_{rl} = -D(\theta)\Phi^{-1}R^{-1}\left[\beta_1 \text{sig}^{\upsilon_1}(s) + \beta_2 \text{sig}^{\upsilon_2}(s) + \beta_3 \text{sign}(s)\right], \tag{26}$$

where β_1, β_2, and β_3 are (7×7) diagonal positive-definite matrices tuned to meet the closed-loop stability conditions and $\text{sig}^{\upsilon_j}(s) = \left[|s_1|^{\upsilon_{j1}}\text{sign}(s_1), \ldots, |s_7|^{\upsilon_{j7}}\text{sign}(s_7)\right]^T$ for $j = 1, 2$ with

$$\upsilon_{1i} = \begin{cases} \upsilon_{1i}^*, & \text{if } |s_i| > 1, \\ 1, & \text{if } |s_i| \leq 1, \end{cases} \tag{27}$$

$$\upsilon_{2i} = \begin{cases} 1, & \text{if } |s_i| > 1, \\ \upsilon_{2i}^*, & \text{if } |s_i| \leq 1, \end{cases} \tag{28}$$

where $\upsilon_{1i}^* > 1$ and $0 < \upsilon_{2i}^* < 1$.

Remark 2. Note that when the robot states are far away from the proposed sliding surface ($|s_i| > 1$), the terms $\beta_1 \text{sig}^{\upsilon_1^*}(s)$ take the lead and are responsible for the convergence speed. Otherwise, when the value of the designed terminal sliding surface becomes small ($|s_i| \leq 1$), the term $\beta_2 \text{sig}^{\upsilon_2^*}(s)$ takes the lead, such that slightly faster convergence is obtained in comparison with the classical reaching law. Finally, the term $\beta_3 \text{sign}(s)$ ensures the robustness and helps for faster convergence.

Finally, the fixed-time TSMC with prescribed performance is obtained by combining Equations (23) and law (26) as follows:

$$\tau = \tau_{eq} + \tau_{rl}. \tag{29}$$

To analyze the overall system's stability using the Lyapunov theory, let us consider:

$$\begin{aligned} V_2 =&0.5s^T s, \\ \dot{V}_2 =&s^T \dot{s}. \end{aligned} \tag{30}$$

Upon substituting the control law in (29) into the studied robot model (3) to obtain the closed-loop error dynamics, then, $\dot{V}_2$ becomes:

$$\begin{aligned}\dot{V}_2 =&s^T\left[\tau_p + N(\theta, \dot{\theta}) - \beta_1 \text{sig}^{\upsilon_1}(s) - \beta_2 \text{sig}^{\upsilon_2}(s) - \beta_3 \text{sign}(s)\right] \\ \leq& -\underline{\beta}_1 \|s\|^{\upsilon_1 + 1} - \underline{\beta}_2 \|s\|^{\upsilon_2 + 1} - \left(\underline{\beta}_3 - \|\tau_p + N(\theta, \dot{\theta})\|\right)\|s\|,\end{aligned} \tag{31}$$

where $\|\bullet\|$ is the Euclidean norm of $\bullet$ and $\underline{\beta}_1$, $\underline{\beta}_2$, $\underline{\beta}_3$ are, respectively, the minimum eigenvalues of the matrices β_1, β_2, and β_3. Moreover, it is clear that the time derivative of $\dot{V}_2$ is negative definite under the condition that $\underline{\beta}_3$ verifies

$$\underline{\beta}_3 \geq \|\tau_p + N(\theta, \dot{\theta})\|. \tag{32}$$

To meet the above stability condition, the value of $\underline{\beta}_3$ will be overestimated, which will lead to a high chattering. As a solution, in the following, the dynamics $N(\theta, \dot{\theta})$ will be estimated using the RBF neural network and added to the control law.

3.2 | RBF

Dealing with a 7-DoF serial-link manipulator with high joint coupling results in significant nonlinearity and computational Coriolis forces. Therefore, it is crucial to not rely on the precise nonlinear expression and instead attempt to estimate and replace the mathematical expression using neural networks. RBF is commonly utilized for estimating nonlinear relationships in multi-input–multi-output systems. In the presence of sufficient sampling data, RBF can establish a precise correlation between the input and output data.

The RBF network is made up of an input layer, a hidden layer that uses nonlinear activation functions, and a linear output layer. These layers are connected to form a typical neural network structure. The algorithm of an RBF network can be expressed as

$$\varphi_j = \exp\left(\frac{-||X - c_j||^2}{2\sigma_j^2}\right), \tag{33}$$

$$N(\theta, \dot{\theta}) = W^T\varphi + e_r, \tag{34}$$

where φ_j for $j = 1, \cdots, k$ is the Gaussian function that is used as an activation function in the hidden layer and $\varphi = [\varphi_1, \cdots, \varphi_k]^T, X = [x_1, \ldots, x_n]$ represents the input vector which contains the measured joint positions and velocities in our case $X = [\theta^T, \dot{\theta}^T]$, c_j represents the center value of each Gaussian function, σ_j is the base width of the function, W denotes the network weights, while e_r is the neural network approximation error. Moreover, $||X - c_j||$ could be expressed as

$$||X - c_j|| = \sqrt{(x_1 - c_{j_1})^2 + \cdots + (x_n - c_{j_n})^2}. \tag{35}$$

The aim of using an RBF is to estimate the nonlinear function $\hat{N}(\theta, \dot{\theta})$. This estimation will replace the usage of the mathematical expression $N(\theta, \dot{\theta})$ which is a (7×1) non-numerical vector. The reproduction of $\hat{N}(\theta, \dot{\theta})$ using RBF will result in having a numerical matrix that has less computational time and less storage space when the controller is applied in real time. $\hat{N}(\theta, \dot{\theta})$, which is the output of the RBF could be expressed as

$$\hat{N}(\theta, \dot{\theta}) = \hat{W}^T\varphi, \tag{36}$$

where $\hat{W}$ is the estimated neural network weights and φ is the output of the Gaussian function. After estimating $\hat{N}(\theta, \dot{\theta})$, the following control action will be added to the computed law in (21) such as

$$\begin{aligned} \tau &= \tau_{eq} + \tau_{rl} + \tau_{nn}, \\ \tau_{nn} &= -\hat{N}(\theta, \dot{\theta}). \end{aligned} \tag{37}$$

Theorem 2. *Consider the robot dynamics in (3) and the designed RBF-based fixed-time sliding mode controller with prescribed performance in (37), if the matrix β_3 is designed to meet the following condition:*

$$\beta_{3i} \geq |\tau_{pi} + e_{ri}|. \tag{38}$$

Then, a fixed-time reaching phase is ensured, and the maximal time is given by:

$$T_{1i} \leq \frac{1}{\beta_{1i}(1 - \upsilon_{2i}^*)} \ln\left(1 + \frac{\beta_{1i}}{\beta_{2i}}\right) + \frac{1}{\beta_{2i}(\upsilon_{1i}^* - 1)} \ln\left(1 + \frac{\beta_{2i}}{\beta_{1i}}\right). \tag{39}$$

Proof. First of all, let us compute the closed-loop error dynamics by substituting (37) in (3):

$$\dot{s} = \tau_p + \tilde{\mathrm{N}}(\theta, \dot{\theta}) - \beta_1 \mathrm{sig}^{\upsilon_1}(s) - \beta_2 \mathrm{sig}^{\upsilon_2}(s) - \beta_3 \mathrm{sign}(s), \tag{40}$$

where $\tilde{\mathrm{N}}(\theta, \dot{\theta})$ is the estimation error such as

$$\begin{aligned} \tilde{\mathrm{N}}(\theta, \dot{\theta}) =& N(\theta, \dot{\theta}) - \hat{N}(\theta, \dot{\theta}) \\ =& W^T\varphi + e_r - \hat{W}^T\varphi = \tilde{W}^T\varphi + e_r. \end{aligned} \tag{41}$$

Now, let us choose the Lyapunov function:

$$V_3 = 0.5\left(s^T s + \mathrm{tr}\left(\tilde{W}^T L \tilde{W}\right)\right), \tag{42}$$

where L is a positive diagonal matrix. By taking the time derivative of V_3 and using (40) and (41):

$$\begin{aligned} \dot{V}_3 =& s^T\dot{s} + \mathrm{tr}\left(\tilde{W}^T L \dot{\tilde{W}}\right) \\ =& s^T\left(\tau_p + e_r - \beta_1 \mathrm{sig}^{\upsilon_1}(s) - \beta_2 \mathrm{sig}^{\upsilon_2}(s) - \beta_3 \mathrm{sign}(s)\right) \\ &+ s^T\tilde{W}^T\varphi - \mathrm{tr}\left(\tilde{W}^T L \dot{\hat{W}}\right) \\ =& s^T\left(\tau_p + e_r - \beta_1 \mathrm{sig}^{\upsilon_1}(s) - \beta_2 \mathrm{sig}^{\upsilon_2}(s) - \beta_3 \mathrm{sign}(s)\right) \\ &- \mathrm{tr}\left(\tilde{W}^T\left(L\dot{\hat{W}} - \varphi s^T\right)\right). \end{aligned} \tag{43}$$

By selecting the following adaptive rule:

$$\dot{\hat{W}} = L^{-1}\varphi s^T. \tag{44}$$

The derivative of the Lyapunov function $\dot{V}_3$ becomes

$$\begin{aligned} \dot{V}_3 =& s^T\left(\tau_p + e_r - \beta_1 \mathrm{sig}^{\upsilon_1}(s) - \beta_2 \mathrm{sig}^{\upsilon_2}(s) - \beta_3 \mathrm{sign}(s)\right) \\ \leq& \sum_{i=1}^{7} -\beta_{1i}|s_i|^{\upsilon_{1i}+1} - \beta_{2i}|s_i|^{\upsilon_{2i}+1} - \left(\beta_{3i} - |\tau_{pi} + e_{ri}|\right)|s_i|. \end{aligned} \tag{45}$$

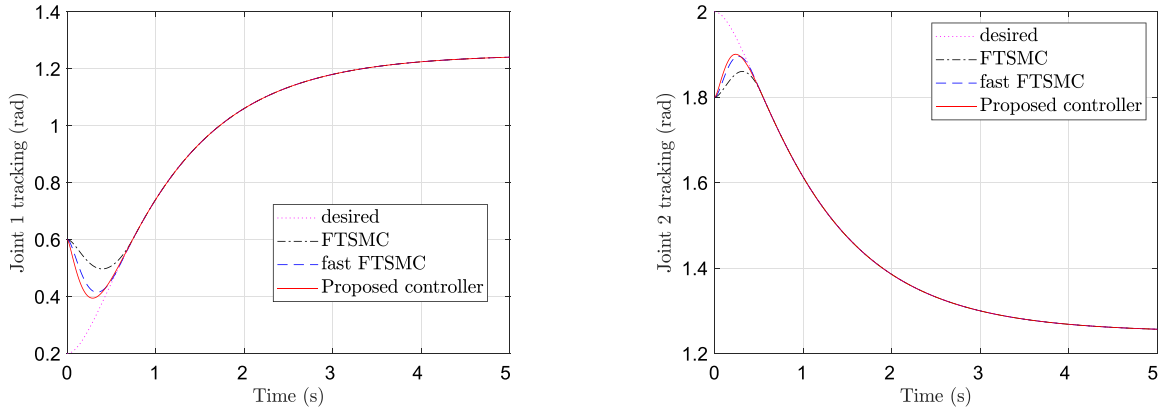


FIGURE 4 Joint tracking comparison. [Color figure can be viewed at wileyonlinelibrary.com]

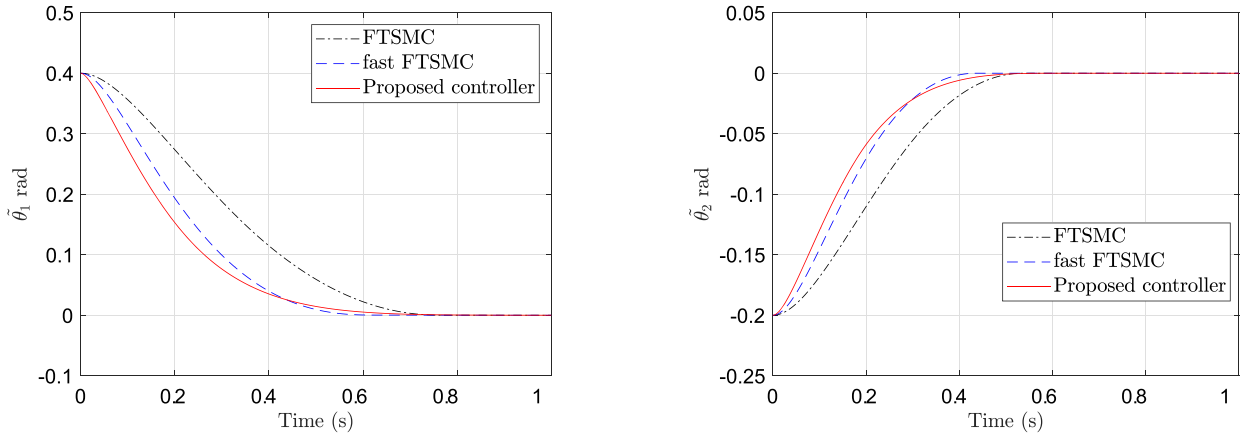


FIGURE 5 Tracking error comparison for Joints 1 and 2. [Color figure can be viewed at wileyonlinelibrary.com]

It can be noticed that $\dot{V}_3$ is negative definite if the condition in (38) is verified. Moreover, Equation (45) can be rewritten as

$$\begin{aligned}\dot{V}_3 &\leq \sum_{i=1}^{7} -\beta_{1i}|s_i|^{\upsilon_{1i}+1} - \beta_{2i}|s_i|^{\upsilon_{2i}+1},\\ \dot{V}_{3i} &\leq -\beta_{1i}(2V_{3i})^{0.5(\upsilon_{1i}+1)} - \beta_{2i}(2V_{3i})^{0.5(\upsilon_{2i}+1)}.\end{aligned} \tag{46}$$

The above result is similar to the one for the sliding phase in (17). Hence, following the same steps, the fixed-time convergence can be proved, and the maximal convergence time is given in (39).

This completes the proof. □

Remark 3. It is important to note that for minor estimation error and perturbations $\tau_p + e_r$ and large values for the gain β_3, the estimated reaching fixed-time upper bound in Equation (17) may be conservative.

4 | CASE STUDIES

In this section, we will demonstrate the validation process for the proposed controller. Two demonstrations will be presented to show the controller's effectiveness. The first one will involve a simulation comparison between the proposed controller, FTSMC, and fast FTSMC on a 2-DoF serial-link RBF's estimation and the proposed controller's effectiveness and the effectiveness of the proposed controller. The second demonstration will be a real-time experiment on the 7-DoF wearable ETS-MARSE, aimed at examining the proposed controller.

4.1 | Numerical simulations

To demonstrate the effectiveness of the proposed controller, it is necessary to compare it with other controllers based on SMC. This comparison will highlight the success of the main objective stated in the introduction, which is to reduce the chattering of the control signals. The comparison will be conducted using simulation and a simplified dynamic model of a 2-DoF robot for clarity. The robot structure and figure can be found in [54]. The dynamic model of the studied robot is as follows:

$$\underbrace{\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}}_{D(\theta)} \ddot{\theta} + \underbrace{\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}}_{C(\theta,\dot{\theta})\dot{\theta}} + \underbrace{\begin{bmatrix} G_1 \\ G_2 \end{bmatrix}}_{G(\theta)} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}}_{\tau}, \tag{47}$$

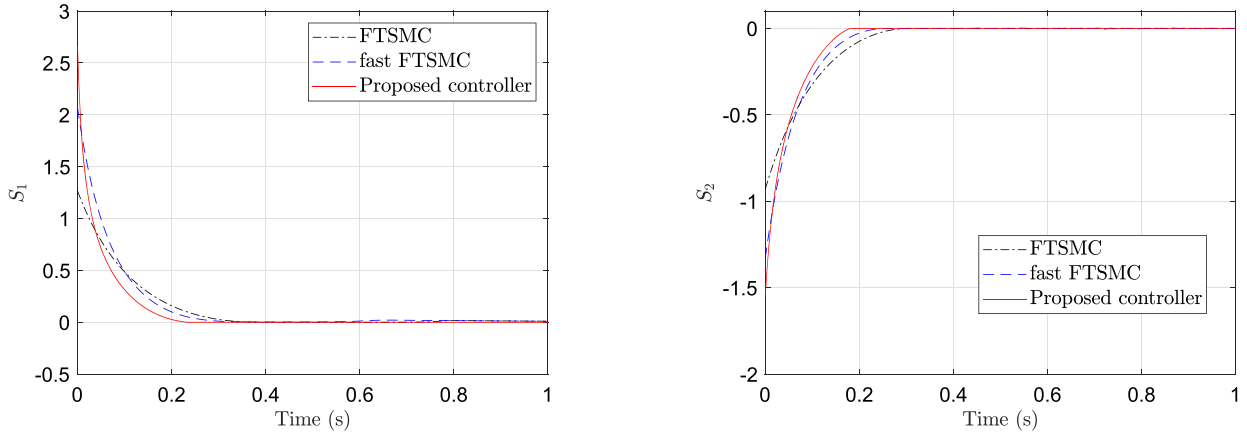


FIGURE 6 Sliding surface comparison for Joints 1 and 2. [Color figure can be viewed at wileyonlinelibrary.com]

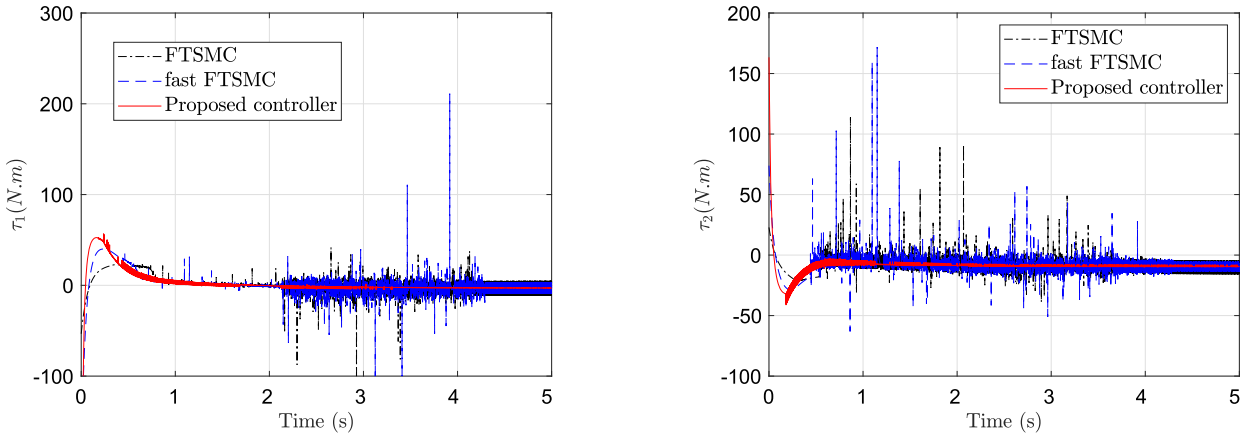


FIGURE 7 Torque comparison for Joints 1 and 2. [Color figure can be viewed at wileyonlinelibrary.com]

where

$$
\begin{aligned}
&D_{11} = 2.96 + 1.6\cos(\theta_2),\ \ D_{12} = D_{21} = 0.96 + 1.2\cos(\theta_2),\\
&D_{22} = 0.96,\\
&C_1 = -1.2\sin(\theta_2)\dot{\theta}_2^2 - 2.4\sin(\theta_2)\dot{\theta}_1\dot{\theta}_2,\ C_2 = 1.2\sin(\theta_2)\dot{\theta}_2^2,\\
&G_1 = 11.772\cos(\theta_1+\theta_2) + 19.62\cos(\theta_1),\\
&G_2 = 11.772\cos(\theta_1+\theta_2).
\end{aligned}
\tag{48}
$$

The proposed controller has been compared with two other FTSMC-based controllers:

- Controller 1: the non-singular fixed-time TSMC τ_{c1} [55];
- Controller 2: a fixed-time technique based on the result presented in Theorem 1 in [56].

The expression of these controllers is given by

$$
\begin{aligned}
\tau_{c1} = &-D(\theta)(\lambda_2\alpha_2)^{-1}\lfloor\dot{\tilde{\theta}}\rceil^{I-\alpha_2}\left(\dot{\tilde{\theta}} + \lambda_1\alpha_1\lfloor\tilde{\theta}\rceil^{\alpha_1-I}\dot{\tilde{\theta}}\right)\\
&-D(\theta)(\lambda_2\alpha_2)^{-1}\lfloor\dot{\tilde{\theta}}\rceil^{I-\alpha_2}\left(K_1\mathrm{sig}^{0.5}(\bar{s}) + K_2\mathrm{sig}^{1.5}(\bar{s})\right)\\
&+D(\theta)\left(\ddot{\theta}_d - K_3\mathrm{sign}(\bar{s})\right) + C(\theta,\dot{\theta})\dot{\theta} + G(\theta),
\end{aligned}
\tag{49}
$$

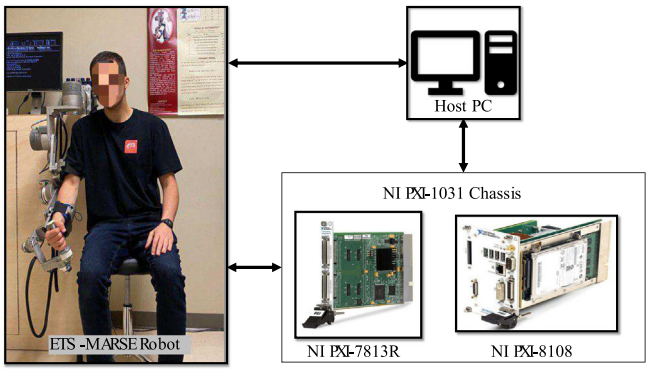


FIGURE 8 ETS-MARSE experimental setup. [Color figure can be viewed at wileyonlinelibrary.com]

where $\bar{s} = \tilde{\theta} + \lambda_1\mathrm{sig}^{\alpha_1}(\tilde{\theta}) + \lambda_2\mathrm{sig}^{\alpha_2}(\dot{\tilde{\theta}})$ with $\alpha_1 = \alpha_1^* I$, $\alpha_2 = \alpha_2^* I$ such as $1 < \alpha_2^* < 2$ and $\alpha_2^* < \alpha_1^*$.

$$
\begin{aligned}
\tau_{c2} = &-D(\theta)\left(\lambda_1\delta_1\lfloor\tilde{\theta}\rceil^{\delta_1-I} + \lambda_2\delta_2\lfloor\tilde{\theta}\rceil^{\delta_2-I} + \lambda_3\right)\dot{\tilde{\theta}}\\
&-D(\theta)\left(K_1\mathrm{sig}^{0.5}\left(\underline{s}\right) + K_2\mathrm{sig}^{1.5}\left(\underline{s}\right) + K_3\underline{s}\right)\\
&+D(\theta)\ddot{\theta}_d + C(\theta,\dot{\theta})\dot{\theta} + G(\theta),
\end{aligned}
\tag{50}
$$

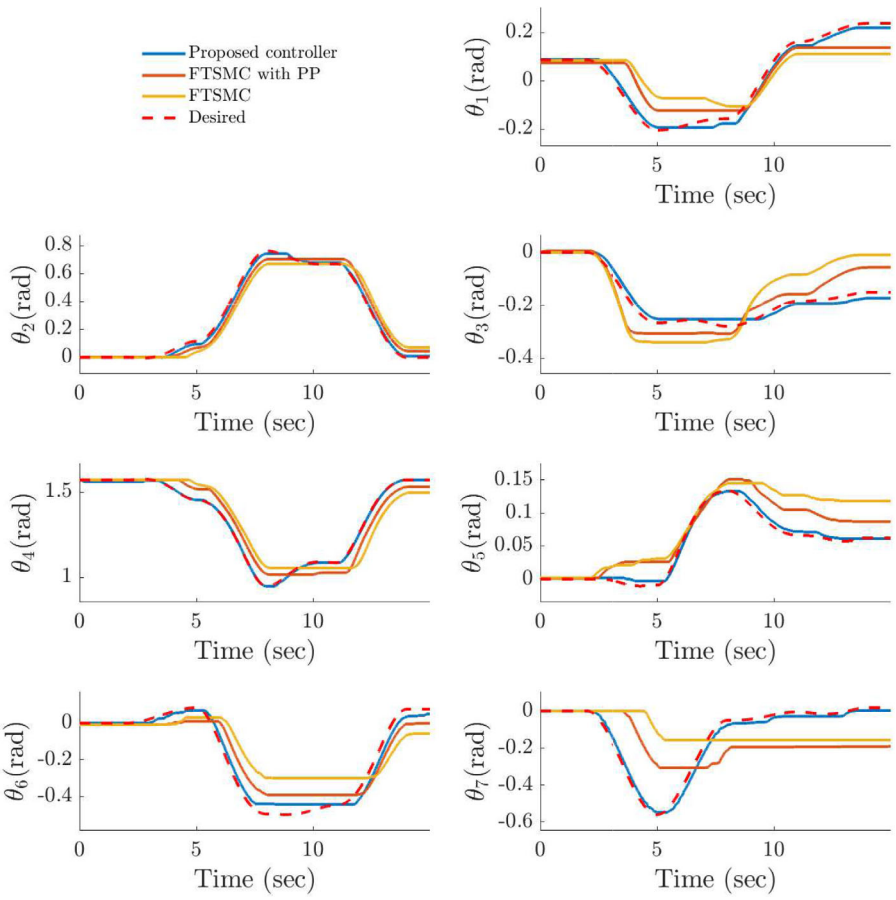


FIGURE 9 Joint space trajectory tracking. [Color figure can be viewed at wileyonlinelibrary.com]

where $\underline{s} = \dot{\tilde{\theta}} + \lambda_1 \text{sig}^{\delta_1}(\tilde{\theta}) + \lambda_2 \text{sig}^{\delta_2}(\tilde{\theta}) + \lambda_3 \tilde{\theta}$ with $\delta_1 = \delta_1^* I$, $\delta_2 = \delta_2^* I$ such as $0 < \delta_1^* < 1$ and $1 < \delta_2^*$.

Using MATLAB/Simulink software, the dynamic model in (47) has been simulated for 5 s and with a step time of 1 ms. The ode3 that is used for the simulation, also known as Bogacki–Shampine solver, is a third-order Runge–Kutta method that offers an efficient trade-off between computational speed and accuracy, making it ideal for real-time control simulations. Its fixed-step approach is particularly suited for complex systems like the ETS-MARSE exoskeleton, where rapid execution is crucial without compromising the stability of the control system. This makes ode3 well suited for hardware-in-the-loop simulations or scenarios requiring moderate accuracy with minimal computational overhead. The first joint has been given a trajectory of an increasing smooth step function. The second joint has been simulated with a decreasing smooth step function.

$$\theta_1^d = 0.35 \exp(-4t) - 1.4 \exp(-t) + 1.25, \quad (51)$$

$$\theta_2^d = -0.25 \exp(-4t) + \exp(-t) + 1.25. \quad (52)$$

In Figure 4, a comparison of the tracking control of a 2-DoF serial-link manipulator using the proposed controller and two other controllers (49) and (50) is shown. The proposed controller achieves faster and more accurate tracking of the desired trajectory for both joints compared to the other controllers.

Figures 5–7 show the tracking error, sliding surface (the first second of the 5 s for clarity), and torque signals, respectively. Based on the tracking error figures for both joints, the proposed controller tracks the error more accurately and converges to zero much faster than the other two controllers. Moreover, the sliding surface of the proposed controller reaches the equilibrium point quicker than its comparatives. The significant difference between the controllers can be observed in Figure 7. The proposed controller shows less chattering in the control signals for both joints, which is not the case for the other two controllers. These controllers require high-torque signals (almost 200 N.m) to maintain a stable steady-state phase, which can lead to hardware damage for the joints and their corresponding motors. Therefore, it is recommended not to apply the FTSMC and the FFTSMC experimentally for our rehabilitation process.

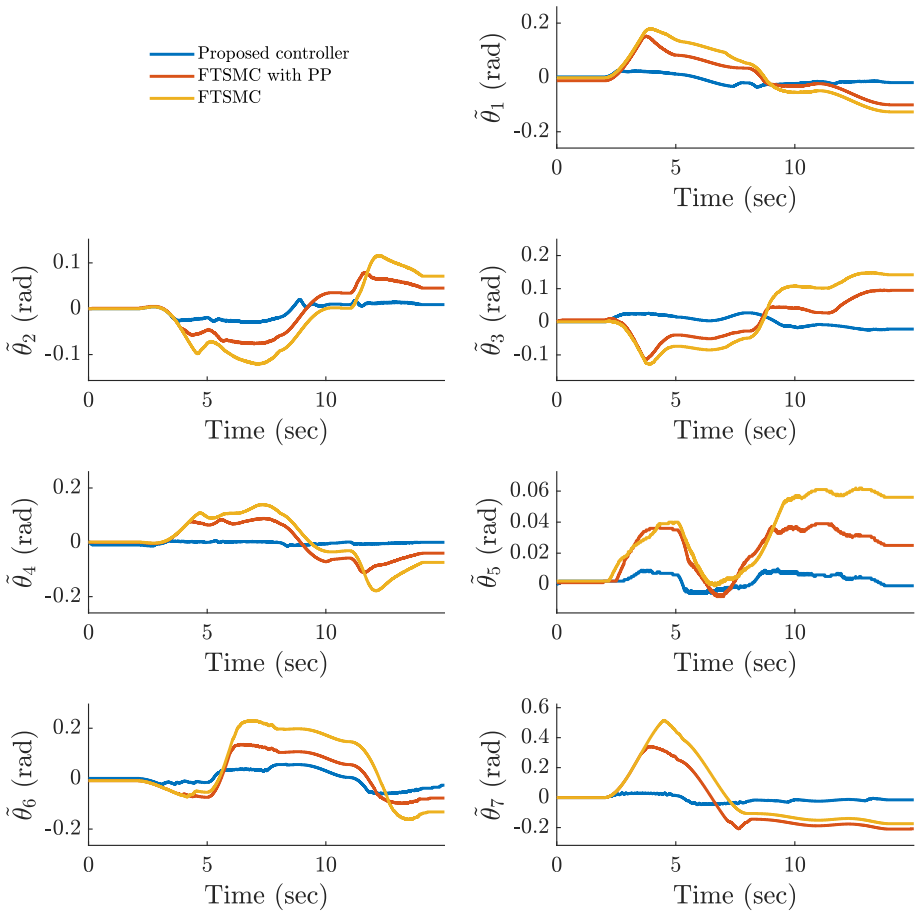


FIGURE 10 Joint space tracking error. [Color figure can be viewed at wileyonlinelibrary.com]

4.2 | Real-time experiment on the 7-DoF exoskeleton robot

This section applies the proposed controller to the ETS-MARSE robot. The robot was tested using a passive rehabilitation mode in which it was assigned to follow a predefined trajectory in Cartesian space (see Figure 12).

4.2.1 | Real-time system

The real-time setup for the experiment comprises four pieces of equipment, depicted in Figure 8: a real-time PC (NI PXI-8108), an FPGA (NI PXI-7813R), a host PC, and the ETS-MARSE robot. The real-time PC features an Intel dual-core processor running at 2.53 GHz and 8 GB of RAM, which is used to execute the proposed controller and the dynamics of the 7-DoF exoskeleton ETS-MARSE. The PXI-7813R FPGA handles both the analog and digital I/O for the actuators and sensors, which include a Hall effect position sensor and a current sensor. The host PC features an Intel Core i7-4770 CPU operating at 3.4 GHz and is equipped with 16 GB of RAM. This setup is utilized for RBF training and to handle the mathematical development of the proposed controller on LabView software. Furthermore, the host PC serves as the user interface during real-time tests, storing and showcasing the results of the trials.

4.2.2 | Adjusting the parameters and conducting the experiment

After conducting trial-and-error tests without the need for further tuning and considering various scenarios with different conditions, the gains for the proposed controller were finally determined to be

$$
\begin{aligned}
&K = \mathrm{diag}(100{,}100{,}100{,}150{,}500{,}400{,}600),\\
&\Lambda = \mathrm{diag}(45{,}45{,}45{,}35{,}20{,}20{,}10),\\
&\phi_i^0 = 0.1,\ \phi_i^f = 0.01,\ \mu_i = 0.3, \alpha_i = 2,\\
&\gamma_{1i}^* = 1.2,\ \gamma_{2i}^* = 0.6,\ \upsilon_{1i}^* = 1.7,\ \upsilon_{2i}^* = 0.7,\\
&\beta_1 = \beta_2 = \beta_3 = 1 * I_{7\times7}, \beta_4 = 0.3 * I_{7\times7}.
\end{aligned}
$$

The robot was given a trajectory to track within 16 s. Three controllers have been tested under the same condition: the proposed controller, FTSMC with PP (prescribed performance), and FTSMC. Figure 9 displays the joint tracking performance of ETS-MARSE. The tracking is clearly better using the proposed controller, even when multiple joints are operating simultaneously to achieve the workspace trajectory. Figure 10 shows the tracking error of each joint in radians. The tracking error of all joints is better using the proposed controller. Figure 11 shows the control torque signal for each joint. The chattering of the signals and the

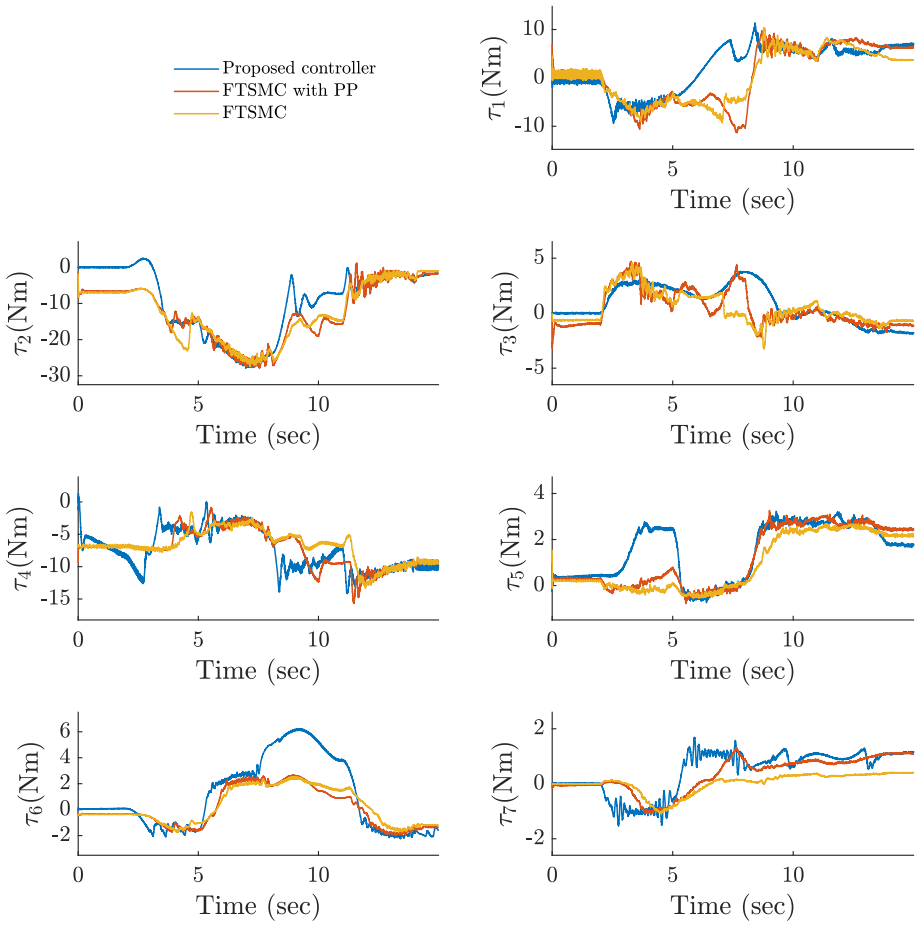


FIGURE 11 Applied computed joint torques. [Color figure can be viewed at wileyonlinelibrary.com]

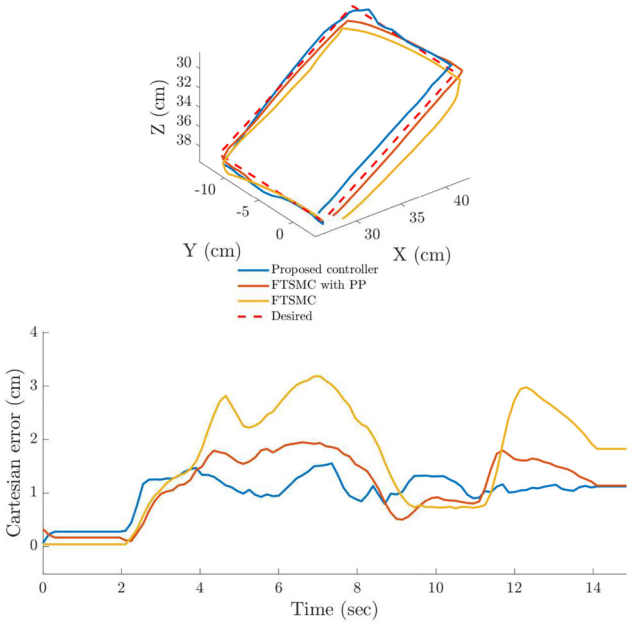


FIGURE 12 3D workspace trajectory tracking. [Color figure can be viewed at wileyonlinelibrary.com]

oscillations are not noticeable when using the proposed controller while running the robot in the real-time scenario. Thus, these signals are safe on the robot hardware.

Figure 12 shows both tracking performance and tracking error in the workspace. The predefined trajectory is a square shape in the 3D space.

Expanded discussion

Figures 9–12 provide a deeper understanding of the controller's performance.

- Figure 9 (joint space trajectory tracking): The tracking results reveal that the proposed controller ensures accurate joint movements across all seven joint. This result is significant as it indicates that the controller can handle the complexities of multi-joint operations typical in rehabilitation tasks.
- Figure 10 (joint space tracking error): The tracking errors demonstrate that the proposed controller is highly effective in minimizing deviation. This is especially important for rehabilitation robotics, where precision is critical to ensure patient safety and the efficacy of exercises.
- Figure 11 (control torque signals): The smooth and stable torque signals indicate that the controller successfully mitigates the chattering commonly associated with SMC. This improvement not only extends the lifespan of the robot hardware but also ensures smoother operation during patient interaction.

TABLE 3 Comparison of IAE, ITAE, and ISE for the seven joints.

	Joint						
	1	2	3	4	5	6	7
IAE							
Proposed controller	0.248	0.192	0.212	0.0706	0.0628	0.421	0.290
FTSMC with PP	0.803	0.599	0.682	0.734	0.323	1.003	2.331
FTSMC	1.148	0.818	1.176	1.010	0.472	1.623	2.388
ITAE							
Proposed controller	2.104	1.533	1.851	0.408	0.474	4.044	2.277
FTSMC with PP	6.657	5.434	6.314	6.661	2.961	8.826	20.107
FTSMC	9.665	7.404	11.455	9.392	4.761	15.023	18.533
ISE							
Proposed controller	0.00519	0.00378	0.00413	0.000555	0.0003555	0.01784	0.00783
FTSMC with PP	0.0641	0.0342	0.0450	0.0513	0.00985	0.0905	0.486
FTSMC	0.127	0.0739	0.128	0.108	0.0230	0.261	0.632

- Figure 12 (3D workspace trajectory tracking): The workspace tracking performance is equally impressive, with a Euclidean distance error of less than 1.5 cm, confirming that the controller can maintain high accuracy even in more complex, three-dimensional movements.

Table 3 presents a comparative analysis of the IAE, ITAE, and ISE performance metrics for seven joints under three different control strategies: proposed controller, FTSMC with PP, and FTSMC. The proposed controller consistently demonstrates superior performance across all joints, achieving significantly lower values in all three metrics. For IAE, the proposed controller shows an improvement of 69.4% on average compared to FTSMC and 53.7% compared to FTSMC with PP. Similarly, for ITAE, the proposed controller achieves an average reduction of 77.8% compared to FTSMC and 64.3% compared to FTSMC with PP. Finally, in ISE, the proposed controller exhibits an impressive reduction of 93.1% on average compared to FTSMC and 84.5% compared to FTSMC with PP. These improvements indicate the effectiveness of the proposed controller in reducing errors and enhancing system performance, making it the most efficient method among the three.

5 | CONCLUSION

This study presented a robust and effective TSMC with prescribed performance for the ETS-MARSE robot deployed in rehabilitation. By integrating an RBF for nonlinear dynamics estimation and incorporating prescribed performance criteria, the controller ensures precise trajectory tracking and rejects the effects of unknown dynamics, human–robot interaction forces, and matched disturbances. The prescribed performance function efficiently reduces tracking errors to a user-defined threshold. In addition, the torques shown in the comparison simulation confirm that the proposed controller has less chattering in the stead-state phase. We proposed a sliding surface incorporating the prescribed performance effect where a new fixed-time stability proof based on Lyapunov theory is provided. The numerical simulation showed the effectiveness of the developed method compared to other recent FTSMCs. Also, the torques shown in the comparison simulation show that the proposed controller has less chattering in the steady-state phase. The experimental validation conducted on the ETS-MARSE robot underscores the efficacy of our controller. To provide a quantitative comparison, the IAE, ITAE, and ISE metrics have been presented for the proposed controller, FTSMC with a prescribed performance function, and standard FTSMC across seven joints. The results demonstrate that our approach significantly outperforms existing methods in tracking accuracy, disturbance rejection, and chattering reduction, highlighting the effectiveness of our control strategy in rehabilitation robotics. In the future, we will try to use other machine-learning approaches such as reinforcement learning to estimate nonlinearities. In addition, we are working on testing the proposed controller in different rehabilitation modes such as active rehabilitation and assistive rehabilitation.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

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

ORCID

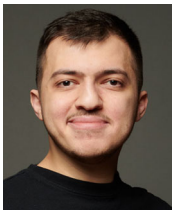
Mahmoud Abdallah https://orcid.org/0009-0004-3118-6044
Yassine Kali https://orcid.org/0000-0001-7887-1584

REFERENCES

1. S. Watson, *How rehab can help you recover from surgery*, 2022. https://www.webmd.com/pain-management/rehab-after-surgery
2. J. Iqbal, O. Ahmad, and A. Malik, *HEXOSYS II—towards realization of light mass robotics for the hand*, 2011 IEEE 14th Int. Multitopic Conf., IEEE, Karachi, Pakistan, 2011, pp. 115–119.
3. J. Kim, F. Porciuncula, H. D. Yang, N. Wendel, T. Baker, A. Chin, T. D. Ellis, and C. J. Walsh, *Soft robotic apparel to avert freezing of gait in Parkinson's disease*, Nature Med. **30** (2024), no. 1, 177–185.
4. J. Iqbal, N. G. Tsagarakis, and D. G. Caldwell, *Four-fingered lightweight exoskeleton robotic device accommodating different hand sizes*, Electron. Lett. **51** (2015), no. 12, 888–890.
5. N. Vitiello, T. Lenzi, S. Roccella, S. M. M. De Rossi, E. Cattin, F. Giovacchini, F. Vecchi, and M. C. Carrozza, *NEUROExos: a powered elbow exoskeleton for physical rehabilitation*, IEEE Trans. Robot. **29** (2012), no. 1, 220–235.
6. J. Iqbal, N. Tsagarakis, A. E. Fiorilla, and D. Caldwell, *Design requirements of a hand exoskeleton robotic device*, 14th IASTED Int. Conf. Rob. Appl. (RA), Vol. **664**, ACTA Press, Cambridge, MA, USA, 2009, pp. 44–51.
7. B. Chen and X. Li, *Neural network control for human-robot interaction in upper limb exoskeletons*, IEEE Trans. Cybern. **50** (2020), no. 5, 1927–1938.
8. S. Kim and C. Park, *Adaptive control for lower limb exoskeletons with time-varying dynamics*, Mechatronics **54** (2018), 193–202.
9. X. Li and H. Zhang, *Fuzzy logic-based control for upper limb exoskeletons with dynamic uncertainties*, Robot. Auton. Syst. **138** (2021), 103637.
10. Z. Ding and L. Yang, *Neural adaptive control for trajectory tracking of exoskeleton robots*, J. Bionic Eng. **17** (2020), no. 4, 822–834.
11. X. Wu and Y. Zheng, *Robust terminal sliding mode control for exoskeletons with prescribed performance*, J. Intell. Fuzzy Syst. **37** (2019), no. 1, 71–84.
12. Z. Xu and Y. Zhang, *Nonlinear adaptive control for exoskeletons using RBF networks*, Neurocomputing **400** (2020), 276–287.
13. Q. Zhu and Y. Zhang, *Sliding mode control for upper limb exoskeletons: a comparative study*, IEEE Access **7** (2019), 117395–117404.
14. Y. L. Chiu and M. J. Chung, *Adaptive neural sliding mode control for lower limb exoskeletons*, IEEE Trans. Ind. Inform. **17** (2021), no. 9, 6185–6196.
15. Q. Li and Z. Li, *Sliding mode control for lower limb exoskeletons with human-robot interaction*, Int. J. Control Autom. Syst. **17** (2019), no. 5, 1292–1303.
16. Y. He and Y. Shen, *Intelligent control of upper limb exoskeletons using neural-fuzzy systems*, Neural Comput. Appl. **29** (2018), no. 9, 389–398.
17. G. Liu and Q. Hu, *Hybrid fuzzy sliding mode control for upper limb exoskeleton robots*, Int. J. Control, Autom. Syst. **19** (2021), no. 1, 189–200.
18. J. Sun and Y. Li, *Model predictive control for rehabilitation robots with trajectory tracking*, IEEE Trans. Syst. Man Cybern.: Syst. **48** (2018), no. 11, 1976–1985.
19. T. Yan and W. Wang, *Model predictive control for lower limb exoskeletons in rehabilitation training*, Control Eng. Pract. **110** (2021), 104758.
20. J. Zhou and Y. Zhao, *Robust control of exoskeletons for rehabilitation using H-infinity methods*, J. Intell. Robot. Syst. **96** (2019), no. 2, 203–215.
21. C. E. Boudjedir, M. Bouri, and D. Boukhetala, *An enhanced adaptive time delay control-based integral sliding mode for trajectory tracking of robot manipulators*, IEEE Trans. Control Syst. Technol. **31** (2022), 1042–1050.
22. A. Ferrara and G. P. Incremona, *Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators*, IEEE Trans. Control Syst. Technol. **23** (2015), no. 6, 2316–2325.
23. X. Lin, Y. Wang, and Y. Liu, *Neural-network-based robust terminal sliding-mode control of quadrotor*, Asian J. Control **24** (2022), no. 1, 427–438.
24. S. Ahmed, H. Wang, and Y. Tian, *Adaptive fractional high-order terminal sliding mode control for nonlinear robotic manipulator under alternating loads*, Asian J. Control **23** (2021), no. 4, 1900–1910.
25. S. Saha, S. Amrr, A. Saidi, A. Banerjee, and M. Nabi, *Finite-time adaptive higher-order SMC for the nonlinear five DOF active magnetic bearing system*, Electronics **10** (2021), 1333.
26. J. Rodas, I. Gonzalez-Prieto, Y. Kali, M. Saad, and J. Doval-Gandoy, *Recent advances in model predictive and sliding mode current control techniques of multiphase induction machines*, Front. Energy Res. **9** (2021), 729034.
27. Y. Kali, K. Benjelloun, A. Fatemi, M. Saad, and M. Benbrahim, *Second order sliding mode with time delay control for uncertain robot manipulators*, 2016 IEEE Int. Multidiscip. Conf. Eng. Technol. (IMCET), IEEE, Beirut, Lebanon, 2016, pp. 160–165.
28. W. K. Alqaisi, B. Brahmi, J. Ghommam, M. Saad, and V. Nerguizian, *Adaptive sliding mode control based on RBF neural network approximation for quadrotor*, 2019 IEEE Int. Symp. Rob. Sens. Environ. (ROSE), IEEE, Ottawa, ON, Canada, 2019, pp. 1–7.
29. Y. Yao, L. Ding, and Y. Wang, *Fractional-order nonsingular terminal sliding mode control of a cable-driven aerial manipulator based on RBF neural network*, Int. J. Aeronaut. Space Sci. **25** (2023), 759–771.
30. C. Fallaha, M. Saad, H. Kanaan, and K. Al-Haddad, *Sliding-mode robot control with exponential reaching law*, IEEE Trans. Ind. Electron. **58** (2010), 600–610.
31. P. Tirop and Z. Jingrui, *Review of control methods and strategies of space tether satellites*, Am. J. Traffic Transp. Eng. **4** (2019), 137.
32. J. Pliego-Jiménez, M. A. Arteaga-Pérez, and M. López-Rodríguez, *Finite-time control for rigid robots with bounded input torques*, Control Eng. Pract. **102** (2020), 104556.
33. G. Bartolini, A. Pisano, E. Punta, and E. Usai, *A survey of applications of second-order sliding mode control to mechanical systems*, Int. J. Control **76** (2003), no. 9–10, 875–892.
34. R. Prakash, L. Behera, S. Mohan, and S. Jagannathan, *Dynamic trajectory generation and a robust controller to intercept a moving ball in a game setting*, IEEE Trans. Control Systems Technol. **28** (2019), no. 4, 1418–1432.
35. J. Mao, J. Yang, X. Liu, S. Li, and Q. Li, *Modeling and robust continuous TSM control for an inertially stabilized platform with couplings*, IEEE Trans. Control Syst. Technol. **28** (2019), no. 6, 2548–2555.
36. D. Zhao, S. Li, and F. Gao, *A new terminal sliding mode control for robotic manipulators*, Int. J. Control **82** (2009), 1804–1813.
37. H. Komurcugil, *Non-singular terminal sliding-mode control of DC–DC buck converters*, Control Eng. Pract. **21** (2013), no. 3, 321–332.

38. S. Ahmed, H. Wang, and Y. Tian, *Robust adaptive fractional-order terminal sliding mode control for lower-limb exoskeleton*, Asian J. Control **21** (2019), 473–482.
39. T. Madani, B. Daachi, and K. Djouani, *Non-singular terminal sliding mode controller: application to an actuated exoskeleton*, Mechatronics **33** (2016), 136–145.
40. A. Riani, T. Madani, A. Benallegue, and K. Djouani, *Adaptive integral terminal sliding mode control for upper-limb rehabilitation exoskeleton*, Control Eng. Pract. **75** (2018), 108–117.
41. X. Bu, *Prescribed performance control approaches, applications and challenges: a comprehensive survey*, Asian J. Control **25** (2023), no. 1, 241–261.
42. D. Shi, J. Zhang, Z. Sun, and Y. Xia, *Adaptive sliding mode disturbance observer-based composite trajectory tracking control for robot manipulator with prescribed performance*, Nonlinear Dyn. **109** (2022), 2693–2704.
43. P. Yang and Y. Su, *Proximate fixed-time prescribed performance tracking control of uncertain robot manipulators*, IEEE/ASME Trans. Mechatron. **27** (2021), 3275–3285.
44. Y. Zhang, L. Fang, T. Song, and M. Zhang, *Adaptive non-singular fast terminal sliding mode based on prescribed performance for robot manipulators*, Asian J. Control **25** (2023), 3253–3268.
45. D. D'Agostino, I. Ilievski, and C. A. Shoemaker, *Learning active subspaces and discovering important features with Gaussian radial basis functions neural networks*, Neural Netw. **176** (2024), 106335.
46. T. Chettibi, *Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions*, Robotica **37** (2019), 539–559.
47. P. Zhang, T. Lü, and L. Song, *RBF networks-based inverse kinematics of 6r manipulator*, Int. J. Adv. Manuf. Technol. **26** (2005), 144–147.
48. M. Rahmani and M. Rahman, *Adaptive neural network fast fractional sliding mode control of a 7-DOF exoskeleton robot*, Int. J. Control, Autom. Syst. **18** (2020), 124–133.
49. L. Liu, C. Zhu, Y.-J. Liu, and S. Tong, *Adaptive finite-time neural constrained control for nonlinear active suspension systems based on the command filter*, IEEE Trans. Artif. Intell. **3** (2021), no. 2, 218–227.
50. B. Niu, X. Liu, Z. Guo, H. Jiang, and H. Wang, *Adaptive intelligent control-based consensus tracking for a class of switched non-strict feedback nonlinear multi-agent systems with unmodeled dynamics*, IEEE Trans. Artif. Intell. **5** (2023), 1624–1634.
51. L. Wang, T. Chai, and C. Yang, *Neural-network-based contouring control for robotic manipulators in operational space*, IEEE Trans. Control Syst. Technol. **20** (2011), no. 4, 1073–1080.
52. C. Jing, H. Xu, and X. Niu, *Adaptive sliding mode disturbance rejection control with prescribed performance for robotic manipulators*, ISA Trans. **91** (2019), 41–51.
53. Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding mode control and observation*, Vol. **10**, Birkhäuser, New York, NY, USA, 2014.
54. Y. Kali, M. Saad, K. Benjelloun, and C. Khairallah, *Super-twisting algorithm with time delay estimation for uncertain robot manipulators*, Nonlinear Dyn. **93** (2018), 557–569.
55. Z. Zuo, *Non-singular fixed-time terminal sliding mode control of non-linear systems*, IET Control Theory Appl. **9** (2015), no. 4, 545–552.
56. C. Chen, L. Li, H. Peng, Y. Yang, L. Mi, and H. Zhao, *A new fixed-time stability theorem and its application to the fixed-time synchronization of neural networks*, Neural Netw. **123** (2020), 412–419.

AUTHOR BIOGRAPHIES



Mahmoud Abdallah received his bachelor's degree from the University of Sharjah in 2016 and his master's degree in 2021 from the same university. He is currently a PhD candidate at the École de Technologie Supérieure in Montreal, Canada. His main research interests include robotics, control systems, and artificial intelligence.

Yassine Kali was born in Fez, Morocco. He received the Ph.D. degree in robotics and control systems from Ecole Mohammadia d'Ingénieurs, University of Mohammed V, Rabat, Morocco, in 2018. From 2018 to 2023, he worked at the GRÉPCI Laboratory of École de Technologie Supérieure (Montreal, Canada) as a Postdoctoral Research Fellow and then as a Research Associate. Since December 2023, he joined Université du Québec en Abitibi-Témiscamingue (Rouyn-Noranda, Canada), where he is teaching control theory and robotics. His research interests include robotics, autonomous vehicles, robust nonlinear and intelligent control.

Maarouf Saad received his B.S. and M.S. degrees from the École Polytechnique of Montreal, Montreal, QC, Canada, in 1982 and 1984, respectively, and his Ph.D. degree from McGill University, Montreal, QC, Canada, in 1988, all in electrical engineering. In 1987, he joined École de Technologie Supérieure, Montreal, where he is currently teaching control theory and robotics courses. His research interests include nonlinear control and optimization applied to robotics and flight control system.

Raouf Fareh received the Ph.D. degree in electrical engineering from the University of Quebec (École de Technologie Supérieure), Montreal, QC, Canada, in 2013. In 2013, he joined Ottawa University, Ottawa, ON, Canada, as a Postdoc-

toral Researcher. He joined the University of Sharjah, Sharjah, UAE, as an Assistant Professor in 2014 and was promoted to an Associate Professor in 2020. His research interests include robotics, control systems, and path planning. In control systems, he mainly works on various control methods such as active disturbance rejection control, distributed control, fractional control, sliding-mode control, and synergetic control.

Maamar Bettayeb received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1976, 1978, and 1981, respectively. He was a Research Scientist with the Bellaire Research Center, Shell Oil Development Company, Houston, TX, USA. From 1982 to 1988, he directed the Instrumentation and Control Laboratory of the High Commission for Research in Algeria, Algiers, Algeria. In 1988, he joined the Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He has been a Professor with the University of Sharjah, Sharjah, UAE, since 2000. He has also been the Vice Chancellor for Research and Graduate Studies with the University of Sharjah, since 2014. His recent research interests include process control, fractional dynamics and control, renewable energies, and engineering education. Dr. Bettayeb is also the Chair of the IT Governance Committee of the University of Sharjah.

How to cite this article: M. Abdallah, Y. Kali, M. Saad, R. Fareh, and M. Bettayeb. *Wearable exoskeleton robot control using radial basis function-based fixed-time terminal sliding mode with prescribed performance*. Asian J. Control. (2025), 1–17, DOI 10.1002/asjc.3743.