robotics

MDPI

Article

Upper-Limb Robotic Rehabilitation: Online Sliding Mode Controller Gain Tuning Using Particle Swarm Optimization

Deira Sosa Méndez [1,*], David Bedolla-Martínez [2], Maarouf Saad [2], Yassine Kali [3], Cecilia E. García Cena [4] and Ángel L. Álvarez [1]

[1] Escuela de Ingeniería de Fuenlabrada, Universidad Rey Juan Carlos, 28942 Madrid, Spain
[2] Electrical Engineering, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[3] DIFIA Laboratory, School of Engineering, Université du Québec en Abitibi-Témiscamingue, Rouyn-Noranda, QC J9X 5E4, Canada
[4] Escuela Técnica Superior de Ingeniería y Diseño Industrial, Center for Automation and Robotics, UPM-CSIC, Universidad Politécnica de Madrid, 28012 Madrid, Spain
* Correspondence: deira.sosa.mendez@urjc.es or sosam.deira@gmail.com

Abstract: Two primary challenges in controlling robotic rehabilitation devices are the uncertainties in dynamic models and, more importantly, the need for controllers capable of adapting to external disturbances due to human–robot interaction. To address these issues, this paper proposes the particle swarm optimization (PSO) algorithm for the real-time gain tuning in the sliding mode controller (SMC) based on the exponential reaching law (ERL). The proposed approach was designed for a seven-degrees-of-freedom (DOF) robotic exoskeleton used in upper-limb physical rehabilitation. The optimization algorithm aims to minimize tracking errors in rehabilitation exercises through the robust ERL controller applied to nonlinear systems with external disturbances. The proposed method was validated through experimental tests conducted on two healthy subjects, and the outcomes indicated a reduction of over 20% in tracking errors compared to heuristically tuned gains. Mathematical analyses of dynamic modeling and algorithm convergence are shown.

Keywords: exoskeleton; exponential reaching law; particle swarm optimization algorithm; sliding mode

Academic Editor: Kean C. Aw

Received: 12 March 2025
Revised: 13 April 2025
Accepted: 14 April 2025
Published: 17 April 2025

Citation: Sosa Méndez, D.; Bedolla-Martínez, D.; Saad, M.; Kali, Y.; García Cena, C.E.; Álvarez, Á.L. Upper-Limb Robotic Rehabilitation: Online Sliding Mode Controller Gain Tuning Using Particle Swarm Optimization. *Robotics* **2025**, *14*, 51. https://doi.org/10.3390/robotics14040051

1. Introduction

According to the World Health Organization in 2019 [1], 2.4 billion people experienced conditions that could benefit from rehabilitation services (1 in 3 impaired people globally). Between 1990 and 2019, there was a 69.4% rise in the population living with disabilities. Musculoskeletal disorders and strokes are among the leading causes of disability worldwide [2]. These conditions often result in limited functional capacity, particularly in terms of mobility.

One of the most common ways to support mobility recovery in individuals is through rehabilitation services, where physical rehabilitation plays a fundamental role. These services are patient-centered and aim to promote independence. Within these treatments, robotic-assisted rehabilitation has proven to be efficient because this method allows for personalized, precise, and repetitive movements, increasing the intensity of the treatment [3,4]. Such devices include exoskeletons, which are robotic systems designed based on bio-inspired models (similar in shape and functions to the human body) that are externally attached to the user [5,6].

The main challenges in robotic rehabilitation with exoskeletons are (a) those that come from their physical characteristics and (b) those that come from their application. The

former addresses engineering design aspects for the physical components, workspace, and robot kinematic/kinetic models. The latter contemplates the nature of rehabilitation treatments, such as precision (personalized treatments), physical assistance by the physiotherapist (resistive or assisted therapy), and coupling between the robot and the patient (generates highly nonlinear systems with changing dynamic models), among others [7,8].

These latter aspects present a significant challenge for developers as they require the implementation of automated control strategies that ensure compliance while also providing security to users. A key aspect in the design of control strategies is the tuning of controller gains. Typically, this tuning is performed heuristically or based on the biomechanical characteristics of a particular patient population. However, this approach presents the following drawbacks: (a) the process is slow and tedious [8], and (b) its application is limited to a small group of users. Therefore, it is essential to develop approaches for automatic controller gain tuning that account for system complexities, model uncertainties, and external disturbances (such as human–exoskeleton interaction). The following section provides a brief literature review on these approaches.

2. Literary Review: Controllers and Gain Tuning Methods

Due to the nature of exoskeleton applications, it is crucial to implement controllers that ensure optimal performance, as these systems interact directly with the user and are exposed to external disturbances. This review focuses on analyzing the most commonly used controllers in exoskeletons and the techniques employed for gain tuning, covering both classical and more recent approaches. Table 1 provides a brief overview of several controllers main characteristics, along with examples of studies where they have been applied.

As shown in Table 1, most controllers combine different approaches to achieve good performance in the application. However, the Proportional–Integral–Derivative (PID) controller stands out as the most widely used due to its simplicity in implementation and tuning, as well as its effectiveness in linear systems. It is followed by sliding mode control (SMC), neural networks (NNs), fuzzy logic (FL), and supertwisting control (ST).

In terms of robustness against uncertainties and disturbances, SMC and ST controllers are particularly noteworthy, with the latter also being more accurate in complex, nonlinear systems. On the other hand, NN- and FL-based approaches are also capable of effectively handling uncertainty and achieve high accuracy, particularly when trained and properly designed. The choice of controller depends on system complexity, the nature of disturbances, and the required level of precision or robustness.

In addition to selecting the type of controller, another crucial factor for its efficiency is the choice of the gain tuning method, as this directly influences the control system stability and performance. For this purpose, various approaches are employed, which differ in complexity and effectiveness. Table 2 presents some of the most commonly used gain tuning methods, ranging from traditional techniques, such as heuristic tuning, to advanced approaches, including optimization algorithms and machine learning.

It should be noted that in most of the reviewed works, the controller gains are presented as constant values over time, without specifying how these values were assigned. In these studies, only the effectiveness of the proposed controller is demonstrated.

Table 1. Controller comparison.

Controller	Advantages	Disadvantages	Works
PID, PD and PI	• Easy to understand and implement. • Effective on linear systems.	• Do not handle nonlinear systems, uncertainties, or large disturbances. • Require manual tuning.	PD: 3-DOF LLE [9] (Sim), 5-DOF ULE S [10] (Exp). PI: 3-DOF ULE Fuzzy-PI [11] (Exp-4 HS). PID: 1-DOF ULE [12] (Exp-6 HS), 1-DOF ULE Fuzzy PID [13] (Exp-5 HS), 2-DOF LLE PID with fuzzy precompensator [14] (Sim with human gait dataset), 2-DOF LLE- robust PID adaptive [15] (Sim and Exp-3 HS), 2-DOF active and passive hybrid hydraulic ankle prosthesis [16] (Sim-Exp), 2-DOF LLE [17] (Sim), 3-DOF LLE [18,19] (Sim), 5-DOF LLE [20] (Sim), 9-DOF LLE online adaptive PID [21] (Sim)
Sliding Mode Control (SMC)	• Robust against uncertainties, external disturbances, and variations in system parameters. • Adjustable nonlinear control for complex dynamic systems.	• It uses high-frequency switching. • It exhibits the chattering phenomenon. • Difficult to adjust due to the high nonlinearity of the control system.	2-DOF ULE [22,23] (Sim), 2-DOF ULE New model free terminal SMC [24] (Sim), 2-DOF ULE Fuzzy SMC [25] (Sim), 2-DOF ULE robust neural adaptive integral SMC [26] (Exp-2 HS), 3-DOF ULE [27] (Sim and Exp), 3-DOF pediatric LLE robust SMC [28] (Sim), 5-DOF ULE adaptive finite-time robust SMC [29] (Sim), 6-DOF ULE adaptive SMC [30] (co-simulation and Exp-1 HS), 7-DOF ULE new adaptive reaching law [31] (Exp- 1 HS), 7-DOF iReHave ULE fractional-order ultra-local model-based neural network SMC [32] (7-DOF co-simulation and 2-DOF Exp).
Supertwisting (ST)	• Robust against disturbances and inaccurate modeling. • It implements a two-stage control, which makes the switching less abrupt.	• It requires a precise parameter adjustment. • It is complex to design and implement for multiple degrees of freedom (DOF).	1-DOF ULE 2nd order SM-ST [33] (Exp), 3-DOF ULE [34] (Sim and Exp-2 HS).
Backstepping	• It is designed to stabilize a special class of nonlinear systems called strict-feedback systems. • Guarantees system stability.	• It is sensitive to variations in the system parameters. • In practice, it requires intensive calculations and it may exhibit the chattering phenomenon.	2-DOF ULE [35,36] (Sim), 2-DOF LLE Backstepping with NN [37] (Sim).
Neural Networks (NNs)	• Adapt well to highly nonlinear systems and capable of managing uncertainties. • Learn and improve their performance over time.	• Require large volumes of data for training. • Their design and calibration are complex.	2-DOF ULE [26] (Exp-2 HS), 2-DOF LLE [17,38] (Sim).
Fuzzy Logic (FL)	• Flexible to address uncertainties, disturbances, or inaccurate systems. • It does not require precise mathematical modeling.	• Difficult to optimize and design for extremely complex systems. • At times, it is less precise than other controllers if not tuned correctly.	Hand [39] (Exp-6 HS), 1-DOF ULE elbow [40–42](Exp).

Table 2. Gain tuning methods comparison.

Method	Advantages	Disadvantages	Works
Tuning by trial and error, manually or pre-tuned.	• Flexible and suitable for a wide variety of systems. • It is commonly used in nonlinear systems or difficult model systems.	• It is slow for high-dimensional systems. • It does not guarantee optimal performance without significant operator experience.	2-DOF ULE: [22] (Sim), [36] (Sim).
Ziegler–Nichols (ZN)	• Widely used for tuning PID controllers. • Easy to apply and useful for the allocation of initial gains.	• Empirical formulas are used to adjust the parameters. • It may not be optimal in terms of robustness and response to disturbances.	PID: 3-DOF LLE [18] (Sim), [19] (Sim).
Variable or adaptive gain method	• The controller parameters are tuned dynamically during operation in response to changes in the nonlinear system or disturbances. • Greater robustness against variations in the system's dynamics.	• The implementation is more complex and requires precise mathematical models or advanced algorithms.	1-DOF ULE applied adaptative PID [12] (Exp-6 HS), 2-DOF LLE applied a robust PID adaptive controller [15] (Sim and Exp-3 HS), 3-DOF LLE applied an optimal PID [19] (Sim), 3-DOF ULE applied fuzzy logic [11] (Exp-4 HS), 5-DOF ULE applied an adaptive finite-time robust SMC [29] (Sim), 9-DOF LLE applied an adaptive PID [21] (Sim).
Neural network-based control (learning-based tuning)	• Controls a nonlinear system by learning from collected data. • Ability to adapt to highly nonlinear systems and manage unmodeled disturbances.	• It requires a large training dataset and is difficult to interpret and manually adjust. • The controller may need to be retrained if the system undergoes significant changes.	2-DOF ULE [26] (EXp-2 HS), 3-DOF LLE [18] (Offline, Sim), 5-DOF LLE [20] (Sim), 7-DOF iReHave ULE [32] (7-DOF Co-simulation and 2-DOF Exp).
Nonlinear model predictive control (NMPC)	• It uses a nonlinear system model to predict its future behavior and optimizes the control signal based on those predictions.	• Computationally intensive (involves solving an optimization problem in real time).	7-DOF ULE robust MPC with an integral sliding mode (Exp- 3 HS) [43].
Heuristic optimization (evolutionary, genetic, or particle swarm algorithms)	• It handles nonlinear systems with a large number of parameters. • Suitable when search space is complex and there are multiple objectives to optimize, such as performance against disturbances and stability. • Explores a broad solution space and automatically tunes the controller parameters.	• It can be slow and require high computational resource consumption. • It finds solutions that are very close to optimal.	PSO: 2-DOF ULE PID [35] (offline and Sim), 2-DOF LLE PID with fuzzy precompensator [14] (Sim with human gait dataset), 2-DOF LLE robust PID adaptive controller [15] (Sim and Exp- 3 HS), 3-DOF LLE PD [9] (Offline and Sim), 3-DOF ULE [34] (Online Sim and Exp-2 HS), 9-DOF LLE online adaptive PID controller [21] (Sim).

According to Table 2, for linear systems, methods such as Ziegler–Nichols (the most common for parameter initialization) and tuning by trial and error (either manually or pre-tuned) tend to be effective and easy to implement. However, in nonlinear systems, which are inherently susceptible to uncertainties and disturbances, the most effective methods include heuristic optimization, robust control, and adaptive control. This is because neural networks require considerable time to process each input and target data sample due to the size of each training set.

Over the years, nature has inspired the development of algorithms to solve complex real-world problems. However, there is no universal applicable heuristic method for optimizing the gains of a nonlinear controller in the presence of disturbances, as the effectiveness of each algorithm depends on the type of nonlinear system, the nature of the disturbances, and the specific requirements of the control problem. Some of the most common heuristic methods used for these types of problems include genetic algorithms (GAs), particle swarm optimization (PSO) [8,9,14–16,19,21,35], simulated annealing (SA), ant colony optimization (ACO), cuckoo search (CS) algorithm, and differential evolution (DE) algorithms.

For optimizing gains in a nonlinear controller in the presence of disturbances, genetic algorithms and PSO are the most widely used and effective due to their flexibility and ability to identify suitable solutions in nonlinear problems with multiple local optima. However, when rapid convergence is required or there are computational limitations, PSO is an excellent choice, as it is an algorithm that is part of swarm intelligence, a significant branch of artificial intelligence [44,45]. Conversely, if robustness against the possibility of getting trapped in local optima is desired and more computational time is available, genetic algorithms or differential evolution are more appropriate options.

In the context of exoskeletons for rehabilitation, where there is physical interaction with users, methods that require precise knowledge of the mathematical model are generally not used, especially if the system is of high order and exhibits nonlinearities or external disturbances [29,46–48]. Therefore, in these applications, it is essential to have a method that adjusts the controller parameters in real time, adapting to the evolution of the system and changes in user characteristics. Designing such solutions are complex and may require specialized hardware and software.

As a result, it is necessary to implement control approaches based on metaheuristic optimization that adapt in real time to different users. These algorithms identify efficient configurations for the controller compared to traditional methods. By adopting these robust approaches, the efficiency of rehabilitation tasks is improved.

Optimization-based approaches overcome the drawbacks of those that do not employ it, so the key steps of this approach are outlined below:

1. Establish an objective function: The aim is to meet specific criteria while considering the associated cost, whether minimum or maximum. This can be achieved through one or more objective functions, or by using a weighted objective function.
2. Optimizer selection: Direct methods, such as the response surface method and gradient descent, as well as swarm-based methods like evolutionary algorithms (EAs), are employed. Direct methods are sensitive to noise, which can reduce their performance. Additionally, they typically provide solutions based on a single criterion, tend to converge to local optima, and may encounter issues related to an unidentified search space. In contrast, swarm-based methods have demonstrated superior performance compared to these traditional approaches [8,44].

Lately, the particle swarm (PS) algorithm has emerged as one of the EAs utilized for optimizing controller parameters in exoskeletons, due to the following characteristics [19,21,49]: (1) its simplicity and high convergence speed; (2) it does not need

any assumptions about the problem to be optimized; (3) it explores extensive solution spaces; (4) it is capable of handling intricate complex nonlinear systems with a high number of dimensions and challenges in dynamic optimization across diverse domains; (5) it does not demand a precise dynamic model or accurate system parameters beforehand; (6) a limited number of parameters require tuning; and (7) it exhibits a high likelihood and efficiency in discovering global optima, among others.

Despite the merits of the PS algorithm, it also has some disadvantages [15,16,19,21]: (1) the time required for convergence might be extended, potentially leading to undesirable chattering phenomena in a control context; (2) the potential problem of early convergence could hinder the ability to adapt to sudden changes in the system and the environment; and (3) there is a risk of early convergence, leading to entrapment in local optima, among others.

In this case, due to the application objectives, the PS algorithm was chosen to optimize the gains of the sliding mode controller. This technique is a robust and well-established control method, designed to handle uncertain dynamics, provide a rapid response to changes, offer robustness against parameter variations, and facilitate implementation [23,27,28,30]. However, its main disadvantage is the occurrence of oscillations. To mitigate this problem and reduce convergence time, an alternative approach is proposed that incorporates an exponential function as the reaching law, which preserves the robustness of SMC [28,31,50,51]. This enables the controller to adapt in real time to changes in the morphology and motor abilities of exoskeleton users.

To the best of our knowledge, the main contributions of this study involve the following:

- Online gain tuning using the exponential reaching law (ERL) and the PSO algorithm on a highly nonlinear system is proposed for the first time.
- This is the first real-time implementation of the robust PSO-ERL algorithm on a 7 DOF exoskeleton for online gain tuning for trajectory tracking.
- It presents the findings from the experimental assessment of gain tuning among healthy subjects exhibiting diverse physiological characteristics.

The work is structured as follows: Section 3 covers the PSO algorithm's mathematical aspects, the exoskeleton's dynamical model (nonlinear considering external disturbances), and the controller (features, stability analysis, etc.). Section 4 details the implementation conditions, experimental results, and analyses performed. The last section concludes and suggests several future study directions.

3. Mathematical Aspects

In this section, we address the main concepts considered for the development of this work.

3.1. Exoskeleton's Uncertain Dynamic Model

During physical therapy treatments, patients wear the exoskeleton for purposes of rehabilitation and/or assistance. Due to this, there are coupled systems with two different dynamics (human and exoskeleton) which must be considered in the real application.

An adaptive controller is required for nonlinear systems that adapt to model uncertainties and external disturbances. So, the dynamics of the ETS-MARSE robot are presented in joint space as follows [43]:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) + F(\dot{\theta}) = \tau_o + \tau_u \tag{1}$$

where $\theta \in \mathbb{R}^{n\times 1}$ contains the joint angular position, $\dot{\theta} \in \mathbb{R}^{n\times 1}$ vector contains the joint angular velocity, $\ddot{\theta} \in \mathbb{R}^{n\times 1}$ vector contains the joint angular acceleration, $M(\theta) \in \mathbb{R}^{n\times n}$ is the inertial matrix, $C(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis terms, $G(\theta) \in \mathbb{R}^{n\times 1}$ is

the gravitational forces, $F(\dot{\theta}) \in \mathbb{R}^{n\times 1}$ characterizes nonlinear Coulomb friction, $\tau_o \in \mathbb{R}^{n\times 1}$ denotes the generalized torque vector, and $\tau_u \in \mathbb{R}^{n\times 1}$ represents the vector of unknown disturbing forces originating from the patient with n equal to 7 (number of joints). Consider the following relations:

$$M = M_o + M_u \tag{2}$$

$$C = C_o + C_u \tag{3}$$

$$G = G_o + G_u \tag{4}$$

where the terms with subscript "o" represent the nominal part and terms with subscript "u" represent the unknown part for M, C, and G.

Isolating $\ddot{\theta}$ from Equation (1), Equation (5) is obtained:

$$\ddot{\theta} = M_o^{-1}(\theta)[\tau_o - C_o(\theta,\dot{\theta})\dot{\theta} - G_o(\theta)] + U_u \tag{5}$$

where U_u includes the unknown variables shown in Equation (6), and $M_o(\theta)$ is a symmetric matrix that is positive definite, ensuring the availability of $M_o^{-1}(\theta)$.

$$U_u = M_o^{-1}(\theta)[\tau_u - M_u(\theta)\ddot{\theta} - C_u(\theta,\dot{\theta})\dot{\theta} - G_u(\theta) - F(\dot{\theta})] \tag{6}$$

A compensatory torque derived from the nominal components of (1) is as follows:

$$\tau_o = M_o(\theta)\nu(t) + C_o(\theta,\dot{\theta})\dot{\theta} + G_o(\theta) \tag{7}$$

where $\nu(t)$ to be designed later and it represents the virtual control that guides the system toward the control goal. Substituting Equation (7) in Equation (5), the following uncertain linear system is obtained.

$$\ddot{\theta} = \nu(t) + U_u \tag{8}$$

3.2. Proposed Controller

The initial step involves selecting the sliding surface S based on the tracking error, as expressed in Equation (9). This indicates that upon reaching the sliding surface, the tracking error decreases to zero. The rate of convergence is linked to the value of Λ_i.

$$S_i = \Lambda_i e_i + \dot{e}_i \tag{9}$$

where the tracking error is calculated as $e_i = \theta_{di} - \theta_i$, where θ_{di} denotes the desired trajectory, and θ_i represents the measured angular position.

The subsequent phase includes choosing the virtual control ν that facilitates the error convergence toward the sliding surface. To achieve this, the following condition must be fulfilled:

$$S_i\dot{S}_i < 0 \qquad\qquad \forall t$$

A controller based on sliding modes with ERL was chosen for nonlinear systems, whose typical reaching law is shown in Equation (10):

$$\dot{S}_i = -\frac{K_i}{D(S_i)}\text{sign}(S_i) \tag{10}$$

$$D(S_i) = \varsigma_i + (1-\varsigma_i)e^{-\rho_i|S_i|^{h_i}}$$

$$\text{sign}(S_i) = \begin{cases} 1, & \text{for } S_i > 0 \\ 0, & \text{for } S_i = 0 \\ -1, & \text{for } S_i < 0 \end{cases}$$

where K_i is a strictly positive gain, ς_i is a strictly positive offset less than one ($0 < \varsigma_i < 1$), h_i is a positive integer greater than zero ($h_i > 0$), and ρ_i is also a positive integer greater than zero ($\rho_i > 0$).

Relation (10) has the following features:

- In the event of an increase in $|S_i|$, $D(S_i) \to \varsigma_i$, leading to the convergence of $K_i/D(S_i)$ toward K_i/ς_i. Consequently, during the reaching phase, $K_i/D(S_i)$ increases, resulting in a faster attraction toward the sliding surface.
- In the case of a decrease in $|S_i|$, $D(S_i) \to 1$, causing $K_i/D(S_i)$ to converge to K_i. Consequently, $K_i/D(S_i)$ gradually decreases, thereby mitigating chattering.

Thus, the ERL does not impact the stability of control through sliding modes since $D(S_i)$ remains strictly positive. Additionally, it enables the controller to dynamically adapt to changes in the sliding surface by allowing the variation in $K_i/D(S_i)$ within the range of K_i to K_i/ς_i.

It is crucial to emphasize that the switching term $K_i sign(S_i)$ in Equation (10) frequently results in heightened control activity, commonly referred to as chattering (an undesirable phenomenon). To address this concern, the paper employs a smoothed fragmented term in the control input, as illustrated in Equation (11):

$$\dot{S}_i = -\frac{K_i}{D(S_i)}\text{sat}\left(\frac{S_i}{\psi_i}\right) \qquad \forall t, K_i > 0 \tag{11}$$

where

$$\text{sat}\left(\frac{S_i}{\psi_i}\right) = \begin{cases} 1, & \text{for } S_i \geq \psi_i \\ \frac{S_i}{\psi_i}, & \text{for } -\psi_i \leq S_i \leq \psi_i, \forall t, 0 < \psi_i \ll 1 \\ -1, & \text{for } S_i \leq \psi_i \end{cases}$$

By employing this substitution, the system's convergence is restricted to a boundary layer surrounding the sliding surface. The size of this neighborhood is directly determined by the parameter ψ_i. Therefore, Equation (11) effectively addresses chattering issues and enhances tracking performance.

In accordance with Equations (5), (9), and (11), we obtain ν_i:

$$\nu_i = \ddot{\theta}_{di} + \Lambda_i(\dot{\theta}_{di} - \dot{\theta}_i) + \frac{K_i}{D(S_i)}\text{sat}\left(\frac{S_i}{\psi_i}\right) \tag{12}$$

The online tuning process will determine the optimal gain values K_i and Λ_i, which adjust to changes during the trajectory tracking.

3.2.1. Controller Stability Analysis

To assess the controller stability, Equation (13) is chosen as the Lyapunov function.

$$H(S_i) = \frac{1}{2}S_i^2 \tag{13}$$

Hence, if $\dot{H} < 0$, it indicates $\dot{S}_i < 0$ for $S_i > 0$ and $\dot{S}_i > 0$ for $S_i < 0$ (refer to Equation (14)). This leads to the emergence of a switching phenomenon to uphold the condition $S_i\dot{S}_i < 0$.

$$\dot{H}(S_i) = S_i \dot{S}_i \tag{14}$$

where

$$\dot{S}_i = \Lambda_i \dot{e}_i + \ddot{e}_i \tag{15}$$

$$\ddot{e}_i = \ddot{\theta}_{di} - \ddot{\theta}_i \tag{16}$$

Replacing Equations (12), (15), and (16) into (14), we obtain the following:

$$\dot{H}(S_i) = S_i \left(\frac{-K_i}{D(S_i)} \text{sat}\left(\frac{S_i}{\psi_i} \right) - U_{ui} \right) \tag{17}$$

To ensure asymptotic stability, the following inequality must be satisfied:

$$|U_{ui}| < \frac{K_i}{D(S_i)} \tag{18}$$

where U_{ui} represents the unknown dynamics (external disturbances and model uncertainties), which is assumed to be locally continuous and limited by $Umax_i$ (Equation (19)):

$$0 < |U_{ui}| \leq Umax_i < \infty \tag{19}$$

3.3. Particle Swarm Optimization Algorithm

Kennedy and Eberhart introduced the conventional PSO algorithm in 1995. This metaheuristic draws inspiration from the collaborative behavior observed in fish and birds, relying on the cooperation of agents through the information sharing. Within the PSO algorithm, every agent is regarded as a particle, functioning as a potential solution in an optimization problem [52,53].

The fundamental model involves the initialization of a population with randomly selected solutions within the search space. The objective is to identify the optimal solution, with each particle continuously refining its behavior over iterations, leveraging its own experiences and insights gleaned from the top-performing particles within the swarm [8,15]. Every particle is defined by vectors representing its position and velocity (Equation (20b)).

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{20a}$$

$$V_i(t+1) = \omega V_i(t) + c_1 s_1 [P_b(t) - X_i(t)] + c_2 s_2 [P_g(t) - X_i(t)] \tag{20b}$$

where X_i denotes the particle position, V_i denotes the particle velocity, P_b denotes the best individual position of each particle i, P_g stands for the entire swarm's best global position, and ω denotes the inertia weight assigned to each particle; c_1 and c_2 denote the cognitive and social acceleration coefficients specific to each particle, where s_1 and s_2 denote two random numbers chosen independently from the interval [0, 1], respectively.

The main three parameters of the PSO algorithm are (1) inertial weight (ω), (2) the cognitive acceleration parameter (c_1), and (3) the social acceleration (c_2).These parameters impact the algorithm's ability to conduct both global and local searches, and its summarization is as follows: (1) the global search is enhanced with a large inertial weight, whereas the local search is enhanced with a small inertial weight; (2) if $c_1 > c_2$ then the global search is facilitated, and (3) if $c_2 > c_1$ then the local search is improved.

3.3.1. Convergence Analysis

As the PSO is a stochastic algorithm, it is imperative to provide a convergence analysis overview. To this end, this subsection considers its deterministic version and presents its convergence analysis based on the discrete-time dynamic system theory [52,53].

In this case, the random numbers are replaced by the average (0.5), so Equation (20b) is written as below:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{21a}$$

$$V_i(t+1) = \omega V_i(t) + c[P(t) - X_i(t)] \tag{21b}$$

where

$$P(t) = \frac{c_1 P_b(t) + c_2 P_g(t)}{c_1 + c_2} \tag{22a}$$

$$c = \frac{c_1 + c_2}{2} \tag{22b}$$

Equations (21a) and (21b) are reformulated in matrix form as follows:

$$\begin{bmatrix} X_i(t+1) \\ V_i(t+1) \end{bmatrix} = A \begin{bmatrix} X_i(t) \\ V_i(t) \end{bmatrix} + BP(t) \tag{23}$$

$$A = \begin{bmatrix} 1-c & \omega \\ -c & \omega \end{bmatrix}, B = \begin{bmatrix} c \\ c \end{bmatrix}$$

where $P(t)$ is the external input, while A and B correspond to the dynamical matrix and input matrix, respectively.

So, the time-dependent behavior of the PSO algorithm relies on the dynamic matrix A's eigenvalues. These eigenvalues are computed using Equation (24).

$$\det(\eta I - A) = 0 \tag{24}$$

The result of Equation (24) is shown in Equation (25):

$$\eta^2 + (c - 1 - \omega)\eta + \omega = 0 \tag{25}$$

The eigenvalues $\eta_{1,2}$ are shown in Equation (26):

$$\eta_{1,2} = \frac{\omega + 1 - c \pm \sqrt{\Delta}}{2} \tag{26}$$

where $\Delta = (\omega + 1 - c)^2 - 4\omega$. Therefore, the algorithm's effectiveness and convergence rely on the values assigned to the parameters ω and c.

The conditions derived from the eigenvalues given in Equation (26) are as follows: (1) $\omega < 1$, (2) $c > 0$, and (3) $2\omega - c + 2 > 0$.

The convergence region is illustrated in the (ω, c) plane, as depicted in Figure 1a. Regardless of the initial position and velocity, the particle will converge to its equilibrium position, as defined by Equations (22a) and (32), only if the algorithm parameters fall within this triangular region.

Before the convergence, the particle may exhibit the following behaviors:

1. When the eigenvalues of matrix A take on complex values, harmonic oscillations manifest around the equilibrium point. This equivalence is represented by Equation (27), and its domain in the plane is illustrated in Figure 1b.

$$\omega^2 + c^2 - 2\omega c - 2\omega - 2c + 1 < 0 \tag{27}$$

2. The oscillating pattern resembling zigzagging emerges near the equilibrium point when there is at least one eigenvalue of matrix A with a negative real part, whether it is a real or complex value. This relationship is expressed by either Equation (28) or (29), and its region in the plane is illustrated in Figure 1c.

$$\omega < 0 \tag{28}$$

$$\omega - c + 1 < 0 \tag{29}$$

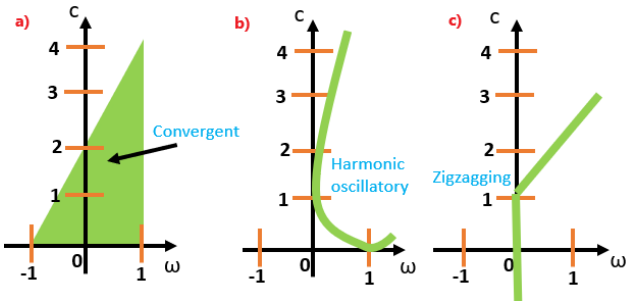


Figure 1. Regions of dynamic behavior within the parameter space (ω, c). (**a**) Convergence region, (**b**) harmonic oscillatory behavior, and (**c**) zigzagging behavior. Information taken from [53].

3.3.2. Equilibrium Point

The particle's initial state is typically not in a state of equilibrium. In practice, it is crucial to determine whether the particle will eventually reach equilibrium, signifying the optimization algorithm's convergence and the trajectory it will traverse in the state space while exploring for optimal points.

Equation (23) is written as follows:

$$Y_{(k+1)} = AY_k + BP(t) \tag{30}$$

where $Y_k = [X_k, V_k]^T$.

For an equilibrium point to exist, Equation (31) must be satisfied for any k.

$$Y^{eq}_{(k+1)} = Y^{eq}_k \tag{31}$$

In the deterministic analysis, the particle must possess zero velocity and be positioned at the attraction point P (Equation (22a)). Consequently, Equation (32) is obtained:

$$Y^{eq} = [P, 0]^T \tag{32}$$

It means $X^{eq} = P$ and $V^{eq} = 0$. Therefore, the equilibrium point is stable if and only if both eigenvalues of matrix A (real or complex) have magnitudes less than 1.

4. Results: Real-Time Implementation

This study performs gain tuning (K_i and Λ_i) of the ERL controller on a highly nonlinear exoskeleton robot, subject to external disturbances, such as initial offsets and user variations. The problem is formulated as an optimization task and addressed using the PSO algorithm.

4.1. Exoskeleton Description

The controller is applied to the ETS-MARSE (École de Technologie Supérieure—Motion Assistive Robotic-exoskeleton for Superior Extremity) with 7 DOF (see Figure 2). This exoskeleton facilitates rehabilitation by assisting movements for individuals with impaired upper extremities. Its design takes into account the upper extremity alignment, featuring 3 degrees of freedom (DOF) for the shoulder (S), 1 DOF for the elbow (E), and 3 DOF for the wrist (W). The modified Denavit–Hartenberg parameters (D-H) for the exoskeleton are detailed in Table 3.

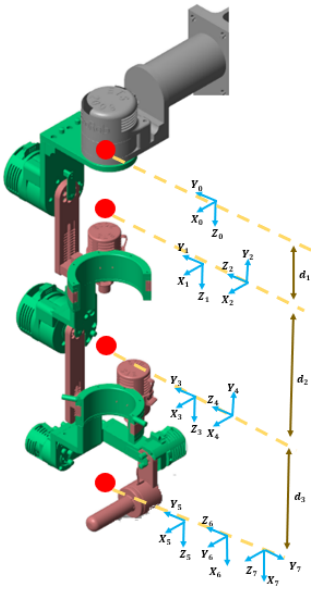


Figure 2. ETS MARSE: exoskeleton.

Table 3. Modified D-H parameters.

Movement	θ_i	a_{i-1}	α_{i-1}	d_i	Range [deg]
S: Horizontal Flex-Ext	θ_1	0	0	d_1	−20 to 70
S: Vertical Flex-Ext	θ_2	0	$-\frac{\pi}{2}$	0	−5 to 120
S: Int-Ext rot	θ_3	0	$+\frac{\pi}{2}$	d_2	−85 to 80
E: Ext-Flex	θ_4	0	$-\frac{\pi}{2}$	0	−5 to 115
W: Pron-sup	θ_5	0	$+\frac{\pi}{2}$	d_3	−85 to 85
W: Ext-Flex	$\theta_6 - \frac{\pi}{2}$	0	$-\frac{\pi}{2}$	0	−25 to 20
W: Ulnar-radial deviation	θ_7	0	$-\frac{\pi}{2}$	0	−55 to 60

The exoskeleton's real-time system consists of three processing units, detailed as follows:

- Host PC (Intel Core i7-4770 CPU @3.4 GHz, and 16 GB RAM): facilitates the transmission of higher-level controls to the exoskeleton via the human–machine interface (developed in LabView 2017).
- A real-time PC (NI PXI-8108, Intel dual-core @2.53 GHz processor and 8 GB RAM): operates the top-level control and manages the exoskeleton dynamics with a sampling time of 1 ms.
- FPGA (NI PXI-7813R): utilized for handling analog and digital inputs and outputs to the actuators and sensors, executing the low-level control (PI) current control loop with a sampling time of 50 µs.

The robot utilizes brushless DC motors, specifically Maxon EC-45 and Maxon EC-90, in combination with harmonic drives. Motors 1 and 2 have a gear ratio of 120:1, while motors 3–7 have a gear ratio of 100:1.

4.2. Tests Description

In this section, the tests to be conducted are described, with the following considerations:

1. The experiments were carried out by donning the exoskeleton on two healthy subjects (one at a time, as shown in Table 4 and Figure 3).
2. A trajectory for passive rehabilitation was executed that encompasses the movement of all joints, taking into account the modified D-H parameters, as detailed in Table 3. This trajectory starts with the user's elbow flexed at 90°.
3. During the experiment, the upper limb of the subjects remained in a passive state while the exoskeleton drives the user's extremity.

Table 4. Users' features.

User	Gender	Age (Years Old)	Weight [kg]	Height [cm]
$User_1$	Male	33	75	160
$User_2$	Female	33	54	154

(**a**) (**b**)

Figure 3. Exoskeleton with users. (**a**) User 1. (**b**) User 2.

To demonstrate the efficiency of the PSO algorithm when tuning the ERL controller gains, the following tests were performed:

1. Assess the ERL performance controller under pre-tuned gains, denoted as PT conditions.
2. Assess the ERL controller's real-time performance through online tuning using PSO, denoted as PSO conditions.
3. Assess the ERL controller's performance using the globally tuned gains obtained after applying the PSO algorithm, denoted as PSOG conditions.

4.2.1. First Test

The gains were chosen heuristically and theirs values are shown in Table 5.

Table 5. ERL: Pre-tuned gains.

Metric	q_1	q_2	q_3	q_4	q_5	q_6	q_7
K_i	50.0	20.0	90.0	60.0	250.0	500.0	450.0
Λ_i	90.0	40.0	70.0	20.0	25.0	10.0	10.0
ς_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Table 5. *Cont.*

Metric	q_1	q_2	q_3	q_4	q_5	q_6	q_7
ρ_i	2.0	2.0	2.0	2.0	2.0	2.0	2.0
h_i	2.0	2.0	2.0	2.0	2.0	2.0	2.0
ψ_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5

4.2.2. Second Test

The ERL controller parameters vary over time and are adaptively tuned online. This flexibility allows for efficient tracking of trajectories, even when confronted with external disturbances like patient forces or physiotherapist assistance. The control scheme is depicted in Figure 4.

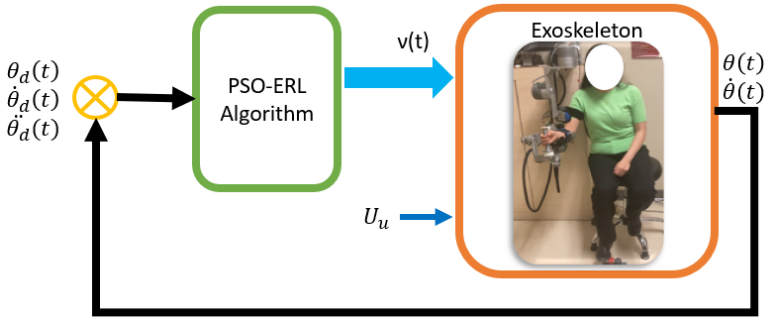


Figure 4. Control scheme.

Commonly, to address the optimization problem, a single objective function is used that encompasses all DOF. In this work, a weighted objective function is employed: it includes position and velocity errors for each degree of freedom (DOF), where the position error is prioritized. So, the function targeted as the objective in the PSO algorithm is presented in Equation (33).

$$f(e, \dot{e}) = \alpha \|e\| + \beta \|\dot{e}\| \tag{33}$$

Here, $\alpha + \beta = 1$, with α and β constrained to the range 0 to 1.

In this scenario, each particle comprises the following two gains, K_i and Λ_i. The goal is to dynamically identify the optimal particle in real time, minimizing the objective function $f(e, \dot{e})$. Therefore Algorithm 1 introduces the proposed robust PSO-ERL algorithm.

Algorithm 1 Proposed robust PSO-ERL algorithm.

1: Define the number of particles in the swarm, n.
2: Initialize the swarm with random values.
3: i $\leftarrow$ 1
4: **repeat**
5: Insert a particle i (K_i, Λ_i)
6: Compute ν using the ERL approach Equation (12)
7: Calculate $\tau_0(t)$ using Equation (7).
8: Utilize τ_0 as the input for the system.
9: Compute the objective function f using Equation (33).
10: Store the value of f for particle i.
11: i $\leftarrow$ i + 1
12: **until i** $>$ **m**
13: Determine $P_i(t)$ and $P_g(t)$.
14: Resolve Equation (20b).
15: **GOTO 4**

In this case, Equation (20a) represents the gain value (K_i or Λ_i), and Equation (20b) represents the corresponding gain change rate. The particle's initial values (K_i and Λ_i) are randomly generated within their respective search space, as defined in Table 6. Subsequently, the objective function is calculated in each iteration to assess each particle, iteratively guiding the PSO-ERL to find the optimal solution. The particle exhibiting the objective function minimum value corresponds to the best global gain.

Table 6. Search space.

Gains Constraints	q_1	q_2	q_3	q_4	q_5	q_6	q_7
K_{min}	25	10	70	40	200	460	430
K_{max}	70	60	130	80	320	540	550
Λ_{min}	70	30	60	2	20	5	1550
Λ_{max}	100	80	110	28	70	40	1950

The remaining parameters for the PSO algorithm and the controller's experimental conditions are detailed in Table 7.

Table 7. Experimental conditions (online gain tuning with PSO).

Metric	Value
Initial offset (each joint)	5 [deg]
External disturbances (see Table 4)	2 users
Number of particles (m)	30.0
ω (inertial factor of PSO)	0.9
c_1 (personal acceleration parameter of PSO)	0.1
c_2 (group acceleration parameter of PSO)	0.3
α (objective function, f)	0.7
β (objective function, f)	0.3
ς (controller parameter, ERL)	0.5
ρ (controller parameter, ERL)	2.0
h (controller parameter, ERL)	2.0
ψ (controller parameter, ERL)	0.5

To select the parameters in Table 7, we considered the following:

- $\omega \approx 1$: Promotes global exploration.
- $c_2 > c_1$: Global optimum is prioritized.
- $\alpha > \beta$: Prioritizing position error minimization.

Once the second test was completed (applying the PSO algorithm to determine the gains K_i and Λ_i of the controller), the values of the global gains were obtained for each user, which are shown in Table 8.

Table 8. Global gains tuned with ERL + PSO.

Metric	q_1	q_2	q_3	q_4	q_5	q_6	q_7
K_{User_1}	52.1	47.4	101.3	68.6	289.8	460.0	550.0
K_{User_2}	25.0	50.5	70.0	63.5	253.4	492.9	503.0
Λ_{User_1}	88.4	37.6	79.5	8.4	46.5	26.7	1654.7
Λ_{User_2}	99.9	80.0	94.1	17.4	24.9	10.4	1664.6

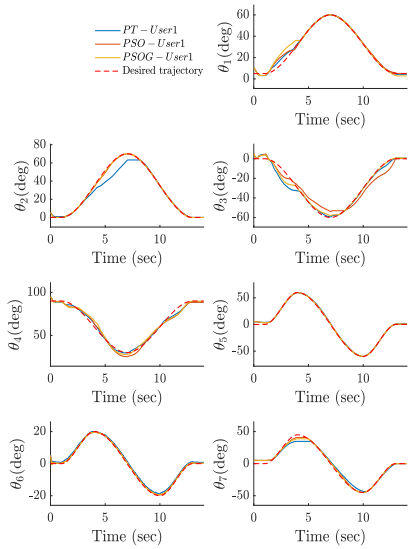
4.2.3. Third Test

It consists in executing the ERL controller with the gain values shown in Table 8.

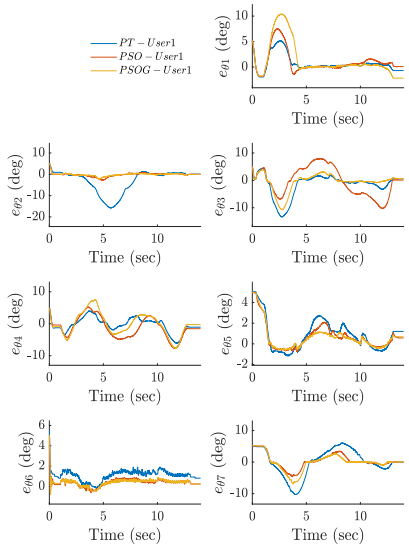
So, in the next section, we perform a comparative analysis of the results obtained in the three experiments.

4.3. Experimental Results and Comparative Analysis

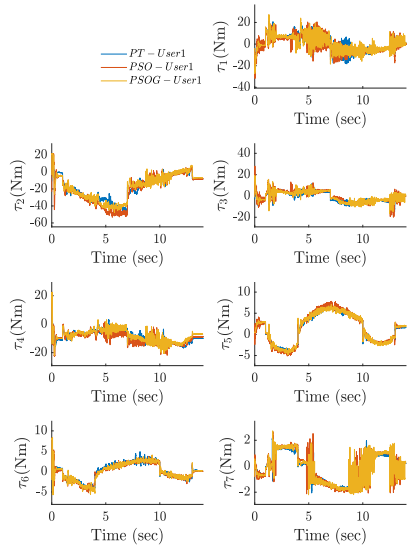
Figure 5 shows the results of the trajectory tracking, tracking error, control signals, and cartesian tracking and error graphs of user 1, respectively. Furthermore, Figure 6 corresponds to user 2. In each figure, there are three signals, each one corresponding to the experiments carried out (PT, PSO, and PSOG).



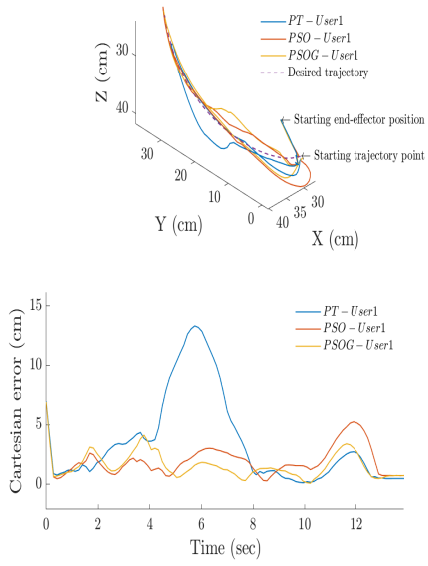
(**a**) Trajectory tracking



(**b**) Tracking error



(**c**) Control signal



(**d**) Cartesian tracking and error

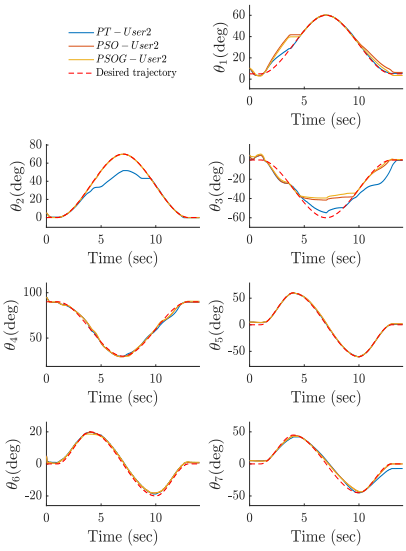
Figure 5. User 1. (**a**) Trajectory tracking. (**b**) Tracking error. (**c**) Control signal. (**d**) Cartesian tracking and error.

To evaluate the PSO algorithm's behavior, Figure 7 presents the evolution of the values of K and λ of a particle, showing an oscillatory and convergent behavior. Choosing the

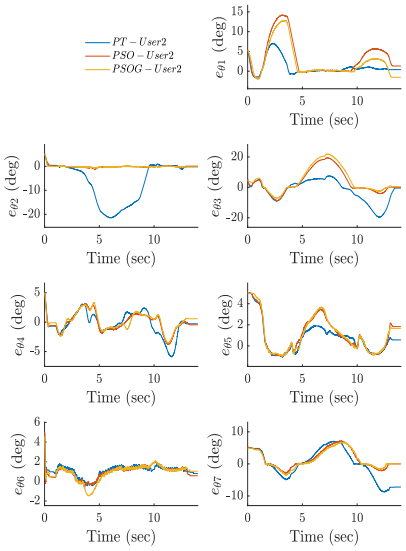
parameters ω, c_1, c_2 guarantees the PSO algorithm's deterministic convergence, whose eigenvalues of Equation (26) are equal to $\eta_{1,2} = 0.85 \pm j0.42$.

According to the results obtained, the average RMSE (root mean square error) for both users in the first test was $[0.05, 0.07] rad$ and the average RMSE for the same users in the third test was $[0.03, 0.05] rad$. This shows that the proposed approach reduces the average RMSE of user 1 by 28.3% and by 23.5% for user 2.

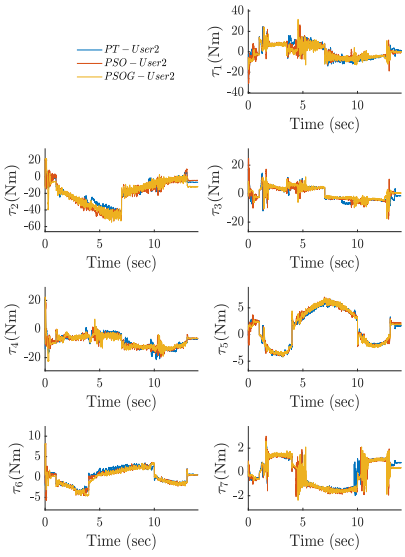
It should be noted that chattering is not eliminated entirely, however, the average RMST (root mean square torque) for the same users in the first test was $[7.33, 7.21]$ and for the third test was $[7.36, 7.33]$. It is concluded that the RMST in the third experiment increased by 0.4% and 1.6% compared to the first test, which means a small increase in the control activity while greatly improving the tracking error.



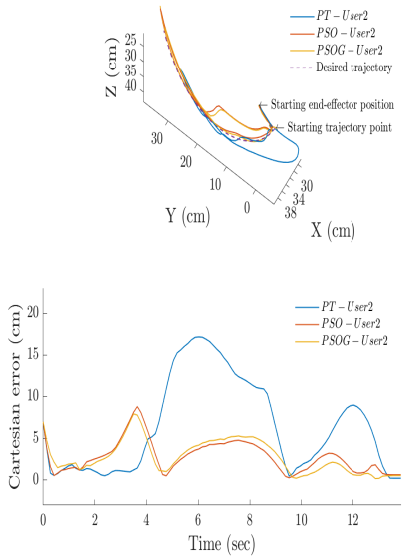
(**a**) Trajectory tracking



(**b**) Tracking error



(**c**) Control signal



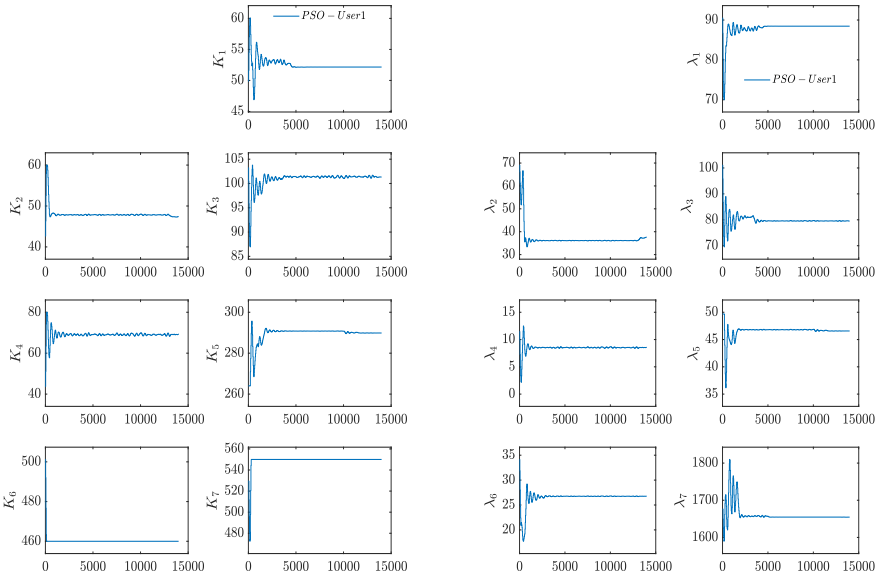
(**d**) Cartesian tracking and error

Figure 6. User 2. (**a**) Trajectory tracking. (**b**) Tracking error. (**c**) Control signal. (**d**) Cartesian tracking and error.

Tables 9 and 10 show the users' results for the RMSE and RMST of each joint, as well as the average for the first and third tests.

Table 9. Results: root mean square error (RMSE).

Joint	$User_1$			$User_2$		
	PT	PSOG	%	PT	PSOG	%
q_1	0.03	0.06	+110.0	0.03	0.08	+109.5
q_2	0.10	0.01	−88.1	0.17	0.00	−94.8
q_3	0.07	0.05	−27.6	0.13	0.16	25.6
q_4	0.03	0.06	+52.4	0.03	0.02	−22.0
q_5	0.03	0.02	−23.4	0.02	0.03	23.3
q_6	0.02	0.00	−55.3	0.02	0.02	−1.7
q_7	0.07	0.04	−38.7	0.08	0.05	−33.0
Mean	0.05	0.03	−28.3	0.07	0.05	−23.4



(a) K_i (b) Λ_i

Figure 7. Global gain evolution. (**a**) K_i. (**b**) Λ_i

Table 10. Results: root mean square torque (RMST).

Joint	$User_1$			$User_2$		
	PT	PSOG	%	PT	PSOG	%
q_1	7.45	7.24	−2.7	7.56	6.47	−14.4
q_2	22.66	22.89	+1.0	22.05	24.36	+10.4
q_3	4.58	4.91	+7.2	4.75	4.20	−11.7
q_4	9.86	9.35	−5.1	9.52	9.54	+0.1
q_5	3.60	3.72	+3.5	3.51	3.59	+2.2
q_6	2.08	2.18	+4.7	2.01	1.99	−0.9
q_7	1.11	1.21	+9.2	1.10	1.16	+5.2
Mean	7.33	7.36	+0.3	7.21	7.33	+1.5

Additionally, it is worth noting that the elapsed time for the first and third experiments was around 238 µs, while for the second experiment it was around 263 µs, representing an 11% increase.

5. Discussion

Rehabilitation exoskeletons are high-order systems with nonlinearities and are subject to external disturbances [29,46–48]. To optimize their performance, it is essential to develop controllers that account for user interaction and allow real-time dynamic adjustment of their parameters.

In most studies, controllers use constant gains that ensure adequate performance in experimental tests, enabling trajectory tracking adapted to a specific group of users [23,54–56]. However, online adaptive control is a promising alternative for improving system response to disturbances [12,15,19,21,29,55,57].

This study implements a robust control approach with dynamic gain adjustment, based on mathematical formulations that consider modeling uncertainties and disturbances [23,27,28,30]. A sliding mode controller with an exponential reaching law was used [31,50], which simultaneously reduced chattering and convergence time without compromising system robustness. Additionally, it provides a quick response to system changes, resistance to parameter variations, and ease of implementation.

To optimize the gains of the nonlinear controller in the presence of disturbances, the PSO algorithm was employed due to its ability to find optimal solutions in problems with multiple local minima, its low computational cost, and its adaptability to various applications [15,44,45].

To the best of our knowledge, this is the first study to experimentally apply this method to a 7 DOF exoskeleton for online gain adjustment in a sliding mode controller. The results obtained with healthy subjects of different morphologies showed an improvement of over 20% in trajectory tracking, along with a significant reduction in chattering.

This work presents a promising solution for tuning robust controllers in dynamically coupled systems. However, controller performance depends on the correct selection of gains within the search space, defined according to the system's physical characteristics. If the exoskeleton lacks dynamic stability and energy efficiency, the algorithm alone will not guarantee significant improvements.

6. Conclusions

This work addresses the challenge of dynamically optimizing the parameters of an exoskeleton in real time for its application in physical rehabilitation, successfully providing unique and optimal parameters for each patient, which improved movement performance. Consequently, a promising solution was presented for tuning robust controllers in coupled and continuously varying systems. It is important to note that the controller's performance is limited by the gains within the search space, which depend on the exoskeleton's physical characteristics. Thus, the efficacy of the proposed algorithm in enhancing control performance is contingent upon the exoskeleton's capacity for robust dynamic stability and energy efficiency.

Obtaining gains from the exponential reaching law controller via the particle swarm optimization algorithm reduced the error in trajectory tracking by more than 20% in two users with different morphologies, opening the path for its applications on different users with different motor conditions. Furthermore, the proposed controller demonstrates the capacity to attenuate the chattering phenomenon, even when the tracking error is enhanced, while concurrently maintaining a low execution time. For future work, an

improved version of the PSO algorithm is proposed to further optimize trajectory tracking and reduce energy consumption.

Author Contributions: Conceptualization, D.S.M. and D.B.-M.; Investigation, D.S.M.; Methodology, D.S.M.; Formal analysis, D.S.M. and D.B.-M. (controller part); Implementation, D.S.M. and D.B.-M.; Validation, D.S.M. and D.B.-M.; Writing—original draft, D.S.M.; Reviewing, D.B.-M., M.S., Y.K., C.E.G.C. and Á.L.Á.; Supervision, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by grant PDC2023-145812-I00 (Project SAMPL2D), which is funded by MICIU/AEI/10.13039/501100011033 and by "Next Generation EU/PRTR".

Data Availability Statement: Data will be made available on request.

Acknowledgments: The first and second authors would like to thank CONAHCYT (Mexican National Council of Humanities, Science and Technology) for the support under grants 795353 and 739833, respectively. CONAHCYT was previously named CONACYT (Mexican National Council of Science and Technology).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant colony optimization
CS	Cuckoo search
DE	Differential evolution
DOF	Degree of fredom
E	Elbow
Exp	Experimental
FL	Fuzzy logic
GA	Genetic algorithm
HS	Healthy subject
LLE	Lower limb exoskeleton
NN	Neural network
PID	Proportional–integral–derivative
PSO	Particle swarm optimization
S	Shoulder
SA	Simulated annealing
Sim	Simulation
SMC	Sliding mode control
ST	Supertwisting
ULE	Upper limb exoskeleton
W	Wrist

References

1. Cieza, A.; Causey, K.; Kamenov, K.; Hanson, S.W.; Chatterji, S.; Vos, T. Global estimates of the need for rehabilitation based on the Global Burden of Disease study 2019: A systematic analysis for the Global Burden of Disease Study 2019. *Lancet* **2020**, *396*, 2006–2017. [CrossRef]
2. World Health Organization. Musculoskeletal Health. July 2022. Consult: December 2023. Available online: https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions (accessed on 22 May 2024) .
3. Moulaei, K.; Bahaadinbeigy, K.; Haghdoostd, A.A.; Nezhad, M.S.; Sheikhtaheri, A. Overview of the role of robots in upper limb disabilities rehabilitation: A scoping review. *Arch. Public Health* **2023**, *81*, 84. [CrossRef] [PubMed]

4. Choutri, K.; Fareh, R.; Rahman, M.H.; Bettayeb, M.; Fadloun, S.; Lagha, M. Reinforcement Learning Fractional Order PID Controller For Upper Limb Rehabilitation Robot. In Proceedings of the 2023 International Conference on Fractional Differentiation and Its Applications (ICFDA), Ajman, United Arab Emirates, 14–16 March 2023; pp. 1–6. [CrossRef]
5. Rocon, E.; Pons, J.L. *Exoskeletons in Rehabilitation Robotics: Tremor Suppression*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 69.
6. Pons, J.L. *Wearable Robots: Biomechatronic Exoskeletons*; John Wiley & Sons Ltd.: Chichester, UK, 2008.
7. Gull, M.A.; Bai, S.; Bak, T. A review on design of upper limb exoskeletons. *Robotics* **2020**, *9*, 16. [CrossRef]
8. Wang, J.; Tang, B.; Pang, M.; Xiang, K.; Ju, Z. Self-adaptive Particle Swarm Optimization with Human-in-the-loop for Ankle Exoskeleton Control. *Sens. Mater.* **2021**, *33*, 3227. [CrossRef]
9. de La Rosa-Hernández, G.A.; Sernaque-Julca, J.C.; Quiroz-Compeán, G.; Rodríguez-Liñan, J.Á; Torres-Treviño, L.M. Tuning of a Control Scheme for an Exoskeleton Using Swarm Intelligence. In Proceedings of the 2023 IEEE EMBS R9 Conference, Guadalajara, Mexico, 5–7 October 2023; IEEE: New York, NY, USA, 2023; pp. 1–4.
10. Sosa Méndez, D.; García Cena, C.E.; Bedolla-Martínez, D.; Martín González, A. Innovative Metaheuristic Optimization Approach with a Bi-Triad for Rehabilitation Exoskeletons. *Sensors* **2024**, *24*, 2231. [CrossRef]
11. Habiba, K.N.; Arifin, A.; Babgei, A.F.; Risciawan, A.; Zazuli, M.I. Design of Fuzzy-PI Controller for Shoulder Exoskeleton with Motoric Progress of Rehabilitation Subject Consideration. In Proceedings of the 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 22–23 November 2022; pp. 32–37. [CrossRef]
12. Safira, L.A.; Arifin, A.; Kusuma, H.; Risciwan, A.; Zazuli, M.I. Adaptive PID Controller Based on Sliding Surface for Controlling Elbow Joint Robot Rehabilitation. In Proceedings of the 2023 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 26–27 July 2023; pp. 316–321. [CrossRef]
13. Sidabalok, S.M.; Arifin, A.; Pramudijanto, J.; Kusuma, H.; Risciwan, A.; Zazuli, M.I. Development of Elbow Joint Exoskeleton Control System Using Fuzzy Pid Control Method for Post Stroke Rehabilitation. In Proceedings of the 2023 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 26–27 July 2023; pp. 310–315. [CrossRef]
14. de Medeiros, R.B.; Muñoz, D.M. Tuning of FP-PID Controller based on PSO Algorithm Applied to a Human Gait. In Proceedings of the 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), São Bernardo do Campo, Brazil, 18–21 October 2022; pp. 1–6. [CrossRef]
15. Belkadi, A.; Oulhadj, H.; Touati, Y.; Khan, S.A.; Daachi, B. On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. *Appl. Soft Comput.* **2017**, *60*, 87–100. [CrossRef]
16. Han, Y.; Liu, C.; Xiu, H.; Li, Z.; Shan, S.; Wang, X.; Ren, L.; Ren, L. Trajectory Control of an Active and Passive Hybrid Hydraulic Ankle Prosthesis Using an Improved PSO-PID Controller. *J. Intell. Robot. Syst.***2022**, *105*, 48. [CrossRef]
17. Belov, M.P.; Truong, D.D.; van Tuan, P. Self-Tuning PID Controller Using a Neural Network for Nonlinear Exoskeleton System. In Proceedings of the 2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT), Saint Petersburg, Russia, 16 June 2021; pp. 6–9. [CrossRef]
18. Al-Waeli, K.H.; Ramli, R.; Haris, S.M.; Zulkoffli, Z.B.; Amiri, M.S. Offline ANN-PID Controller Tuning on a Multi-Joints Lower Limb Exoskeleton for Gait Rehabilitation. *IEEE Access* **2021**, *9*, 107360–107374. [CrossRef]
19. Amiri, M.S.; Ramli, R.; Ibrahim, M.F.; Wahab, D.A.; Aliman, N. Adaptive Particle Swarm Optimization of PID Gain Tuning for Lower-Limb Human Exoskeleton in Virtual Environment. *Mathematics* **2020**, *8*, 2040. [CrossRef]
20. Belov, M.P.; Truong, D.D.; Khoa, T.D. Self-Tuning Subordinate Control System Based on Neural Network for Nonlinear Electric Drivers of Lower Limbs of Exoskeleton. In Proceedings of the 2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), Saint Petersburg, Russian, 25–28 January 2022; pp. 549–553. [CrossRef]
21. Liu, J.; Fang, H.; Xu, J. Online Adaptive PID Control for a Multi-Joint Lower Extremity Exoskeleton System Using Improved Particle Swarm Optimization. *Machines* **2021**, *10*, 21. [CrossRef]
22. Dali, Hassen, M.; Laamiri, I.; Bouguila, N. A Robust Adaptive Non-Singular Terminal Sliding Mode Control: Application to an Upper-Limb Exoskeleton with Disturbances and Uncertain Dynamics. *Inf. Technol. Control* **2024**, *53*, 171–186. [CrossRef]
23. Mahmoud, I.; Saidi, I. Trajectory Tracking Control of Upper-Limb Rehabilitation Exoskeleton Based on Robust Control. In Proceedings of the 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC ASET), Hammamet, Tunisia, 22–25 March 2022; pp. 330–335. [CrossRef]
24. Bembli, S.; Haddad, N.K.; Belghith, S. An exoskeleton – upper limb system control using a robust Model free terminal sliding mode with EMG signal. In Proceedings of the 2021 International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France, 3–5 November 2021; pp. 1–8. [CrossRef]
25. Hassen, M.D.; Jouirou, R.; Bouguila, N. Robust Fuzzy Terminal Sliding Mode Control for Passive Control of Upper-Limb Exoskeleton. In Proceedings of the 2024 International Conference on Control, Automation and Diagnosis (ICCAD), Paris, France, 15–17 May 2024; pp. 1–6. [CrossRef]
26. Jebri, A.; Madani, T.; Djouani, K.; Benallegue, A. Robust adaptive neuronal controller for exoskeletons with sliding-mode. *Neurocomputing* **2020**, *399*, 317–330. [CrossRef]

27. Silawatchananai, C.; Howimanporn, S. Robust control design of arm exoskeleton based on sliding mode control. In Proceedings of the 2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI), Kitakyushu, Japan, 1–15 September 2020; pp. 542–547. [CrossRef]
28. Narayan, J.; Bharti, R.R.; Dwivedy, S.K. Robust Sliding Mode Control with Reaching Laws for a Pediatric Lower-Limb Exoskeleton System. In Proceedings of the 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 1–3 March 2022; pp. 1–6. [CrossRef]
29. Ratiba, F.; Mohamed, G.; Mustapha, H.; Noura, A. Adaptive Finite-Time Robust Sliding Mode Controller For Upper Limb Exoskeleton Robot. In Proceedings of the 2022 19th International Multi-Conference on Systems, Signals & Devices (SSD), Sétif, Algeria, 6–10 May 2022; pp. 1255–1260. [CrossRef]
30. Zhang, G.; Wang, J.; Yang, P.; Guo, S. A learning control scheme for upper-limb exoskeleton via adaptive sliding mode technique. *Mechatronics* **2022**, *86*, 102832. [CrossRef]
31. Brahmi, B.; El-Monajjed, K.; Rahman, M.H.; Ahmed, T.; El-Bayeh, C.; Khan, M.R.; Saad, M. Novel Adaptive Reaching Law for Sliding Mode Control of an Upper Limb Exoskeleton Robot. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 1432–1437. [CrossRef]
32. He, D.; Wang, H.; Tian, Y.; Guo, Y. A Fractional-Order Ultra-Local Model-Based Adaptive Neural Network Sliding Mode Control of n -DOF Upper-Limb Exoskeleton with Input Deadzone. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 760–781. [CrossRef]
33. Fazli, E.; Rakhtala, S.M.; Mirrashid, N.; Karimi, H.R. Real-time implementation of a super twisting control algorithm for an upper limb wearable robot. *Mechatronics* **2022**, *84*, 102808. [CrossRef]
34. Tiaiba, H.; Daachi, M.E.H.; Madani, T. Real-time adaptive super twisting algorithm based on PSO algorithm: Application for an exoskeleton robot. *Robotica* **2024**, *42*, 1816–1841. [CrossRef]
35. Aichaoui, M.; Ikhlef, A. Adaptive Backstepping Control for Upper Limb Rehabilitation Robot using PSO Tuning. In Proceedings of the 2022 19th International Multi-Conference on Systems, Signals & Devices (SSD), Sétif, Algeria, 6–10 May 2022; pp. 601–608. [CrossRef]
36. Dong, J.; Jia, Z.; Li, E.; Lv, Q. AAN Controller with Adaptive Gain for Upper Limb Exoskeleton. *IEEE Access* **2024**, *12*, 112767–112775. [CrossRef]
37. Xia, P.; Yang, Y.; Jin, S.; Liu, X.; Dong, X. Neural Networks Iterative Learning Impedance Control of Lower Limb Exoskeleton for the Later Stage of Rehabilitation. In Proceedings of the 2022 IEEE 11th Data Driven Control and Learning Systems Conference (DDCLS), Chengdu, China, 3–5 August 2022; pp. 250–255. [CrossRef]
38. Huang, P.; Yuan, W.; Li, Q.; Feng, Y. Neural Network-Based Optimal Control of a Lower-limb Exoskeleton Robot. In Proceedings of the 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), Chongqing, China, 3–5 July 2021; pp. 154–159. [CrossRef]
39. Agnanto, D.; Arifin, A.; Babgei, A.F. Multi-input Multi-output Fuzzy Logic Controller for Hybrid Exoskeleton and Functional Electrical Stimulation for Hand Movements Rehabilitation of Hemiparesis Patients. In Proceedings of the 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 21–22 July 2021; pp. 272–277. [CrossRef]
40. Sangeetha, T.S.; Soman, S.; Sivanandan, K.S.S.; Parameswaran, A.P.; Baiju, T. Intelligent Control of Exoskeletons for Human Limbs using Knowledge-Based Fuzzy Inference System. In Proceedings of the 2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 October 2023; pp. 173–178. [CrossRef]
41. Dan, S.; Cheng, H.; Zhang, Y.; Liu, H. A Fuzzy Indrect Adaptive Robust Control for Upper Extremity Exoskeleton Driven by Pneumatic Artificial Muscle. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, Guangxi, China, 7–10 August 2022; pp. 839–846. [CrossRef]
42. Soman, S.; Sangeetha, T.S.; Sivanandan, K.S.S.; Parameswaran, A.P.; Baiju, T. Personalized Estimation and Modification of Elbow Angular Displacement for Upper Limb Exoskeletons:A Fuzzy Logic-Based Approach. In Proceedings of the 2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 October 2023; pp. 179–184. [CrossRef]
43. Bedolla-Martinez, D.; Kali, Y.; Saad, M.; Ochoa-Luna, C.; Rahman, M.H. Learning human inverse kinematics solutions for redundant robotic upper-limb rehabilitation. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106966. [CrossRef]
44. Samala, R.K. Particle Swarm Optimization. In *Swarm Intelligence-Recent Advances and Current Applications*; IntechOpen: London, UK, 2023.
45. Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [CrossRef]
46. Fitzsimons, K.; Kalinowska, A.; Dewald, J.P.; Murphey, T.D. Task-based hybrid shared control for training through forceful interaction. *Int. J. Robot. Res.* **2020**, *39*, 1138–1154. [CrossRef]
47. Saveriano, M.; Abu-Dakka, F.J.; Kramberger, A.; Peternel, L. Dynamic movement primitives in robotics: A tutorial survey. *Int. J. Robot. Res.* **2023**, *42*, 1133–1184. [CrossRef]

48. Dehio, N.; Smith, J.; Wig, D.L.; Mohammadi, P.; Mistry, M.; Steil, J.J. Enabling impedance-based physical human–multi–robot collaboration: Experiments with four torque-controlled manipulators. *Int. J. Robot. Res.* **2022**, *41*, 68–84. [CrossRef]
49. Sreejeth, M.; Kumar, R.; Tripathi, N.; Garg, P. Tuning A PID Controller using Metaheuristic Algorithms. In Proceedings of the 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 1–3 June 2023; pp. 276–282. [CrossRef]
50. Komurcugil, H.; Bayhan, S.; Guler, N.; Abu-Rub, H. A New Exponential Reaching Law Approach to the Sliding Mode Control: A Multilevel Multifunction Converter Application. *IEEE Trans. Ind. Electron.* **2023**, *70*, 7557–7568. [CrossRef]
51. Fallaha, C.J.; Saad, M.; Kanaan, H.Y.; Al-Haddad, K. Sliding-Mode Robot Control with Exponential Reaching Law. *IEEE Trans. Ind. Electron.* **2011**, *58*, 600–610. [CrossRef]
52. Gopal, A.; Sultani, M.M.; Bansal, J.C. On stability analysis of particle swarm optimization algorithm. *Arab. J. Sci. Eng.* **2020**, *45*, 2385–2394. [CrossRef]
53. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [CrossRef]
54. Palazzi, E.; Luzi, L.; Dimo, E.; Meneghetti, M.; Vicario, R.; Luzia, R.F.; Vertechy, R.; Calanca, A. An affordable upper-limb exoskeleton concept for rehabilitation applications. *Technologies* **2022**, *10*, 22. [CrossRef]
55. Aole, S.; Elamvazuthi, I.; Waghmare, L.; Patre, B.; Bhaskarwar, T.; Prasetyo, T. ADRC for Upper Limb Exoskeleton: A Simulation Study. In Proceedings of the 2022 IEEE 5th International Symposium in Robotics and Manufacturing Automation (ROMA), Malacca, Malaysia, 6–8 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
56. Qureshi, M.U.; Mudassir, M. Robust Control Design For 3 Degree Of Freedom Fully Actuated Upper Limb Exoskeleton. In Proceedings of the 2021 16th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 22–23 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
57. Zhao, Y.; Wu, H.; Zhang, M.; Mao, J.; Todoh, M. Design methodology of portable upper limb exoskeletons for people with strokes. *Front. Neurosci.* **2023**, *17*, 1128332. [CrossRef]