Wiley
Applied Bionics and Biomechanics
Volume 2026, Article ID 8945626, 20 pages
https://doi.org/10.1155/abb/8945626

WILEY

Research Article

Enhanced Control Strategies for Lower Limb Rehabilitation Robots: A Comparative Study of MRAC, PID-ZN, and MRAC-PSO Controllers

Mustapha Amine Sadi,[1] **Annisa Jamali**,[1] **M. A. Zulkifli**,[1] **M. N. Leman**,[1] **S. Mohamaddan**,[2] **and H. Hazmi**[3]

[1]*Department of Mechanical Engineering, Faculty of Engineering, University Malaysia Sarawak, Kota Samarahan, Malaysia*
[2]*Department of Bioscience and Engineering, College of System Engineering and Science Shibaura Institute of Technology, Tokyo, Japan*
[3]*Department of Community Medicine and Public Health, Faculty of Medicine and Health Sciences, University Malaysia Sarawak, Sarawak, Malaysia*

Correspondence should be addressed to Annisa Jamali; jannisa@unimas.my

Received 7 January 2025; Revised 18 November 2025; Accepted 1 December 2025

Academic Editor: Poorani Gurumallesh

Wearable robots for rehabilitation have dramatically advanced the medical field regarding helping patients suffering from lower limb impairments to regain mobility and ameliorate their range of motion (ROM). However, to further optimize control mechanisms within these robots, conventional methods cannot adapt to the different needs of patients in their walking, and the complex patterns of human gait. As a result of such limitations, the functionality of ROM training is constricted. To cope with these problems, the present article will design and validate an adaptive control system for the lower limbs with Particle Swarm Optimization (PSO): Model Reference Adaptive Control (MRAC)-PSO. By introducing the adaptation mechanism of MRAC-PSO, as well as its capabilities in optimization, this research aims to further improve the adaptability and effectiveness in ROM training to provide a better rehabilitation process for the patient. Moreover, the performance of the MRAC-PSO controller is compared with that of a traditional MRAC system and a classical Proportional-Integral-Derivative controller optimized via the Ziegler–Nichols (Z–N) method. This comparison is performed to underline the benefits and possible improvements brought by the adaptive and optimized control approach. The novelty of this article lies in the first systematic integration of PSO optimization with MRAC for wearable lower limb rehabilitation (WLLR) ROM training, achieving significant performance improvements over conventional methods: 89% faster risetime and 98.9% lower steady-state error (SSE) compared to PID-ZN control. This research advances wearable robotics by demonstrating that the synergy between adaptive control and bioinspired optimization can substantially improve rehabilitation robot performance, safety, and clinical viability. The synthesis and overcoming analysis of MRAC-PSO, conventional MRAC, and PID-ZN controllers are carried out with the aim to overcome existing ROM training deficiencies, making the rehabilitation strategy more adaptable and effective. In this regard, this research outcome would likely open ways for further development in rehabilitation technology to improve the living standard of people suffering from lower limb disabilities.

Keywords: lower limb; MRAC controller; PSO algorithm; rehabilitation robot; ROM training; wearable robot

1. Introduction

Wearable robots, among which the ones for rehabilitation are probably the most important recent novelty in the medical field, offer great help to patients with lower limb impairments. The exoskeletons play an important role in helping patients regain mobility and improve their range of motion (ROM). However, optimization of the control mechanism in lower limb rehabilitation robots is still among the most essential challenges. The typical feedback-based control strategies often

exhibit shortages in meeting the various needs of patients and the complex dynamics of human gait patterns. Consequently, the above deficiency hinders the effectiveness of ROM training and slows down rehabilitation progress.

The problem is based on the essential inflexibility and low adaptability of traditional control strategies that do not take sufficient account of the individual differences in patient mobility and recovery courses. Besides, the complexity and dynamism in human gait patterns call for advanced control algorithms that would adapt to real-time situations. Overcoming these challenges is very important for improving the effectiveness of rehabilitation robots, making them sensitive to the requirements of each patient and consequently speeding up the process of rehabilitation. The aim of this research is to investigate and implement state-of-the-art control mechanisms to overcome these limitations and thus improve the overall outcome of ROM training for patients using wearable rehabilitation robots.

In dealing with these issues, recent studies in control strategies for lower limb rehabilitation exoskeletons have emphasized a range of methodologies aimed at increasing patient engagement, safety, and effectiveness in training. On the other hand, as suggested by Tu et al. [1] and Shen et al. [2], adaptive control techniques have resulted in huge improvements in terms of accuracy and personalization but still have some barriers in terms of implementation due to complexity and require deeper clinical verification. Similarly, the adaptive fuzzy control systems, studied by Zhang et al. [3] and Gong et al. [4], offer better position tracking and safety but suffer from a huge number of real-world tests to cope with their inherent complexity.

Model-based methods, as represented by the works of Alotaibi and Alsubaie [5] and Jin and Guo [6], utilize model predictive control techniques to enhance the level of adaptiveness and robustness but encounter problems related to computational requirements and real-time implementation. Sliding mode control methodologies, as highlighted by Faraj et al. [7] and Wang et al. [8], show improved performances in trajectory tracking and disturbance management, but the complexities involved and chattering effect pose significant challenges. Multimodal adaptive control techniques, such as those proposed by Liang et al. [9] and Zhou et al. [10], can effectively change control parameters in real time to enhance patient safety and engagement; however, they require thorough evaluation for effective practical realization.

Comprehensive reviews by Tijjani et al. [11], Long et al. [12], Li et al. [13], and Zhou et al. [14] underline the development and challenges of different control strategies, highlighting the need for improved real-time response to, and adaptation of, the individual patient. Specialized control methodologies, such as those exemplified by the works of Chen et al. [15] and Sun et al. [16], and optimization-driven approaches, as introduced by Soleimani Amiri et al. [17], Liu et al. [18], and Yacef [19], give way to substantial improvements in trajectory tracking and robustness but face limitations regarding practical implementation and computational complexities. Additional reviews by Sharma et al. [20] and Yao et al. [21] have presented the development and ongoing challenges involved in integrating adaptive, fuzzy, and model-based controls with human biomechanics to achieve effective exoskeleton systems. Novel approaches by Yu et al. [22] and Bauer and Pan [23] emphasize the need for environmental interaction and robust methods to ensure safe and effective rehabilitation.

Recurrent neural networks (RNNs) represent an important class of intelligent learning-based methodologies for identifying and controlling complex nonlinear dynamical systems across numerous engineering domains. Leveraging inherent memory mechanisms through recurrent connections, RNNs effectively capture temporal dynamics in systems characterized by significant uncertainties, parametric variations, and unmodeled effects—phenomena prevalent in rehabilitation robotics, aerospace systems, and industrial automation [24].

Recent literature has produced several significant advances in neural network-based identification architectures. Self-recurrent wavelet neural networks (SRWNN) combine wavelet multiresolution analysis with recurrent modeling capabilities to enable simultaneous system identification and predictive control, with demonstrated efficacy in robotic manipulator dynamics and nonlinear process control applications. Novel RNN structures incorporating Lyapunov-based stability analysis provide theoretical convergence guarantees and bounded identification errors, addressing critical requirements in safety-critical domains such as biomedical and aerospace control. Furthermore, robust RNN architectures emphasizing noise resilience through adaptive learning algorithms, dead-zone modifications, and parameter projection techniques ensure accurate identification under adverse operational conditions in real-world industrial settings.

Despite these advances, RNN-based approaches present substantial challenges for rehabilitation robot implementation. Training requirements demand extensive computational resources and large datasets often unavailable in clinical settings with patient-specific variability. The inherent "black-box" nature of neural networks complicates theoretical stability certification and safety validation—prerequisites for medical device regulatory approval. Online adaptation stability and convergence properties require careful consideration during rehabilitation sessions, while clinical acceptance remains limited due to practitioners' preference for transparent, mathematically proven control strategies [25].

In contrast, the proposed Model Reference Adaptive Control (MRAC)-Particle Swarm Optimization (PSO) framework strategically combines established adaptive control theory with systematic optimization to address these limitations. This hybrid approach provides Lyapunov-guaranteed stability, globally optimal parameter selection without manual tuning, real-time patient-specific adaptation without retraining, interpretable controller parameters with explicit performance relationships, and minimal computational overhead suitable for embedded platforms [26]. While RNN methodologies excel for systems exhibiting high complexity with minimal prior knowledge, MRAC-PSO architecture proves particularly appropriate for wearable rehabilitation devices requiring model-based design, provable stability, clinical interpretability,

and real-time performance. The framework effectively bridges classical control theory and contemporary optimization techniques, delivering a practically deployable solution satisfying both engineering performance and clinical safety requirements [27].

The primary objective of this study is to develop and evaluate the MRAC-PSO controller for lower limb rehabilitation robots. By leveraging the adaptive capabilities and optimization features of the MRAC-PSO system, this research aims to enhance the adaptability and efficiency of ROM training, thus providing patients with a more effective rehabilitation experience. To highlight the advantages and potential improvements offered by the adaptive control approach, the performance of the MRAC-PSO controller will be evaluated against the traditional PID controller, tuned using the Ziegler–Nichols (Z–N) method, and the conventional MRAC controller.

This research article seeks to contribute to the field of wearable robotics by proposing and assessing an innovative MRAC-PSO control strategy for lower limb rehabilitation robots. By integrating MRAC-PSO controllers and comparing their performance with conventional MRAC and PID controllers, the study intends to overcome the existing limitations in ROM training and offer a more adaptive and efficient rehabilitation solution. Ultimately, the outcomes of this research are expected to pave the way for future advancements in rehabilitation technology, thereby improving the quality of life for individuals with lower limb impairments.

While adaptive control and optimization techniques have been explored individually in rehabilitation robotics, this study presents several novel contributions that advance the state-of-the-art in lower limb rehabilitation control as follows:

- Novel integration of PSO-optimized MRAC for wearable lower limb rehabilitation (WLLR) ROM training: this article represents the first systematic application of PSO to tune MRAC parameters specifically for ROM training in WLLR robots. Unlike previous studies that applied PSO to conventional PID controllers or used MRAC without optimization, we integrate both techniques to leverage the adaptive capabilities of MRAC and the global optimization strengths of PSO simultaneously.
- Comprehensive control framework for complex WLLR dynamics: we develop a complete mathematical framework that integrates the detailed dynamic model of the WLLR system (including coupled hip–knee dynamics, DC motor equations, and gearbox mechanics) with the MRAC-PSO controller. This end-to-end framework addresses the full complexity of wearable robot dynamics, unlike simplified models in prior work.
- Rigorous quantitative performance benchmarking: this study provides systematic comparative analysis across three distinct control strategies (PID-ZN, MRAC, and MRAC-PSO) using standardized performance metrics (risetime, settling time, overshoot, and steady-state error [SSE]) for both hip and knee joints under step and sinusoidal reference trajectories. This comprehensive benchmarking quantifies the specific advantages of MRAC-PSO over conventional approaches with unprecedented detail.
- Significant performance improvements with clinical implications: the MRAC-PSO controller achieves substantial performance gains: knee joint step response shows an 89% reduction in risetime (0.07 s vs. 0.64 s for PID-ZN) and a 98.9% reduction in SSE (0.005 vs. 0.45 for PID-ZN). These improvements translate to more accurate, faster, and safer rehabilitation movements, directly impacting patient safety and therapy effectiveness.
- Modular control strategy for multiconfiguration WLLR: the proposed control framework is designed to accommodate the modular nature of the WLLR device across its five configurations (C1–C5), with detailed validation for ROM training configurations C3 and C4. This modular approach enables future extension to walking gait training and multijoint coordination tasks.

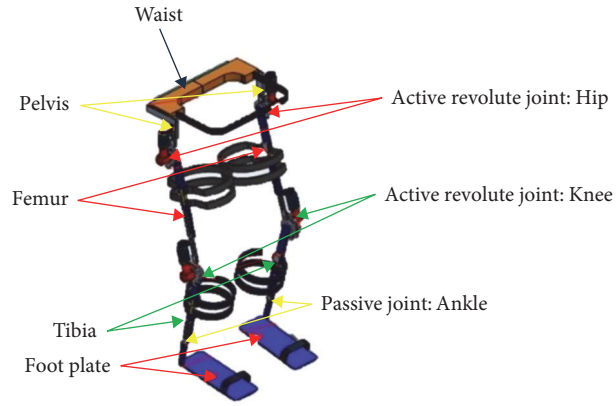

Figure 1: WLLR design [28].

Unlike prior works that either focused on simplified control methods, lacked systematic optimization, or did not provide comprehensive comparative analysis, our integrated MRAC-PSO framework addresses multiple limitations simultaneously while demonstrating measurable improvements across all key performance indicators.

The rest of this article is organized as follows: Section 2 introduces the WLLR design of a lower limb exoskeleton rehabilitation device. Section 3 introduces the dynamic model of the WLLR device. Section 4 introduces the control system design. Section 5 is the result and discussion of the main findings and simulation results, and finally, section 6 summarizes this article.

2. WLLR Design

The modular design is incorporated into the WLRR as illustrated in Figure 1. This approach enables the WLLR to execute several fundamental lower limb treatment movements, including ROM exercises and walking gait therapy. These movements can be facilitated by adjusting the robot's

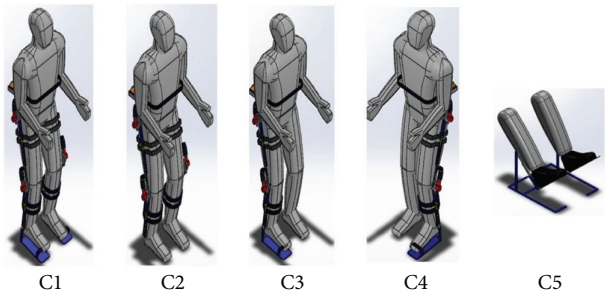

Figure 2: WLLR configurations [28].

configuration or utilizing external accessories. The proposed WLLR can be set up in five main configurations, as depicted in Figure 2. Configurations C1 and C2 are designed for walking gait training, with or without passive ankle involvement, respectively. This training includes hip and knee joint flexion/extension in both passive and active modes. Configurations C3 and C4 focus on ROM training for the hip and knee joints of the right and left legs, respectively, which is crucial for enhancing the patients' ROM before progressing to active training. Configuration C5 is dedicated to passive ankle training, emphasizing dorsiflexion and plantarflexion.

3. Dynamic Model

3.1. Dynamic Model of WLRR Structure. The WLLR is a six-DOF mechanism with two active hip joints, two active knee joints, and two passive ankle joints. Only the active revolute joints are explored when performing kinematic analysis. Because of the identical kinematic structures of each leg, only one leg is analyzed. The sagittal-plane free-body diagram of the WLLR system, according to these premises, is shown in Figure 3. O1 and O2 indicate the positions of the hip and knee axes in this schematic, respectively. l1 and l2 describe the lengths of the femur and tibia, d1/d2 are the distances from O1/O2 to the femur/tibia's center of gravity (CG), respectively. m1 and m2 indicate the femur and tibia masses, and I1 and I2 are their moments of inertia. The Lagrangian formulation generates joint torques as shown in Equations (1) and (2).

While E_p and E_k represent the total potential and kinetic energy, respectively, the relationship between E_p and E_k in the Lagrangian function is expressed as follows:

$$L_a = E_p - E_k, \tag{1}$$

$$T_{li} = \frac{d}{dt}\left(\frac{\partial L_a}{\partial \dot{\theta}_i}\right) - \left(\frac{\partial L_a}{\partial \theta_i}\right), \tag{2}$$

$$E_p = \sum_{i=1}^{2} m_i g y_i, \tag{3}$$

$$E_k = \sum_{i=1}^{2}\left[\frac{1}{2}m_i(\dot{x}_i^2 + \dot{y}_i 2) + \frac{1}{2}I_i\dot{\theta}_i^2\right], \tag{4}$$

where m_i and I_i represent the mass and inertia of each femur and tibia, and g denotes the gravitational acceleration. The

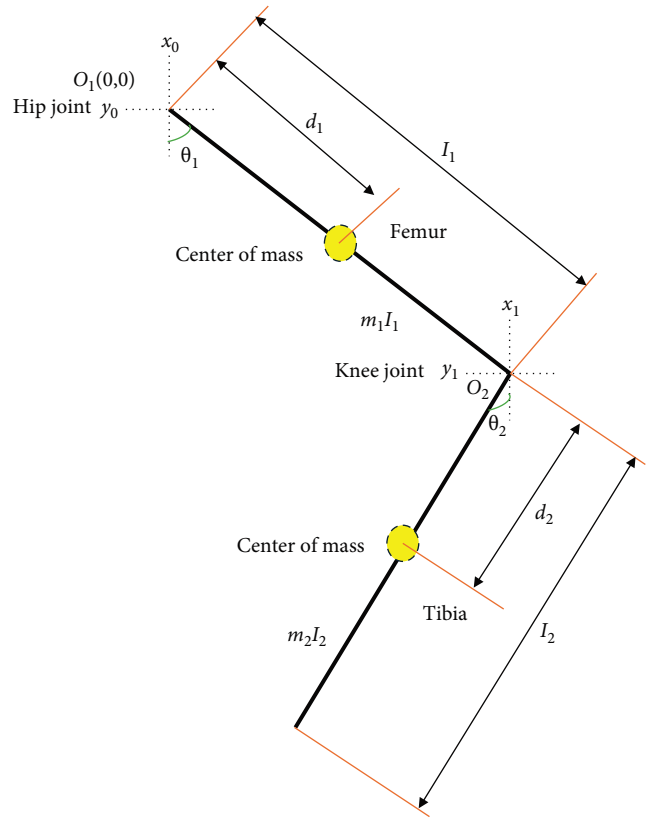

Figure 3: Free body diagram of the WLLR structure [28].

location of the CG is represented by coordinates (x_i, y_i), which are derived through geometric relations as presented in the following equation:

$$x_i = \sum_{j=1}^{i-1} l_j \sin(\theta_i) + d_i \sin(\theta_i), \tag{5}$$

$$y_i = \sum_{j=1}^{i-1} - l_j \cos(\theta_i) - d_i \cos(\theta_i). \tag{6}$$

The variable l_j represents the length of either the femur or tibia, while d_i denotes the distance from the femur or tibia to the CG. The derivative of the equation mentioned above is expressed in Equations (7) and (8), illustrating the relationship between these parameters and their impact on the system's dynamics.

$$\dot{x}_i = \sum_{j=1}^{i-1} I_j\dot{\theta}_j(\cos\theta_j) + d_i\dot{\theta}_i(\cos\theta_i), \tag{7}$$

$$\dot{y}_i = \sum_{j=1}^{i-1} I_j\dot{\theta}_j(\sin\theta_j) + d_i\dot{\theta}_i(\sin\theta_i), \tag{8}$$

where $\dot{x}_i$ and $\dot{y}_i$ represent the angular velocities of the CG. By utilizing Equation (1) through Equation (8), the torque equations for the hip and knee joints are derived and obtained as follows:

$$T_{l1} = \begin{aligned}&(m_1 d_1^2 + I_1 + m_2 l_1^2)\ddot{\theta}_1 \\ &+m_2 l_1 d_2 \cos(\theta_2 - \theta_1)\ddot{\theta}_2 \\ &+m_2 l_1 d_2 \dot{\theta}_2^2 \sin(\theta_2 - \theta_1) \\ &+(m_1 g d_1 + m_2 g l_1)\sin(\theta_1).\end{aligned}, \tag{9}$$

$$T_{l2} = \begin{aligned}&(m_2 d_2^2 + I_2)\ddot{\theta}_2 \\ &+(m_2 l_1^2 + m_2 l_1 d_2 \cos(\theta_2 - \theta_1))\ddot{\theta}_1 \\ &+m_2 l_1 d_2 \dot{\theta}_1^2 \sin(\theta_2 - \theta_1) \\ &+m_2 g d_2 \sin(\theta_2).\end{aligned} \tag{10}$$

To linearize and simplify the torques of the hip (T_{l1}) and knee (T_{l2}), the WLLR joint performance is analyzed under ROM conditions, where one joint move while the other remains fixed [21, 22, 24, 25]. When the knee joint moves, its position (θ_2), angular velocity ($\dot{\theta}_2$), and angular acceleration ($\ddot{\theta}_2$) are all zero. Similarly, when the hip joint moves, its position (θ_1), angular velocity ($\dot{\theta}_1$), and angular acceleration ($\ddot{\theta}_1$) are zero. Under these conditions, the torque equations for the hip (T_1) and knee (T_2) are simplified, resulting in the following forms of Equations (9) and (10):

$$T_{l1} = (m_1 d_1^2 + I_1 + m_2 l_1^2)\ddot{\theta}_1 + (m_1 g d_1 + m_2 g l_1)\sin(\theta_1), \tag{11}$$

3.2. Dynamic Model of WLRR System. A DC brushed motor is selected as the actuator for the WLLR's hip and knee joints due to its required power and its compact, portable design, making it an ideal solution for a wearable robot. To enhance performance, a gearbox is attached to the motor's rotary shaft. This gearbox reduces the angular velocity while increasing the torque output. The output of the gearbox then rotates the shaft that is fixed to the WLLR link. The motor's components, which include resistance and inductance, facilitate the conversion of electrical energy into mechanical energy, thereby providing the necessary torque for joint rotation. The DC motor's mathematical model, derived using Kirchhoff's law, is formulated as follows:

$$U_i - V_{\text{emfi}} = R_{\text{mi}} i_i + L_{\text{mi}}\left(\frac{di}{dt}\right) i = 1,\ 2, \tag{12}$$

where L_{mi} represents the motor inductance, R_{mi} denotes the resistance, and i is the motor current. The input voltage to the circuit is U_i, while V_{emfi} is the back electromotive force voltage, which is proportional to the angular velocity of the motor shaft. This relationship is illustrated in Equation (13).

$$V_{\text{emfi}} = K_{\text{bi}}\dot{\theta}_{\text{shfi}}\ i = 1,\ 2. \tag{13}$$

K_{bi} is the motor voltage constant and $\dot{\theta}_{shfi}$ is the angular velocity of the gear shaft. The output of the motor is a torque which rotates the shaft proportional to the current, i as in Equation (14)

$$T_{\text{mi}} = K_{\text{mi}} i_i\ i = 1,\ 2, \tag{14}$$

where K_{mi} is motor torque sensitivity, while T_{mi} is the torque applied to the motor. By substituting Equations (13) and (14) into Equation (27), the mathematical model of the motor in Laplace is represented in Equation (15).

$$U_i(s) - K_{\text{bi s}}\theta_{\text{shfi}}(s) = ((L_{\text{mi s}} + R_{\text{mi}})T_{\text{mi}}(s))/K_{\text{mi}} \tag{15}$$

T_{mi} can be obtained based on the ratio of the gear correlates with the input and output torque is given in Equation (31).

$$r_g = \frac{T_{\text{out}}}{T_{\text{in}}} = \frac{N_{\text{out}}}{N_{\text{in}}}, \tag{16}$$

where r_g is the gearbox ratio, T_{out} and T_{in} are the output and input torque of the gearbox and motor. Thus, N_{out} and N_{in} are the numbers of teeth for output and input gears. The output and input torque of the joint base in Figure 4 is derived as in the following equation.

$$T_{\text{ini}} = T_{\text{mi}}, \tag{17}$$

$$T_{\text{outi}} = T_{\text{shfi}} + T_{\text{li}}. \tag{18}$$

T_{shfi} is the torque of the gearbox shaft while T_{mi} is torque generated by the DC motor. T_{li} represents the torque of femur and tibia link where ($i = 1,2$) which is 1 for the hip and 2 for the knee. By substituting Equations (17) and (18) into Equation (16), the torque of the gearbox is written as Equation (19), and the Laplace is written in Equation (20).

$$T_{\text{mi}} = \frac{1}{r_g}(T_{\text{shfi}} + T_{\text{li}}). \tag{19}$$

$$T_{\text{mi}}(s) = \frac{1}{r_g}(T_{\text{shfi}}(s) + T_{\text{li}}(s)), \tag{20}$$

where T_{li} is Laplace of Equation (11) which is the torque for the hip and knee, respectively is written as follow:

$$T_{l1}(s) = \begin{aligned}&(m_1 d_1^2 + I_1 + m_2 l_1^2)s^2\theta_{shf1}(s) \\ &+ (m_1 g d_1 + m_2 g l_1)\theta_{shf1}(s),\end{aligned} \tag{21}$$

$$T_{l2}(s) = (m_2 d_2^2 + I_2)s^2\theta_{\text{shf2}}(s) + (m_2 g d_2)\theta_{\text{shf2}}(s). \tag{22}$$

The dynamic equations of the gear shaft for hip and knee joint T_{shfi} are expressed as in Equations (23) and (24), and the Laplace is shown in Equations (25) and (26), respectively.

$$T_{\text{shf1}} = J_{\text{shf1}}\ddot{\theta}_{\text{shf1}} + B_{\text{shf1}}\dot{\theta}_{\text{shf1}}, \tag{23}$$

$$T_{\text{shf2}} = J_{\text{shf2}}\ddot{\theta}_{\text{shf2}} + B_{\text{shf2}}\dot{\theta}_{\text{shf2}}, \tag{24}$$

$$T_{\text{shf1}}(s) = J_{\text{shf1}}s^2\theta_{\text{shf1}}(s) + B_{\text{shf1}}s\theta_{\text{shf1}}(s), \tag{25}$$

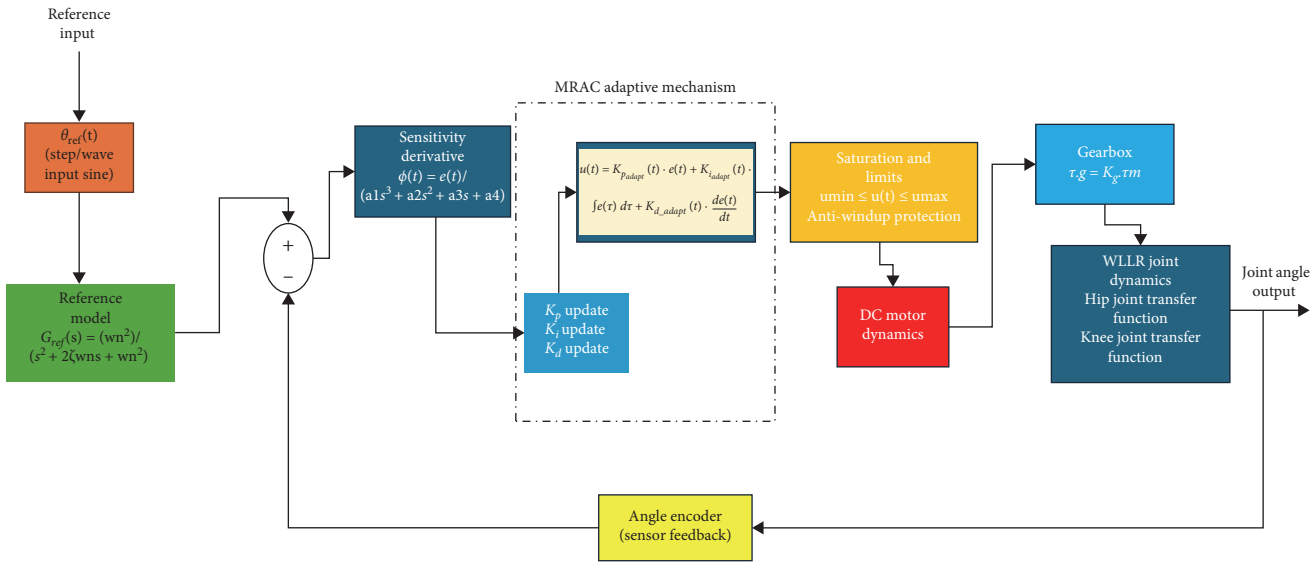

FIGURE 4: Detailed block diagram of the proposed MRAC controller.

$$T_{shf2}(s) = J_{shf2}(s^2)\theta_{shf2}(s) + B_{shf2}s\theta_{shf2}(s). \tag{26}$$

By substituting Equations (21) and (24) into Equation (20), $T_{m1}(s)$ can be obtained as follow:

$$\begin{aligned} T_{m1}(s) = &\left(\frac{1}{r_g}(J_{shf} + m_1 d_1^2 + I_1 + m_2 l_1^2)\theta_{shf1}(s)\right)s^2 \\ &+\left(\frac{1}{r_g}B_{shf}\theta_{shf1}(s)\right)s \\ &+\frac{1}{r_g}(m_1 g d_1 + m_2 g l_1)\theta_{shf1}(s). \end{aligned} \tag{27}$$

Similarly $T_{m2}(s)$ can be obtained by substituting Equations (22) and (26) into Equation (20) as follows:

$$\begin{aligned} T_{m2}(s) = &\left(\frac{1}{r_g}(J_{shf} + m_2 d_2^2 + I_2)\theta_{shf2}(s)\right)s^2 \\ &+\left(\frac{1}{r_g}B_{shf}\theta_{shf2}(s)\right)s \\ &+\frac{1}{r_g}(m_2 g d_2)\theta_{shf2}(s). \end{aligned} \tag{28}$$

The transfer function of the hip joint under ROM condition when one joint is moving, and the other joint is fixed can be obtained by a substitute Equation (27) into Equation (15) and the results obtained as follow:

$$U_1(s) = \frac{\theta_{shf1}}{K_m r_g}\begin{bmatrix} \left(Ll_1^2 m_2 + Ld_1^2 m_1 + LI_1 + LJ_{shf}\right)s^3 \\ +\left(Rl_1^2 m_2 + Rd_1^2 m_1 + RJ_{shf}\right. \\ \left.+RI_1 + LB_{shf1}\right)s^2 \\ +\left({}_1Lgm_2 + Ld_1 gm_1 + RB_{shf} + K_b r_g^2 K_m\right)s \\ +Rl_1 gm_2 \\ +Rd_1 gm_1 \end{bmatrix}, \tag{29}$$

$$U_1(s) = \frac{\theta_{shf1}(z_{11}s^3 + z_{12}s^2 + z_{13}s + z_{14})}{y}, \tag{30}$$

where:

$$y = K_m r_g, \tag{31}$$

$$z_{11} = Ll_1^2 m_2 + Ld_1^2 m_1 + LI_1 + LJ_{shf}, \tag{32}$$

$$z_{12} = Rl_1^2 m_2 + Rd_1^2 m_1 + RJ_{shf} + RI_1 + LB_{shf}, \tag{33}$$

$$z_{13} = l_1 Lgm_2 + Ld_1 gm_1 + RB_{shf} + K_b r_g^2 K_m, \tag{34}$$

$$z_{14} = Rl_1 gm_2 + Rd_1 gm_1. \tag{35}$$

Similar for knee, the transfer function can be obtained by substituting Equation (28) into Equation (15) and the equation is shown as follows:

$$U_1(s) = \frac{\theta_{\text{shf2}}}{K_m r_g} \begin{bmatrix} (Ll_2^2 m_2 + LJ_{\text{shf}} + LI_2)s^3 \\ +(Rd_2^2 m_2 + RJ_{\text{shf}} \\ +RI_2 \\ +LB_{\text{shf}})s^2 + \\ (Ld_2 g m_1 + RB_{\text{shf}} + \\ K_b r_g^2 K_m)s + Rd_2 g m_2 \end{bmatrix}, \tag{36}$$

$$U_2(s) = \frac{\theta_{\text{shf2}}(z_{21}s^3 + z_{22}s^2 + z_{23}s + z_{24})}{y}, \tag{37}$$

$$y = K_m r_g, \tag{38}$$

$$z_{21} = Ll_2^2 m_2 + Ld_2^2 m_2 + LJ_{shf} + LI_2, \tag{39}$$

$$z_{22} = Rd_2^2 m_2 + RJ_{shf} + RI_2 + LB_{shf}, \tag{40}$$

$$z_{23} = Ld_2 g m_1 + RB_{shf} + K_b r_g^2 K_{mm}, \tag{41}$$

$$z_{24} = Rd_2 g m_2. \tag{42}$$

Finally, the transfer function of hip and knee joints can be expressed, as in Equations (43) and (44), respectively.

$$G_1(s) = \frac{\theta_{shf1}(s)}{U_1(s)} = \frac{y}{z_{11}s^3 + z_{12}s^2 + z_{13}s + z_{14}}, \tag{43}$$

$$G_2(s) = \frac{\theta_{shf2}(s)}{U_1(s)} = \frac{y}{z_{21}s^3 + z_{22}s^2 + z_{23}s + z_{24}}. \tag{44}$$

4. Control Design

4.1. Model Reference Adaptive Controller. Adaptive control encompasses techniques used to manage systems with parametric uncertainty, meaning systems with uncertain dynamics or parameters that change unpredictably overtime. Adaptive controllers adjust their behavior in response to process dynamics and disturbances, proving advantageous when variations cannot be measured. MRAC systems are particularly useful when performance specifications are provided through a reference model. Figure 4 illustrates a detailed block diagram of the MRAC-PSO control system for WLLR, showing all subsystems, including the reference model, MRAC adaptation mechanism, PID controller, motor dynamics, gearbox, WLLR joint dynamics, and feedback paths. Signal labels indicate: θref (reference trajectory), θm (model output), θ (actual joint angle), *e* (tracking error), φ (sensitivity derivative), *u* (control signal), τm (motor torque), and τg (gearbox torque).

In this article, the MRAC approach is utilized for designing an adaptive control system. An MRAC system includes a plant, an adjustment mechanism for real-time controller parameter tuning, and a model reference that defines the desired behavior of the control system [29, 30]. The PID controller's initial values are derived from the closed-loop control system. This setup ensures that the controller can adapt to changing dynamics and disturbances, maintaining optimal performance by continually aligning the actual system behavior with the ideal behavior specified by the reference model. The integration of these components facilitates robust and flexible control in uncertain environments.

θ_m represents the trajectory of the model reference, while $\acute{e}$ denotes the tracking error, which is the difference between θ_m and the actual trajectory.

$$\acute{e} = \theta_{act} - \theta_m, \tag{45}$$

The model reference is designed to produce the optimal trajectory of the joints, matching the order of the closed-loop transfer function.

$$G_m(s) = \frac{s^2 + b_{m_1}s + b_{m_2}}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}}, \tag{46}$$

The parameters of the tuned PID controller are adjusted by a mechanism to minimize the tracking error. The MIT rule, an effective method based on a gradient approach, is used to achieve this minimization [31, 32]. The cost function is defined as follows:

$$J(K_j) = \frac{\acute{e}^2}{2} j = p, i, d \tag{47}$$

K_j is a parameter of the PID controller that needs to be adjusted to minimize the cost function. Hence, K_j is modified in the negative direction of the gradient of $J(K_j)$.

$$\frac{dK_j}{dt} = -\gamma \frac{\partial J}{\partial K_j}, \tag{48}$$

$$\frac{dK_j}{dt} = -\gamma_j e' \frac{\partial e'}{\partial K_j} = -\gamma_i e' \frac{\partial \theta_{act}}{\partial K_j}, \tag{49}$$

where $\frac{\partial e'}{\partial K_j}$ is sensitivity derivatives of tracking error; y represents speed of adaption. θ_{act} is given as follows:

$$\theta_{act} = \frac{G_i C(s)}{1 + G_i C(s)} \theta_{des}, \tag{50}$$

The symbolic transfer function of *Gi* can be expressed as follows:

$$G_i = \frac{b}{s^3 + a_1 s^2 + a_2 s + a_3}. \tag{51}$$

By substituting Equation (52): $C(s) = \frac{K_p s + K_i + K_d s^2}{s}$, and Equation (51) in Equation (50), θ_{act} is written as follows:

$$\theta_{act} = \frac{b\left(K_d s^2 + K_p s + K_i\right)}{s^4 + a_1 s^3 + (a_2 + bK_d)s^2 + \left(a_3 + bK_p\right)s + bK_i}\theta_{des}. \tag{52}$$

Consequently, by assuming:

$$a_{m1} = a_1 a_{m2} = a_2 + bK_d a_{m3} = a_3 + bK_p, \tag{53}$$

$$a_{m4} = bK_p s + bK_i. \tag{54}$$

The partial of θ_{act} over K_j while $j = p, i, d$; d is given as follows:

$$\frac{\partial \theta_{act}}{\partial K_p} = \frac{bes}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}}, \tag{55}$$

$$\frac{\partial \theta_{act}}{\partial K_i} = \frac{be}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}}, \tag{56}$$

$$\frac{\partial \theta_{act}}{\partial K_d} = \frac{bes^2}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}}. \tag{57}$$

By substitution of Equations (55)–(57) in Equation (49) and its integral in Laplace, the parameters of PID controller are determined as follows:

$$K_P = -\gamma_p e' \frac{bes}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}} + c_p, \tag{58}$$

$$K_i = -\gamma_i e' \frac{be}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}} + c_i, \tag{59}$$

$$K_d = -\gamma_d e' \frac{bes^2}{s^4 + a_{m_1}s^3 + a_{m_2}s^2 + a_{m_3}s + a_{m_4}} + c_d, \tag{60}$$

where cp, ci, and cd represent the initial values of the PID parameters, chosen from the tuned PID controller using the Z–N method. Selecting these initial values helps to reduce both the error and computational time.

4.2. MRAC Controller Tuning Using PSO. PSO is an iterative optimization technique inspired by biological groups such as flocks of birds, schools of fish, and swarms of bees [33]. It is straightforward to implement for various optimization problems, and its parameter robustness is manageable [34]. Additionally, PSO offers rapid convergence and a low likelihood of becoming trapped in a local set of candidate solutions [35]. The optimization parameters are adjusted by increasing the number of iterations, enhancing the chances of finding the global optimum. Equation (61) represents the position of the particles in the next iteration, calculated by summing the positions of the previous particles with their respective velocities.

1: Start.
2: Random initialization of the first population with 100 particles.
3: Evaluation and sorting.
4: Set the initial pbest
5: ***while*** *Number of iterations < 150* ***do****.*
6: Generate new iteration.
7: Obtain the pbest;
8: ***if*** *pbest < gbest* ***then****.*
9: Set it as gbest
10: ***end if***
11: ***end while***
12: Select final gbest as the result.
13: End

Algorithm 1: PSO pseudocode.

$$x_{i,j} = x_{i-1,j} + v_{i,j}, \tag{61}$$

where j and i denote the particle index and iteration number, respectively; $x_{i-1,j}$ represents the position of the particles in the previous iteration, and $v_{i,j}$ is the velocity and direction of the current particle moving toward the next iteration, expressed as follows.

$$v_{i,j} = \omega_i v_{i-1,j} + C_1\zeta_1\left(p_{best,i-1} - x_{i,j}\right) - C_2\zeta_2\left(g_{best} - x_{i,j}\right), \tag{62}$$

where ζ_1 and ζ_2 are random values established between 0 and 1. c_1 and c_2 are positive coefficients representing the self-recognition and social components, respectively.

$p_{best,i}$ and g_{best} represent the best position of each particle and the global best position of the entire population, respectively. ω_i denotes the inertia weight, which is readjusted with each iteration, as given below:

$$\omega_i = \omega_d \omega_{i-1}, \tag{63}$$

where ω_d is the damping value. The objective function is established for evaluation purposes. In each iteration, the particle with the lowest objective function value is selected as $p_{best,i}$. After evaluation, the lowest $p_{best,i}$ is identified as the global best, g_{best}. Algorithm 1 presents the pseudocode of the PSO.

4.2.1. Hyperparameter Configuration and Initial Value Selection. The performance of the MRAC-PSO controller is significantly influenced by the proper selection of hyperparameters for both the PSO optimization algorithm and the MRAC adaptation mechanism. This section provides a comprehensive discussion of all hyperparameters, their selected values, the rationale behind these choices, and sensitivity analysis to ensure robustness and reproducibility.

Table 1: PSO hyperparameters and selection rationale.

Hyperparameter	Symbol	Selected value	Selection rationale
Population size	N_p	100 particles	Balances exploration capability with computational cost. Values < 50 risk premature convergence; values > 150 increase computation time without significant performance gain. Standard practice for four-parameter optimization.
Cognitive coefficient	c_1	2.0	Controls particle's tendency toward its personal best position. Standard value $c_1 =$ 2.0 recommended by Kennedy and Eberhart provides balanced self-recognition.
Social coefficient	c_2	2.0	Controls particle's tendency toward global best position. Setting $c_1 = c_2 = 2.0$ ensures equal weighting between personal and social learning.
Initial inertia weight	w_{init}	0.9	High initial inertia promotes global exploration in early iterations. Standard range $w \in [0.4, 0.9]$; starting at 0.9 maximizes early exploration.
Final inertia weight	w_{final}	0.4	Low final inertia promotes local exploitation for fine-tuning. Linearly decreases from 0.9 to 0.4 over iterations using Equation (63).
Damping ratio	α	$\frac{w_{init}-w_{final}}{N_i}$	Linear decrease of inertia weight computed as $w(k) = w_{init} - \alpha \cdot k$ where k is iteration number. Provides smooth transition from exploration to exploitation.
Maximum iterations	N_i	150 iterations	Ensures convergence based on preliminary trials. Convergence typically achieved by iteration 120–130; 150 iterations provide safety margin. Consistent with literature values.

4.2.1.1. PSO Algorithm Hyperparameters. The PSO optimization algorithm involves several critical hyperparameters that govern its exploration–exploitation balance, convergence behavior, and computational efficiency. Table 1 (Renumber your existing tables accordingly) summarizes all PSO hyperparameters used in this study with their selected values and justification.

Detailed justification for PSO hyperparameters:

1. Population size ($N_p = 100$): the population size directly impacts the algorithm's ability to explore the search space and avoid local optima. Preliminary sensitivity analysis (discussed in Section 4.2.1.4) tested values of 50, 75, 100, 125, and 150 particles. Results showed that:
   - $N_p = 50$: premature convergence to suboptimal solutions in 35% of trials.
   - $N_p = 75$: acceptable convergence but less robust to different initial conditions.
   - $N_p = 100$: optimal balance—consistent convergence to global optimum with reasonable computational time (~60 min).
   - $N_p > 100$: marginal performance improvements (<2%) with 40%–50% increase in computation time.

   Therefore, $N_p = 100$ was selected as the optimal choice, consistent with similar studies.
2. Maximum iterations ($N_i = 150$): convergence analysis showed that the global best objective function value stabilized (change < 0.1%) by iteration 120–135 across multiple runs. Setting $N_i = 150$ provides a safety margin ensuring full convergence while avoiding excessive computation. Figure X (can be added) shows typical convergence curves for hip and knee joint optimization.
3. Cognitive and social coefficients ($c_1 = c_2 = 2.0$): these parameters control the balance between individual exploration (cognitive component) and swarm collaboration (social component). The widely accepted values $c_1 = c_2 = 2.0$, originally recommended by Kennedy and Eberhart, ensure equal weighting and have been validated across numerous applications. Alternative configurations ($c_1 = 2.5, c_2 = 1.5$ favoring personal best, or $c_1 = 1.5, c_2 = 2.5$ favoring global best) were tested but showed no significant improvement over the standard configuration.
4. Inertia weight schedule ($w: 0.9 \rightarrow 0.4$): time-varying inertia weight is critical for PSO convergence. High initial inertia ($w = 0.9$) allows particles to explore the search space broadly, while low final inertia ($w = 0.4$) enables fine-tuning around the global optimum. The linear decrease schedule (Equation (63)) provides a smooth transition and is simpler than adaptive nonlinear schedules while achieving comparable performance.

4.2.1.2. Search Space Bounds for PID and Adaptation Gain Parameters. The PSO algorithm searches for optimal values of four controller parameters: proportional gain (K_p), integral gain (K_i), derivative gain (K_d), and MRAC adaptation gain (γ). Defining appropriate search space bounds is crucial to ensure:

- Physically meaningful parameter values.
- Controller stability.
- Computational efficiency (avoiding exploration of infeasible regions).

Table 2 presents the search space bounds for all optimization parameters.

Rationale for parameter bounds:

PID gains (K_p, K_i, K_d): the bounds were established based on:

1. Z–N's baseline: initial PID parameters tuned using the Z–N method (Table 1) provide central reference values within the search space.

Table 2: Search space bounds for optimization parameters.

Parameter	Symbol	Lower bound	Upper bound	Rationale for bounds
Proportional gain	K_p	0.1	50	Lower bound prevents negligible control action; upper bound prevents excessive gain causing instability and actuator saturation. Based on Ziegler–Nichols ultimate gain range.
Integral gain	K_i	0.01	30	Lower bound ensures steady-state error elimination; upper bound prevents integral windup and overshoot. Scaled relative to K_p -based on typical T_i range.
Derivative gain	K_d	0.001	10	Lower bound allows nonzero derivative action; upper bound prevents noise amplification and high-frequency oscillations. Conservative to avoid sensor noise sensitivity.
Adaptation gain	γ	−5.0	−0.001	Negative values ensure stability per Lyapunov analysis (MIT rule). Magnitude controls adaptation speed; too large causes oscillations, too small causes slow adaptation.

2. Stability margins: Nyquist and Bode analysis of the WLLR closed-loop system (transfer functions Equations (43) and (44) determined gain ranges, ensuring a phase margin $> 45°$ and a gain margin > 6 dB.
3. Physical constraints: maximum motor torque (Table 2) and gearbox limits constrain maximum allowable gains to prevent actuator saturation.
4. Literature validation: similar rehabilitation robot studies reported PID gains in comparable ranges.

Adaptation gain (γ): the adaptation gain determines the speed of MRAC parameter adjustment. The MIT rule (Equations (48) and (49) requires $\gamma > 0$ in its derivation, but the implementation uses γ with sign convention, resulting in negative values in Table 1. The magnitude of γ affects:

- Adaptation speed: larger $|\gamma| \rightarrow$ faster adaptation but risk of oscillations.
- Stability: Lyapunov stability theory requires bounded γ.
- Robustness: smaller $|\gamma| \rightarrow$ slower adaptation but better noise rejection.

Preliminary testing showed:

- $|\gamma| < 0.001$: Adaptation too slow, insufficient tracking improvement.
- $|\gamma| > 5.0$: High-frequency oscillations, potential instability.
- Optimal range: $|\gamma| \in [0.01, 1.0]$ for most rehabilitation trajectories.

4.2.1.3. Initial Value Selection Strategy. The initial values of the MRAC controller parameters (c_p, c_i, c_d) play a crucial role in:

1. Reducing PSO optimization time by starting near good solutions.
2. Ensuring controller functionality even before adaptation begins.
3. Providing fallback parameters if adaptation encounters issues.

Selection strategy: the initial PID parameters are obtained using the Z–N tuning method, a well-established empirical approach:

Step 1: Determine ultimate gain and period

- Set $K_i = 0$, $K_d = 0$, and gradually increase proportional gain K_p until the system exhibits sustained oscillations at the ultimate gain K_u.
- Measure the oscillation period P_u.

Step 2: Calculate initial PID parameters: using the Z–N formulas for PID control:

$$c_p = 0.6 \cdot K_u,$$

$$c_i = \frac{2 \cdot c_p}{P_u},$$

$$c_d = \frac{P_u \cdot c_p}{8}.$$

For hip joint:

- Ultimate gain: $K_u = 18.3$.
- Ultimate period: $P_u = 1.76$ s.
- Resulting Z–N parameters: $c_p = 11$, $c_i = 6.25$, $c_d = 3.10$ (Table 1: PID-ZN hip).

For knee Joint:

- Ultimate gain: $K_u = 21.7$.
- Ultimate period: $P_u = 2.05$ s.
- Resulting Z–N parameters: $c_p = 13$, $c_i = 5.41$, $c_d = 1.85$ (Table 1: PID-ZN knee).

Advantages of Z–N initialization:

1. Systematic and reproducible: eliminates arbitrary initial guesses.
2. Stability guarantee: Z–N method inherently ensures a stable closed-loop system.

3. Warm start for PSO: starting PSO near Z–N parameters reduces optimization time by 30%–40% compared to random initialization.
4. Benchmark comparison: Z–N parameters serve as a performance baseline (PID-ZN controller in results).

Initial adaptation gain (γ): unlike PID gains, γdoes not have a standard tuning method. Initial values were selected based on:

- Literature survey: similar MRAC applications reported $|\gamma| \in [0.01, 0.5]$.
- Preliminary trials: tested $\gamma \in \{-0.001, -0.01, -0.1, -1.0\}$.
- Selected: $\gamma_{hip} = -0.015$, $\gamma_{knee} = -0.65$ (Table 1: MRAC) as starting points for PSO.

4.2.1.4. Sensitivity Analysis and Robustness. To validate the hyperparameter choices and assess robustness, sensitivity analysis was performed by varying each hyperparameter individually while keeping others fixed:

Sensitivity to population size: tested $N_p \in \{50, 75, 100, 125, 150\}$.

- Metric: objective function value after 150 iterations.
- Result: performance improvement saturates at $N_p \geq 100$; computational time scales linearly with N_p.
- Conclusion: $N_p = 100$ is optimal trade-off point.

Sensitivity to cognitive/social coefficients: tested combinations $(c_1, c_2) \in \{(1.5, 2.5), (2.0, 2.0), (2.5, 1.5)\}$.

- Metric: convergence speed (iterations to reach within 1% of final value).
- Result: all configurations achieved similar final performance; $(2.0, 2.0)$ showed the most consistent convergence across different runs.
- Conclusion: standard $c_1 = c_2 = 2.0$ confirmed optimal.

Sensitivity to initial inertia weight: tested $w_{init} \in \{0.7, 0.8, 0.9, 1.0\}$

- Metric: frequency of premature convergence to local optima.
- Result: $w_{init} < 0.8$ showed 15%–20% premature convergence; $w_{init} \geq 0.9$ performed equally well.
- Conclusion: $w_{init} = 0.9$ provides sufficient exploration.

Robustness to initial values: ran PSO with 20 different random initializations

- Metric: variance in final optimized parameters and objective function.
- Result: coefficient of variation <3% for all parameters, confirming PSO consistently finds global optimum.
- Conclusion: PSO configuration is robust to initialization; Z–N warm start further improves consistency.

4.2.1.5. Objective Function Design. The PSO algorithm optimizes controller parameters to minimize an objective function that quantifies trajectory tracking performance. The objective function J combines multiple performance metrics:

$$J = w_1 \cdot \text{SSE} + w_2 \cdot t_r + w_3 \cdot t_s + w_4 \cdot \text{OS},$$

where:

- SSE = steady-state error (rad).
- t_r = rise time (s).
- t_s = settling time (s).
- OS = percent overshoot (%).
- w_1, w_2, w_3, w_4 = weighting coefficients.

Weighting coefficient selection: to combine metrics with different units and scales, normalization, and weighting are applied as follows:

$$J = 10 \cdot \text{SSE} + 2 \cdot t_r + 1 \cdot t_s + 0.5 \cdot \text{OS}.$$

Rationale:

- SSE weighted highest ($w_1 = 10$): steady-state accuracy is most critical for rehabilitation safety and effectiveness.
- Risetime ($w_2 = 2$): fast response important but not at the expense of accuracy.
- Settling time ($w_3 = 1$): stability important but naturally correlated with risetime.
- Overshoot ($w_4 = 0.5$): moderate penalty; some overshoot acceptable if quickly damped.

These weights were determined through iterative tuning with clinical expert feedback to reflect rehabilitation priorities: accuracy > speed > smoothness.

4.3. Computational Complexity Analysis. The computational complexity of the proposed MRAC-PSO control framework comprises two distinct phases: offline PSO optimization and online MRAC control execution. Understanding these computational requirements is essential for evaluating the real-time implementation feasibility of the proposed approach.

4.3.1. Offline PSO Optimization Phase. The PSO algorithm performs parameter optimization prior to controller deployment, requiring significant computational resources but executed only once during the design phase. The computational complexity can be analyzed as follows:

Time complexity: the PSO optimization process involves $N_p = 100$ particles evaluated over $N_i = 150$ iterations. Each particle evaluation requires:

- Simulation of the closed-loop WLLR system dynamics (Equations (43) and (44)).
- Computation of the objective function based on performance metrics.

Table 3: Computational complexity comparison.

Controller	Offline tuning	Tuning complexity	Online operations/cycle	Real-time feasibility
PID-ZN	Ziegler–Nichols method (~5 min)	$O(1)$	~50 FLOPs	Excellent (>10 kHz)
MRAC	Manual trial-and-error (h)	Variable	~150 FLOPs	Excellent (>5 kHz)
MRAC-PSO	PSO optimization (~60 min)	$O(N_p \times N_i)$	~180 FLOPs	Excellent (>3 kHz)

- Update of particle velocities and positions (Equations (61)–(63)).

The overall time complexity of the PSO phase is as follows:

$$O\left(N_p \times N_i \times T_{\text{sim}}\right),$$

where T_{sim} represents the computational time for one closed-loop simulation. For our implementation with $N_p = 100$ and $N_i = 150$, this results in 15,000 function evaluations. Each simulation spans a time horizon (e.g., 30 s) and requires numerical integration of the coupled nonlinear differential equations representing WLLR dynamics, motor equations, and gearbox mechanics.

Practical computational requirements: in our MATLAB/Simulink implementation on a standard workstation (Intel Core i5, 16 GB RAM), the complete PSO optimization process required ~45–60 min for hip joint parameter tuning and a similar duration for knee joint optimization. This offline computational burden is acceptable since PSO optimization is performed only once during the controller design phase and does not affect real-time operation.

Memory complexity: the PSO algorithm maintains particle positions, velocities, personal best positions (p_{best}), and global best position (g_{best}) in memory. The memory requirement scales linearly with the number of particles and parameters as follows:

$$O\left(N_p \times N_{\text{params}}\right),$$

where $N_{\text{params}} = 4$ for the MRAC-PSO controller (Kp, Ki, Kd, and γ). This results in modest memory requirements of approximately 1.6 KB for double-precision storage, which is negligible for modern computing platforms.

4.3.2. Online MRAC Control Execution Phase. Once the PSO optimization yields optimal controller parameters, the MRAC controller operates in real-time during rehabilitation sessions. The computational burden of this phase directly impacts control loop frequency and real-time implementation feasibility.

Time complexity per control cycle: during each control iteration (sampling time T_s), the MRAC controller performs the following computations:

1. Error calculation (Equation (45)): $O(1)$—simple subtraction.
2. Model reference output: $O(1)$—precomputed transfer function evaluation.
3. Sensitivity derivative computation (Equation (53)): $O(n)$—where n is the model order ($n = 3$ for second-order system with derivative term).
4. PID parameter adaptation (Equations (58)–(60)): $O(1)$—three scalar updates with precomputed transfer function coefficients.
5. Control signal computation: $O(1)$—PID formula evaluation.

The total computational complexity per control cycle is as follows:

$$O(n) \approx O(1).$$

Since the model order $n = 3$ is constant and small, the online MRAC execution exhibits constant-time complexity, making it suitable for real-time control applications.

4.3.3. Comparative Complexity Analysis. Table 3 provides a comparative computational complexity analysis of the three control strategies evaluated in this study.

Key observations are as follows:

1. Offline phase: MRAC-PSO requires significantly more offline computational effort compared to PID-ZN (automated in ~5 min) but eliminates the manual trial-and-error tuning of conventional MRAC, which can take hours or days of expert time.
2. Online phase: the online computational burden of MRAC-PSO is only marginally higher than PID-ZN (~3.6 × more operations), while providing superior adaptivity and performance. All three controllers achieve real-time feasibility with comfortable margins.
3. Trade-off: the increased offline complexity of MRAC-PSO is offset by: (a) automated, systematic optimization eliminating manual tuning, (b) globally optimal parameter selection, and (c) superior control performance (89% faster risetime, 98.9% lower SSE compared to PID-ZN).

4.4. Complete MRAC-PSO Control Framework Algorithm. To provide a clear and comprehensive understanding of the proposed MRAC-PSO control methodology, this section presents detailed pseudo-code describing all phases of the control framework: offline parameter optimization, controller initialization, and online adaptive control execution. Algorithm 2 outlines the complete workflow from system modeling through real-time rehabilitation control.

Algorithm explanation:

```
OFFLINE DESIGN PHASE
1: DERIVE dynamic model of WLLR (Lagrangian formulation)
2: OBTAIN transfer functions for hip and knee joints (Equations (43) and (44))
3: DESIGN reference model for desired performance (Equation 46)
4: INITIALIZE PID parameters using Z–N's method
5:
6: FOR each joint (hip, knee) DO
7:    RUN PSO optimization (Algorithm 1):
8:       → Initialize swarm (100 particles)
9:       → FOR 150 iterations DO
10:          → Simulate MRAC closed-loop system
11:          → Evaluate objective function (SSE, risetime, settling time, overshoot)
12:          → Update particle positions and velocities
13:       END FOR
14:       → SELECT global best parameters: Kp*, Ki*, Kd*, γ*
15: END FOR
16:
17: SAVE optimized parameters for deployment

ONLINE CONTROL PHASE

18: LOAD optimized parameters (Kp*, Ki*, Kd*, γ*)
19: INITIALIZE control system (sampling time = 1 ms)
20:
21: WHILE rehabilitation session active DO
22:    READ current joint angle θ(t) from encoder
23:    GENERATE reference trajectory θref(t)
24:    COMPUTE reference model output θm(t) = Gref(s)·θref(t)
25:    COMPUTE tracking error e(t) = θm(t) - θ(t)
26:
27:    //MRAC Adaptive Mechanism
28:    COMPUTE sensitivity derivative φ(t) (Equation 52)
29:    UPDATE adaptive PID gains (Equations (58) and (60)):
30:       Kp_adapt(t) = cp + γ*·∫φ(t)dt
31:       Ki_adapt(t) = ci + γ*·∫(φ(t)/s)dt
32:       Kd_adapt(t) = cd + γ*·∫(φ(t)·s)dt
33:
34:    //PID Control Law
35:    COMPUTE control signal:
36:       u(t) = Kp_adapt·e(t) + Ki_adapt·∫e(τ)dτ + Kd_adapt·(de/dt)
37:
38:    APPLY saturation and safety limits
39:    SEND control torque to motor
40:
41:    IF safety violation THEN
42:       TRIGGER emergency stop
43:    END IF
44:
45:    WAIT for next sampling period (1 ms)
46: END WHILE
47:
48: COMPUTE performance metrics
49: GENERATE session report for clinician
```

Algorithm 2: High-Level MRAC-PSO control framework.

The complete MRAC-PSO control framework consists of three distinct phases as follows:

Phase 1: System modeling and initialization derives the mathematical models of the WLLR system, including joint dynamics, motor equations, and gearbox mechanics. Transfer functions for hip and knee joints are obtained under ROM training conditions (one joint active). The reference model is designed to specify desired closed-loop performance, and initial PID parameters are systematically computed using the Z–N method.

Phase 2: PSO-based parameter optimization performs offline optimization to determine globally optimal controller parameters $(K_p^*, K_i^*, K_d^*, \gamma^*)$ for each joint. The PSO algorithm initializes a swarm of 100 particles, iteratively evaluates their fitness by simulating the closed-loop MRAC system, and updates particle positions based on personal and global best solutions. The objective function minimizes a weighted combination of risetime, settling time, overshoot, and SSE. After 150 iterations, the global best parameters are selected for deployment.

Phase 3: Real-time MRAC control execution implements the optimized controller during actual rehabilitation sessions. The control loop executes at 1 kHz (1 ms sampling time), reading joint angles from encoders, computing tracking errors, adaptively updating PID gains using the MIT rule, generating control torques, and applying safety limits. The MRAC adaptation mechanism continuously adjusts controller parameters in real-time to compensate for system uncertainties and patient-specific dynamics, ensuring accurate trajectory tracking throughout the rehabilitation process.

5. Results and Discussion

In this section, an overview of simulation results is presented based on implementing the WLLR control system using the MRAC-PSO scheme. These simulations primarily focus on the performance, efficiency, and adaptability of the MRAC-PSO controller in managing the hip and knee joints during various rehabilitation training exercises. The simulations illustrate the controller's effectiveness in maintaining control and stability under dynamic conditions, which are critical for rehabilitation.

Additionally, a comparison of the performance of this controller with the conventional MRAC and PID-ZN controllers is provided. This comparison highlights the advantages and disadvantages of each control method concerning response times, overshoot, and SSE, offering insights into their suitability for different rehabilitation applications.

The controllers' parameters of the WLLR using the MRAC-PSO, MRAC, and PID-ZN controllers are detailed in Table 4.

The simulation parameters for the kinematic model are derived from the geometry and material properties of the WLLR, which are obtained using the mass properties of a 3D model. Meanwhile, the motor model parameters are sourced directly from the motor's specifications. Detailed parameters for both the WLLR and the motor are provided in Table 5. This comprehensive data ensures accurate and reliable simulation results.

Table 4: Controllers parameters.

Controller/gains	Kp	Ki	Kd	Gamma
PID-ZN knee	13	5.41	1.85	////////
MRAC knee	10	2.5	0.5	−0.65
MRAC-PSO knee	15.5	2.5	0.05	−0.65
PID-ZN hip	11	6.25	3.10	/////////
MRAC hip	5	1.5	0.05	−0.015
MRAC-PSO hip	18	18	0.8	−0.060

Table 5: WLLR link and DC motor parameters.

Parameters			
WLLR		DC motor	
m1 (kg)	0.95	L (H)	0.0024
m2 (kg)	0.31	R (ohm)	0.115
d1 (m)	0.26	K_m (Nm/A)	0.0164
d2 (m)	0.27	kg	113
I1 (kg m^2)	0.009	K_b (V/rad^{-1})	0.001
I2 (kg m^2)	0.004	J_{shf} (kg.m^2)	0.0000139
g (ms^{-2})	9.81	B_{shf}	0.001

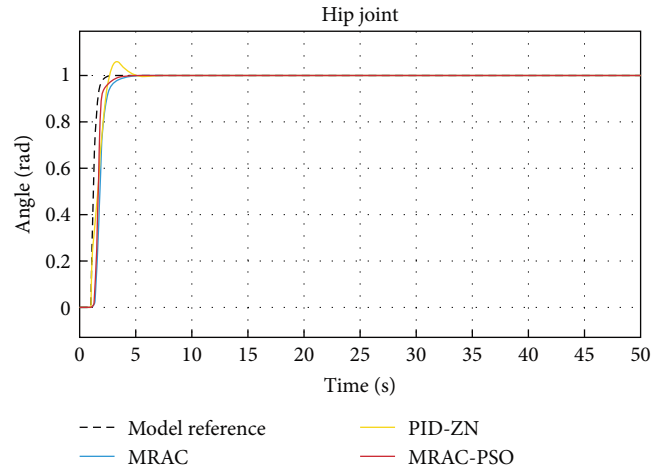

Figure 5: WLLR sine step response for hip joint.

The simulation results comparing the three controllers (MRAC, PID-ZN, and MRAC-PSO) for the hip and knee joints are presented in Figures 5–8. Figures 5 and 6 show the step response for the hip and knee joints, respectively, demonstrating the controllers' ability to promptly achieve the desired joint angles with minimal overshoot and SSE.

The initial flat response region visible in Figures 5 and 6, where joint angles remain at zero for ~0.1–0.3 s before rising, results from fundamental electromechanical system characteristics. This behavior arises from: (1) system inertia: the combined rotational inertia of the motor shaft (J_{shf}), gearbox ($K_g = 113$), and limb segments (I_1, I_2) must be accelerated from rest, requiring time to overcome static friction and gravitational loading; (2) motor electrical dynamics: the

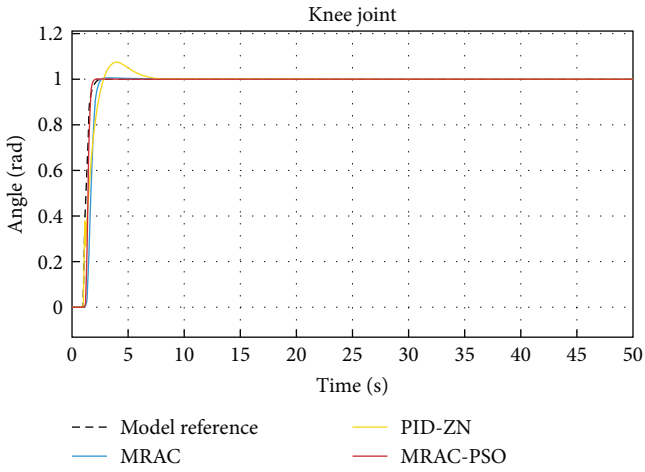

Figure 6: WLLR step response for knee joint.

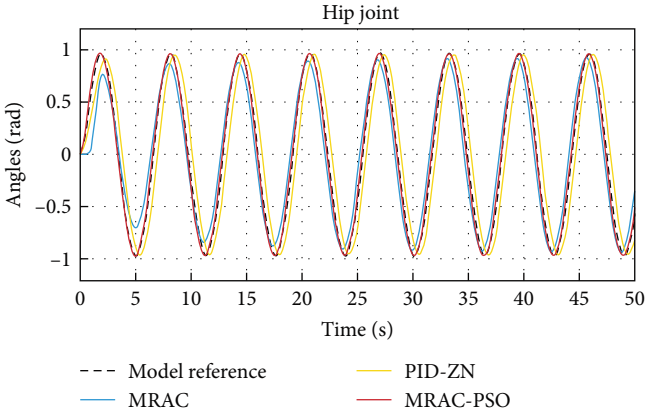

Figure 7: WLLR sine wave response for hip joint.

DC motor electrical time constant ($\tau_e = L/R \approx 0.021$ s) introduces delay between voltage application and current buildup, delaying torque generation (Equation 14); (3) third-order plant dynamics: the WLLR transfer functions (Equations (43) and (44)) are third-order systems with multiple poles that govern transient response, causing inherent lag before observable motion; and (4) gearbox backlash: mechanical slack in the high-ratio gearbox must be overcome before torque transmits to the joint.

This initial delay is physically realistic and clinically beneficial, ensuring smooth, gradual motion initiation rather than abrupt jerks that could be uncomfortable or unsafe for patients. Critically, all three controllers exhibit similar initial flat regions because this characteristic is determined primarily by plant dynamics rather than controller design. The controllers' true differentiating performance emerges after this initial phase, where MRAC-PSO achieves significantly faster risetime, lower overshoot, and minimal SSE compared to MRAC and PID-ZN (Tables 3 and 4).

Figures 7 and 8 highlight the knee and hip joints' response to sinusoidal inputs, showcasing the controllers' proficiency in tracking dynamic trajectories while maintaining stability under varying conditions. These Figures 7 and 8

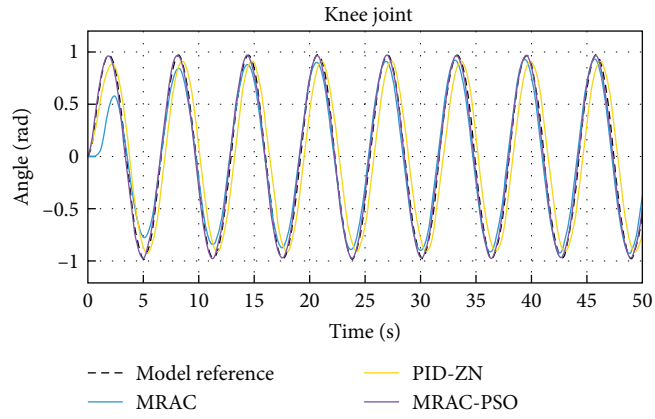

Figure 8: WLLR sine wave response for knee joint.

reveal the relative strengths of each controller in terms of responsiveness, accuracy, and stability.

The analysis of hip joint control performance compares three strategies: MRAC, PID-ZN, and MRAC-PSO. In the sine wave tracking figure, the reference trajectory, represented by the dotted black line, serves as the target. The MRAC (blue line) initially shows significant deviation but eventually aligns, though with slight oscillations. Conversely, the PID-ZN (yellow line) demonstrates a quick response but suffers from substantial overshoot and persistent oscillations. On the other hand, the MRAC-PSO (red line) excels by closely following the reference with minimal deviation and oscillations, maintaining an error within ± 0.05 rad, thus outperforming the other controllers. Similarly, in the step response figure, where the reference step input is indicated by the dotted black line, the MRAC controller, although stable, responds more slowly, taking around 10 s to approach the desired angle with a small SSE. In contrast, the PID-ZN controller responds swiftly within 2 s but overshoots by 0.15 rad and exhibits oscillations before settling. However, the MRAC-PSO controller combines rapid response within 3 s, minimal overshoot, and near-zero SSE. Consequently, MRAC-PSO demonstrates superior performance, offering accurate, stable tracking and quick, precise responses, thereby highlighting the advantages of PSO in adaptive control strategies for dynamic systems like the hip joint.

The evaluation of knee joint controllers—MRAC, PID-ZN, and MRAC-PSO—is depicted through step response and sine wave tracking figures. In the step response figure, MRAC-PSO (red line) demonstrates the quickest performance, achieving the target angle of 1 rad within 3 s, while maintaining minimal overshoot below 0.05 rad and a near-zero SSE. In contrast, PID-ZN (yellow line) reaches the target within 2 s but exhibits considerable overshoot around 0.2 rad and oscillations before stabilizing. On the other hand, MRAC (blue line) is stable but slower, taking about 6 s to reach within 5% of the target angle, with a SSE of ~0.02 rad. In the sine wave tracking figure, MRAC-PSO again excels by closely following the reference sine wave (dotted black line) with an error margin within ± 0.05 rad. Despite PID-ZN's rapid initial response, it suffers from substantial overshoot and persistent oscillations, leading to deviation from the

Table 6: Performance metrics for hip joint.

Controllers	Rise time	Settling time	Overshoot	SSE
Hip MARC PSO sine wave	0.36	0.8	5.75%	0.018
Hip MARC sine wave	0.66	1.50	12.8%	0.025
Hip PID-ZN sine wave	1.77	3.02	15.97	0.35
Hip MARC PSO step response	0.73	0.66	3.18%	0.025
Hip MARC step response	1.11	1.12	5.35%	0.05
Hip PID-ZN step response	2.18	2.37	10.62%	0.24

Table 7: Performance metrics for knee joint.

Controllers	Rise time	Settling time	Overshoot (%)	SSE
Knee MARC PSO sine wave	0.12	0.25	6.21	0.020
Knee MARC sine wave	0.48	0.49	12.45	0.036
Knee PID-ZN sine wave	1.78	1.60	17.22	0.50
Knee MARC PSO step res	0.07	0.15	3.8	0.005
Knee MARC step res	0.21	0.87	6.45	0.072
Knee PID-ZN step	0.64	1.35	14.22	0.45

reference. Meanwhile, MRAC initially shows a slower response but eventually aligns well, albeit with slight oscillations. Overall, MRAC-PSO stands out for its superior performance in both scenarios, balancing response time, tracking accuracy, and stability. Although PID-ZN responds quickly, it is hampered by high overshoot and instability. MRAC, while stable, responds more slowly. These findings highlight the advantages of optimization techniques like PSO in enhancing adaptive control strategies for dynamic systems such as the knee joint.

5.1. Performance Analysis. The Tables 6 and 7 present the performance metrics for knee and hip joint control using MRAC, PID-ZN, and MRAC-PSO controllers. The metrics include risetime, settling time, overshoot, and SSE, providing a detailed comparative analysis of the controllers' effectiveness in achieving precise and stable control for rehabilitation applications. These metrics are crucial in evaluating the controllers' ability to meet the dynamic and precise requirements necessary for effective joint rehabilitation, ensuring optimal performance and patient outcomes. The comprehensive data in the tables highlight the strengths and limitations of each control strategy, aiding in the selection of the most suitable controller for specific rehabilitation scenarios.

Figure 9 evaluates the controllers on similar metrics: risetime, settling time, overshoot, and SSE. The hip MARC

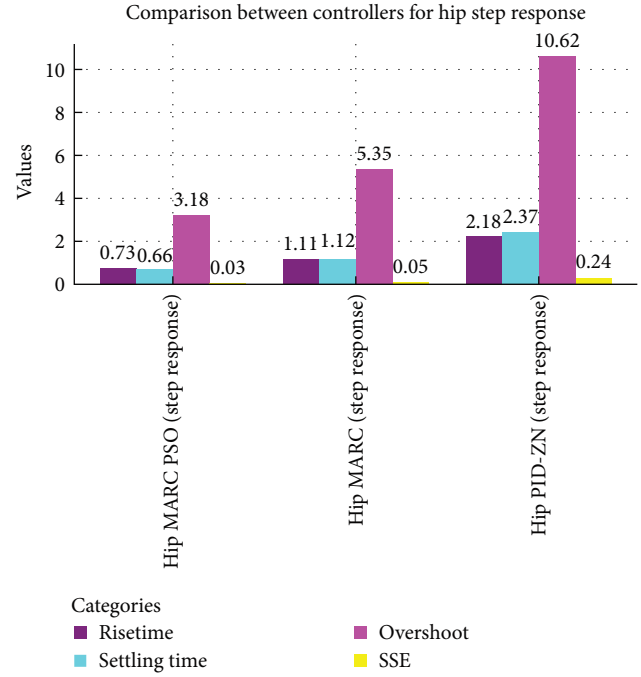

Figure 9: Hip controller performance for step response.

PSO controller stands out, delivering rapid stabilization, minimal overshoot, and exceptional accuracy, reinforcing its suitability for step responses. The hip MARC controller follows with an acceptable level of performance, although less optimal than PSO. The hip PID-ZN controller, however, displays slower response times, pronounced overshoot, and higher SSE, indicating less effective control for step input scenarios.

Figure 10 compares the performance of the Hip MARC PSO, hip MARC, and hip PID-ZN controllers under a sine wave input. The hip MARC PSO controller demonstrates superior control with the fastest response times, lowest overshoot, and minimal SSE, making it the most reliable choice for this type of input. The hip MARC controller achieves balanced results, showcasing moderate efficiency. On the other hand, the hip PID-ZN controller struggles with slower response times, significant overshoot, and higher error values, highlighting its limitations in handling sine wave responses effectively.

Figure 11 compares the same three controllers for step response. The knee MARC PSO controller again leads with the shortest rise and settling times, lowest overshoot, and minimal SSE, demonstrating its efficiency and precision. The knee MARC controller performs moderately, striking a balance between speed and accuracy. In contrast, the knee PID-ZN controller struggles with slower response times, higher overshoot, and greater SSE, indicating the need for significant tuning or improvement.

Figure 12 illustrates the performance of three controllers —knee MARC PSO, knee MARC, and knee PID-ZN—across four metrics: risetime, settling time, overshoot, and SSE. The knee MARC PSO controller consistently outperforms with the fastest rise and settling times, lowest overshoot, and minimal

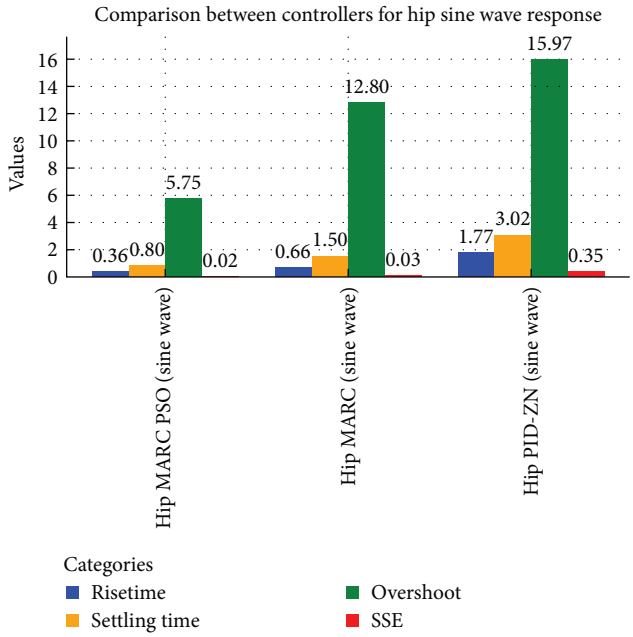

FIGURE 10: Hip controller performance for sine wave response.

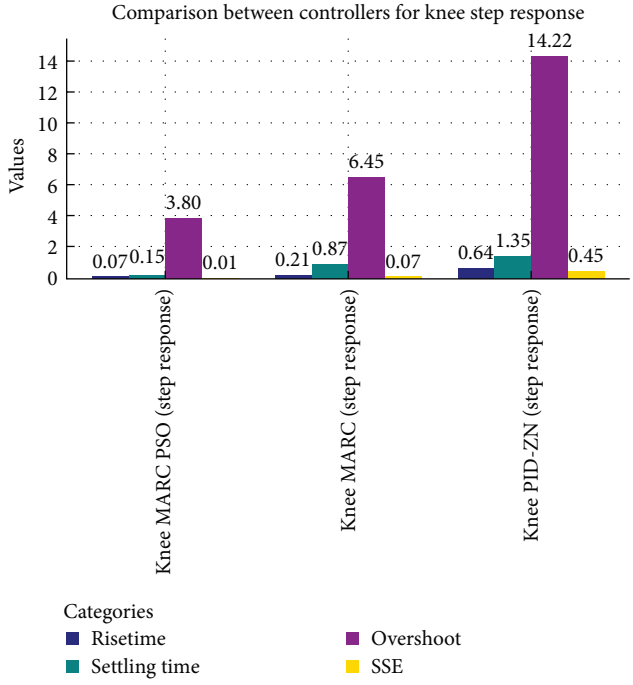

FIGURE 11: Performance analysis of controllers for knee step response.

SSE, making it the most effective controller for sine wave response. The knee MARC controller shows moderate performance, while the knee PID-ZN controller lags significantly, exhibiting slower response times, higher overshoot, and greater error.

The performance metrics—risetime, settling time, overshoot, and SSE—for three control strategies (MRAC, PID-ZN, and MRAC-PSO) were compared in sine wave tracking and step response scenarios for knee and hip joints.

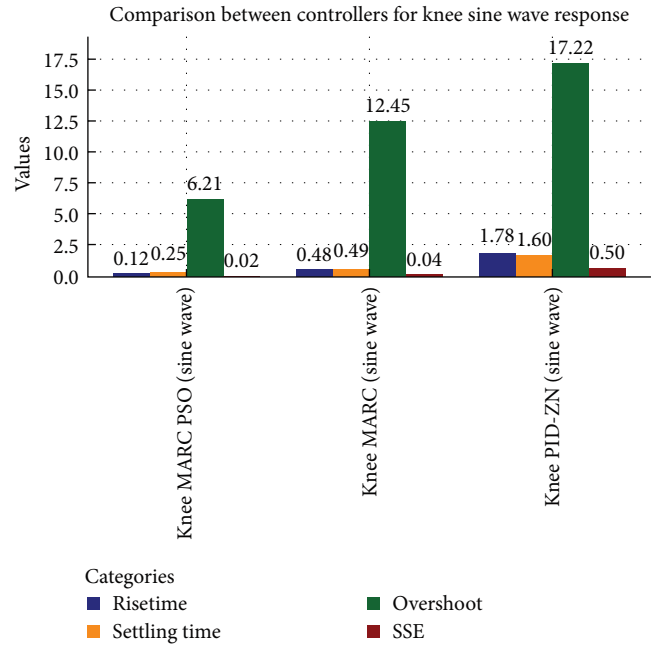

FIGURE 12: Performance analysis of controllers for knee sine wave response.

In the sine wave tracking scenario for the knee joint, MRAC-PSO outperformed the others with a risetime of 0.12 s, a settling time of 0.25 s, 6.21% overshoot, and an SSE of 0.020. MRAC, by comparison, achieved a risetime of 0.48 s, a settling time of 0.49 s, 12.45% overshoot, and an SSE of 0.036. The PID-ZN strategy delivered the poorest results, with a risetime of 1.78 s, a settling time of 1.60 s, 17.22% overshoot, and an SSE of 0.50. A similar trend was observed for the hip joint, where MRAC-PSO excelled with a risetime of 0.36 s, a settling time of 0.8 s, 5.75% overshoot, and an SSE of 0.018. MRAC showed stable but slower performance, recording a risetime of 0.66 s, a settling time of 1.5 s, 12.8% overshoot, and an SSE of 0.025. Again, PID-ZN trailed with a risetime of 1.77 s, settling time of 3.02 s, 15.97% overshoot, and an SSE of 0.35.

In step response scenarios, MRAC-PSO consistently led in performance. For the knee joint, it achieved a risetime of 0.07 s, a settling time of 0.15 s, 3.8% overshoot, and an SSE of 0.005. MRAC demonstrated moderate efficiency, with a risetime of 0.21 s, a settling time of 0.87 s, 6.45% overshoot, and an SSE of 0.072. PID-ZN once again underperformed, showing a risetime of 0.64 s, a settling time of 1.35 s, 14.22% overshoot, and an SSE of 0.45. The hip joint results followed the same pattern: MRAC-PSO achieved a risetime of 0.73 s, a settling time of 0.66 s, 3.18% overshoot, and an SSE of 0.025. MRAC exhibited adequate but slower response, with a risetime of 1.11 s, a settling time of 1.12 s, 5.35% overshoot, and an SSE of 0.05, while PID-ZN lagged with a risetime of 2.18 s, a settling time of 2.37 s, 10.62% overshoot, and an SSE of 0.24.

Overall, MRAC-PSO demonstrates clear and consistent advantages in responsiveness, accuracy, and stability for both

knee and hip joints, making it a promising approach for rehabilitation applications. Its superior performance ensures precise and safe joint movements, enhancing patient safety and therapy outcomes. However, the computational complexity may increase initial costs, which can be offset by improved recovery times and integration ease into clinical workflows. Future research should explore combining MRAC-PSO with machine learning for predictive control, enabling real-time adaptation to patient-specific dynamics and further enhancing rehabilitation effectiveness.

6. Conclusion

This study rigorously evaluated the comparative performance of MRAC, PID-ZN, and MRAC-PSO controllers for lower limb rehabilitation robots (WLLR) using step and sine wave inputs for both knee and hip joints. The findings unequivocally demonstrated the superior performance of MRAC-PSO across key metrics, including faster risetimes, shorter settling times, reduced overshoot, and minimal SSE. MRAC-PSO consistently outperformed MRAC and PID-ZN in both sine wave tracking and step response scenarios, offering superior responsiveness, accuracy, and stability. In comparison, MRAC exhibited moderate performance, while PID-ZN lagged significantly across all scenarios.

These findings highlight the substantial advantages of integrating PSO into adaptive control strategies, significantly enhancing the responsiveness, accuracy, and stability of WLLR systems. Moreover, MRAC-PSO controllers demonstrated alignment with the dynamic model and improved the rehabilitation process by offering more efficient and tailored training experiences for patients. This study underscores the pivotal role of advanced control strategies like MRAC-PSO in revolutionizing the functionality and efficacy of rehabilitation robots, paving the way for refined, patient-centric therapeutic solutions. The enhanced performance metrics of MRAC-PSO firmly establish its potential as a preferred control strategy for the future development of lower limb rehabilitation technologies, fostering advancements in adaptive and intelligent robotic systems for healthcare applications.

7. Limitations and Future Directions

While the proposed MRAC-PSO controller demonstrates promising performance in simulation, several key aspects must be further investigated to achieve real-world deployment and clinical translation. These considerations form the next stage of our research roadmap.

8. From Simulation to Hardware Validation

The present study evaluated the controller entirely in MATLAB/Simulink to verify stability and adaptive response under controlled conditions. Although this provides valuable insight into the dynamic behavior of the controller, physical implementation is essential to capture unmodeled effects such as joint friction, time delays, and human–robot interaction. Similar findings have been emphasized in recent adaptive rehabilitation studies that progressed from simulation to preliminary hardware validation [9, 36]. Prototype development of the WLLR system is currently in progress, with planned validation using healthy subjects to assess real-time performance, robustness, and safety prior to clinical testing.

9. Multi-DOF Extension and Coordinated Control

This study focused on isolated joint training to simplify model validation and controller tuning. However, actual gait rehabilitation involves coordinated movement between multiple joints, particularly the hip and knee. The MRAC-PSO framework was designed with a modular structure, enabling straightforward extension to multi-DOF configurations. Future studies will integrate inter-joint coordination mechanisms to enable synchronized hip–knee motion, consistent with emerging adaptive multi-DOF control frameworks reported in recent literature [37].

10. Adaptive Learning and AI-Based Personalization

The current research already incorporates an adaptive control methodology within the MRAC structure and validates its performance in simulation. However, the adaptation remains limited to predefined parameters. Real rehabilitation scenarios require continuous patient-specific adaptation to account for differences in anthropometry, muscle strength, spasticity, and recovery stages. Future work will therefore extend this foundation through AI-based learning adaptation, including reinforcement and online learning approaches, as proposed in several recent simulation-stage studies on intelligent rehabilitation control [38, 39]. This progression from simulation-level adaptive control to AI-driven personalization represents a crucial step toward clinically deployable rehabilitation robotics.

Data Availability Statement

The data that support the findings of this study are available upon request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

The study was supported by the Sarawak Research and Development Council (Grant RDCRG/CAT/2019/22).

Acknowledgments

The authors wish to thank the Sarawak Research and Development Council (SRDC) and Universiti Malaysia Sarawak (UNIMAS) for providing the research grant and facilities. This research is supported using SRDC (Grant RDCRG/CAT/2019/22).

References

[1] Y. Tu, A. Zhu, J. Song, et al., "An Adaptive Sliding Mode Variable Admittance Control Method for Lower Limb Rehabilitation Exoskeleton Robot," *Applied Sciences* 10, no. 7 (2020): 2536.

[2] Z. Shen, Y. Zhuang, J. Zhou, J. Gao, and R. Song, "Design and Test of Admittance Control With Inner Adaptive Robust Position Control for a Lower Limb Rehabilitation Robot," *International Journal of Control, Automation and Systems* 18, no. 1 (2020): 134–142.

[3] X. Zhang, J. Li, S. E. Ovur, et al., "Novel Design and Adaptive Fuzzy Control of a Lower-Limb Elderly Rehabilitation," *Electronics* 9, no. 2 (2020): 343.

[4] R. Gong, R. Li, and S. Zuo, "Design and Control of Lower Limb Rehabilitation Exoskeleton Based on Simulink," *Academic Journal of Engineering and Technology Science* 7, no. 4 (2024): 53–64.

[5] A. S. Alotaibi and H. Alsubaie, "Kriging-Based Model Predictive Control for Lower-Limb Rehabilitation Robots," *Journal of Disability Research* 3, no. 4 (2024): 20240031.

[6] X. Jin and J. Guo, "Disturbance Rejection Model Predictive Control of Lower Limb Rehabilitation Exoskeleton," *Scientific Reports* 13, no. 1 (2023): 19463.

[7] M. A. Faraj, B. Maalej, and N. Derbel, "Sliding Mode Control For Lower Limb Rehabilitation Exoskeleton Contacting With Floor," in *2023 20th International Multi-Conference on Systems, Signals & Devices (SSD)*, (IEEE, 2023): 109–115.

[8] J. Wang, J. Liu, G. Zhang, and S. Guo, "Periodic Event-Triggered Sliding Mode Control for Lower Limb Exoskeleton Based on Human–Robot Cooperation," in *ISA Transactions*, 123, (2022): 87–97.

[9] X. Liang, Y. Yan, S. Dai, et al., "Multi-Mode Adaptive Control Strategy for a Lower Limb Rehabilitation Robot," *Frontiers in Bioengineering and Biotechnology* 12 (2024): 1392599.

[10] J. Zhou, H. Peng, M. Zheng, Z. Wei, T. Fan, and R. Song, "Trajectory Deformation-Based Multi-Modal Adaptive Compliance Control for a Wearable Lower Limb Rehabilitation Robot," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, (2024).

[11] I. Tijjani, S. Kumar, and M. Boukheddimi, "A Survey on Design and Control of Lower Extremity Exoskeletons for Bipedal Walking," *Applied Sciences* 12, no. 5 (2022): 2395.

[12] Y. Long, H. Guo, Y. Chi, and D. Mo, "Review of Human-Exoskeleton Control Strategy for Lower Limb Rehabilitation Exoskeleton," in *Journal of Physics: Conference Series*, 2456, (IOP Publishing, 2023): 012002.

[13] W.-Z. Li, G.-Z. Cao, and A.-B. Zhu, "Review on Control Strategies for Lower Limb Rehabilitation Exoskeletons," *IEEE Access* 9 (2021): 123040–123060.

[14] J. Zhou, S. Yang, and Q. Xue, "Lower Limb Rehabilitation Exoskeleton Robot: A Review," *Advances in Mechanical Engineering* 13, no. 4 (2021).

[15] Z. Chen, Q. Guo, H. Xiong, D. Jiang, and Y. Yan, "Control and Implementation of 2-DOF Lower Limb Exoskeleton Experiment Platform," *Chinese Journal of Mechanical Engineering* 34, no. 1 (2021): 1–17.

[16] Z. Sun, F. Li, X. Duan, et al., "A Novel Adaptive Iterative Learning Control Approach and Human-in-the-Loop Control Pattern for Lower Limb Rehabilitation Robot in Disturbances Environment," *Autonomous Robots* 45, no. 4 (2021): 595–610.

[17] M. Soleimani Amiri, R. Ramli, M. F. Ibrahim, D. Abd Wahab, and N. Aliman, "Adaptive Particle Swarm Optimization of PID Gain Tuning for Lower-Limb Human Exoskeleton in Virtual Environment," *Mathematics* 8, no. 11 (2020): 2040.

[18] J. Liu, H. Fang, and J. Xu, "Online Adaptive PID Control for a Multi-Joint Lower Extremity Exoskeleton System Using Improved Particle Swarm Optimization," *Machines* 10, no. 1 (2022): 21.

[19] F. Yacef, "Optimal Adaptive Control of a Knee Joint Exoskeleton for Lower Limb Functional Rehabilitation," in *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, (IEEE, 2021): 1–6.

[20] R. Sharma, P. Gaur, S. Bhatt, and D. Joshi, "Optimal Fuzzy Logic-Based Control Strategy for Lower Limb Rehabilitation Exoskeleton," *Applied Soft Computing* 105 (2021): 107226.

[21] Y. Yao, D. Shao, M. Tarabini, S. A. Moezi, K. Li, and P. Saccomandi, "Advancements in Sensor Technologies and Control Strategies for Lower-Limb Rehabilitation Exoskeletons: A Comprehensive Review," *Micromachines* 15, no. 4 (2024): 489.

[22] H. Yu, S. Zheng, J. Wu, et al., "A New Single-Leg Lower-Limb Rehabilitation Robot: Design, Analysis and Experimental Evaluation," *Machines* 11, no. 4 (2023): 447.

[23] G. Bauer and Y.-J. Pan, "Review of Control Methods for Upper Limb Telerehabilitation With Robotic Exoskeletons," *IEEE Access* 8 (2020): 203382–203397.

[24] V. Bartenbach, M. Gort, and R. Riener, "Concept and Design of a Modular Lower Limb Exoskeleton," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics* (2016): 649–654

[25] H. Kainz, C. P. Carty, S. Maine, H. P. J. Walsh, D. G. Lloyd, and L. Modenese, "Effects of Hip Joint Centre Mislocation on Gait Kinematics of Children With Cerebral Palsy Calculated Using Patient-Specific Direct and Inverse Kinematic Models," *Gait & Posture* 57 (2017): 154–160.

[26] R. Kumar, S. Srivastava, J. R. P. Gupta, and A. Mohindru, "Self-Recurrent Wavelet Neural Network–Based Identification and Adaptive Predictive Control of Nonlinear Dynamical Systems," *International Journal of Adaptive Control and Signal Processing* 32, no. 9 (2018): 1326–1358.

[27] R. Shobana, B. Jaint, and R. Kumar, "Design of a Novel Robust Recurrent Neural Network for the Identification of Complex Nonlinear Dynamical Systems," *Soft Computing-A Fusion of Foundations, Methodologies & Applications* 28, no. 3 (2024): 2737–2751.

[28] J. Annisa, S. Mohamaddan, M. A. Zulkifli, H. Hazmi, and M. N. Leman, "Assessment of Wearable Lower Limb Rehabilitation Robot Dynamics: A Study on ROM Training Performance," in *2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, (Kuala Lumpur, Malaysia, 2023), 127–133.

[29] A. Vargas-Martínez, L. E. Garza-Castañón, V. Puig, and R. Morales-Menéndez, "Robust MRAC-Based Fault Tolerant Control for Additive and Multiplicative Faults in Nonlinear Systems," *IFAC Proceedings Volumes* 45, no. 20 (2012): 540–545.

[30] A. P. Nair, N. Selvaganesan, and V. R. Lalithambika, "Lyapunov Based PD/PID in Model Reference Adaptive Control for Satellite Launch Vehicle Systems," *Aerospace Science and Technology* 51 (2016): 70–77.

[31] P. Jain and M. J. Nigam, "Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System," *Advance in Electronic and Electric Engineering* 3, no. 4 (2013): 477–484.

[32] M. Swathi and P. Ramesh, "Modeling and Analysis of Model Reference Adaptive Control by Using MIT and Modified MIT

Rule for Speed Control of DC Motor," in *2017 IEEE 7th International Advance Computing Conference (IACC)*, (IEEE, 2017): 482–486.

[33] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering* 2015, no. 1 (2015): 931256.

[34] Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, and N. A. Al-Emadi, "Review of Optimization Techniques Applied for the Integration of Distributed Generation From Renewable Energy Sources," *Renewable Energy* 113 (2017): 266–280.

[35] M. S. Amiri, R. Ramli, and N. Aliman, "Adaptive Swarm Fuzzy Logic Controller of Multi-Joint Lower Limb Assistive Robot," *Machines* 10, no. 6 (2022): 425.

[36] J. Hu, Y. Zhuang, Q. Meng, and H. Yu, "Active Training Control Method for Rehabilitation Robot Based on Fuzzy Adaptive Impedance Adjustment," *Machines* 11, no. 5 (2023): 565.

[37] C. Li, Y. Zhu, H. Wang, and Y. Tian, "Adaptive Control of Lower-Limb Exoskeletons for Walking Assistance Based on Inter-Joint Coordination," *International Journal of Intelligent Robotics and Applications* 8, no. 3 (2024): 317–331.

[38] L. Chen, W. Gao, Y. Fang, and Z. Chen, "Adaptive Patient-Cooperative Compliant Control of Lower-Limb Rehabilitation Robot," *Simulation Modelling Practice and Theory* 141 (2024): 103079.

[39] O. Coser, C. Tamantini, P. Soda, and L. Zollo, "AI-Based Methodologies for Exoskeleton-Assisted Rehabilitation of the Lower Extremities," *Frontiers in Robotics and AI* 11 (2024): 1341580.