

## Article

# Human–Robot Cooperation Control Strategy Design Based on Trajectory Deformation Algorithm and Dynamic Movement Primitives for Lower Limb Rehabilitation Robots

Jie Zhou <sup>1,2</sup>, Yao Sun <sup>1,2</sup>, Laibin Luo <sup>1,2</sup>, Wenxin Zhang <sup>1,2</sup> and Zhe Wei <sup>1,2,\*</sup>

<sup>1</sup> School of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China; zhoujie@sut.edu.cn (J.Z.); sunyao@sut.edu.cn (Y.S.); luolaibin63@163.com (L.L.); zwx2693882648@163.com (W.Z.)

<sup>2</sup> The Key Laboratory of Intelligent Manufacturing and Industrial Robots in Liaoning Province, Shenyang 110870, China

\* Correspondence: weizhe@sut.edu.cn; Tel.: +86-186-4206-0001

**Abstract:** Compliant physical interactions, interactive learning, and robust position control are crucial to improving the effectiveness and safety of rehabilitation robots. This paper proposes a human–robot cooperation control strategy (HRCCS) for lower limb rehabilitation robots. The high-level trajectory planner of the HRCCS consists of a trajectory generator, a trajectory learner, a desired trajectory predictor, and a soft saturation function. The trajectory planner can predict and generate a smooth desired trajectory through physical human–robot interaction (pHRI) in a restricted joint space and can learn the desired trajectory using the locally weighted regression method. Moreover, a triple-step controller was designed to be the low-level position controller of the HRCCS to ensure that each joint tracks the desired trajectory. A nonlinear disturbance observer is used to observe and compensate for total disturbances. The radial basis function neural networks (RBFNN) approximation law and robust term are adopted to compensate for observation errors. The simulation results indicate that the HRCCS is robust and can achieve compliant pHRI and interactive trajectory learning. Therefore, the HRCCS has the potential to be used in rehabilitation robots and other fields involving pHRI.

**Keywords:** human–robot cooperation control; lower limb rehabilitation robots; physical human–robot interaction; interactive learning



**Citation:** Zhou, J.; Sun, Y.; Luo, L.; Zhang, W.; Wei, Z. Human–Robot Cooperation Control Strategy Design Based on Trajectory Deformation Algorithm and Dynamic Movement Primitives for Lower Limb Rehabilitation Robots. *Processes* **2024**, *12*, 924. <https://doi.org/10.3390/pr12050924>

Academic Editors: Ravendra Singh and Chih-Te Yang

Received: 17 March 2024

Revised: 17 April 2024

Accepted: 27 April 2024

Published: 1 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Annually, approximately 5.5 million people die from strokes, while more than 44 million survive them [1]. Gait training is essential to stroke rehabilitation because almost two-thirds of stroke survivors have an initial motor impairment and more than 30% of them cannot walk independently [2]. Lower limb rehabilitation robots (LLRRs) have recently drawn increased attention because they enable patients to walk naturally on the ground and the clinical effectiveness of LLRRs has been proven [3,4].

Control strategies are crucial for rehabilitation robots, and they are usually reasonably selected according to the patient's motor disability level. Various passive control strategies have been proposed for periodic, repetitive motion control in patients with weak residual muscle strength. Proportional–integral–derivative controllers and their variants are commonly used in LLRRs to help patients track the reference trajectory [5,6]. However, given their high nonlinearity and dynamic coupling, achieving accurate tracking with a model-free proportional–integral–derivative controller is challenging for LLRRs [7]. Computed torque control can achieve feedback linearization of a nonlinear dynamic model, but the model parameters' uncertainties and external disturbances affect its control performance [8]. In [9], an output-constrained controller was proposed for a lower limb exoskeleton, in which a finite-time extended state observer was used to estimate and compensate for

unmeasured joint velocity and lumped uncertainty. A triple-step controller with a linear extended observer was proposed and applied in the LLRR, achieving higher accuracy and lower energy consumption than the other comparison controllers [10]. Moreover, an adaptive neural network-based saturated controller was designed for LLRRs, in which the radial basis function neural networks (RBFNN) and robust terms were adopted to compensate for the unknown dynamics [11]. However, those passive control strategies reject patients' active contribution during rehabilitation training.

Active control strategies are suitable for patients with residual muscle strength, encouraging patients to participate actively in rehabilitation training. In [12], the position-constrained assist-as-needed control strategy was proposed for a knee exoskeleton and can smoothly switch between human-dominated and robot-dominated modes. In [13], a gait phase blended controller was designed for LLRRs to enhance transparency, solving the torque discontinuity problem and increasing the stride length. Similarly, to the path controller [14], a human-cooperative adaptive fuzzy controller was proposed, improving the motor variability of users during robot-assisted walking [15]. Compared with the active control strategies implemented by low-level position controllers, the active control strategies designed based on high-level trajectory planners are more straightforward in expanding their applications. In [16], a time-invariant force-field-based path controller was designed for the Indego exoskeleton, allowing users to adjust the step time by changing the inter-joint coordination tunnel shape. Many admittance control strategies were designed for LLRRs to establish a more compliant behavior, thus reshaping the reference trajectory through physical human–robot interaction (pHRI) [17,18]. Considering the influence of pHRI on both the current and future states of robotic systems [19], a human–robot cooperation controller has been proposed based on the trajectory deformation algorithm (TDA), which can generate and predict the desired trajectory [20]. Additionally, the dynamic movement primitive (DMP) method is well established for learning and generalizing trajectories [21,22] and can update the weighted parameters incrementally to model reference trajectories and describe varying walking patterns [23]. However, to our knowledge, a method for designing a human–robot cooperation control strategy (HRCCS) based on the TDA and DMPs that can predict, generate, and learn desired trajectories while ensuring trajectory tracking accuracy and robustness has not been investigated.

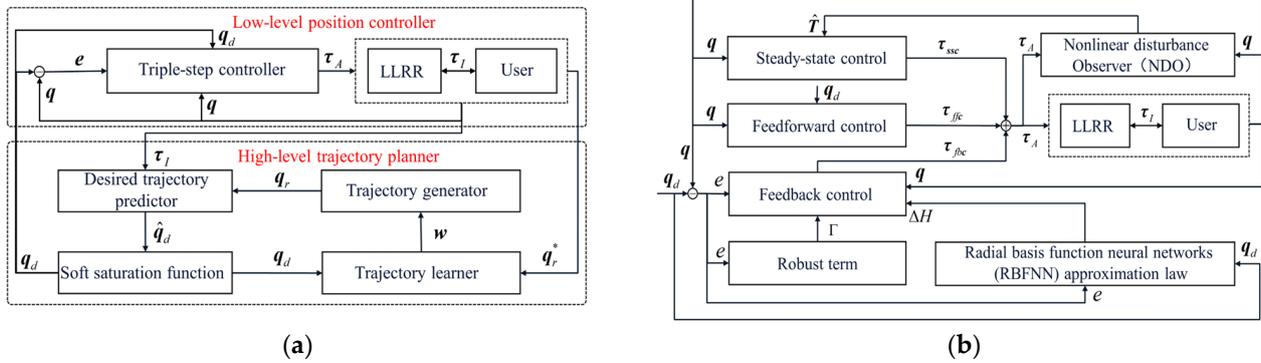
In this article, we propose a human–robot cooperation control strategy (HRCCS) for LLRRs, including a high-level trajectory planner and a low-level position controller. The main contributions are summarized as follows:

- (1) We design a high-level trajectory planner based on the TDA, DMPs, and the soft saturation function. It can generate reference trajectories by generalizing the trajectory model, predict a smooth desired trajectory through pHRI, and learn the desired trajectory using the locally weighted regression method.
- (2) We propose a triple-step controller as the low-level position controller to ensure trajectory tracking accuracy and robustness. A nonlinear disturbance observer (NDO) is used to observe and compensate for total disturbances, and the radial basis function neural networks (RBFNN) approximation law and robust terms are adopted to compensate for observation errors.
- (3) We conduct robustness verification, compliant interaction, and trajectory interactive learning simulation experiments. The results indicate that the HRCCS can ensure trajectory tracking accuracy and robustness and can achieve compliant pHRI and trajectory interactive learning.

## 2. Methods

Illustrated in Figure 1a is the HRCCS for LLRRs, including a high-level trajectory planner and a low-level position controller. The high-level trajectory planner consists of a trajectory generator, a trajectory learner, a desired trajectory predictor, and a soft saturation function and can generate, predict, and learn the desired trajectory in a restricted joint space. Moreover, a triple-step controller is designed to be the low-level position controller. As

shown in Figure 1b, the triple-step controller consists of steady-state control, feedforward control, and feedback control. The NDO and the RBFNN approximation law are integrated into the steady-state control and feedback control to enhance the accuracy and robustness of trajectory tracking.



**Figure 1.** Control framework for the LLRR: (a) Control framework of the HRCCS; (b) Control framework of the triple-step controller.

### 2.1. High-Level Trajectory Planner

The high-level trajectory planner comprises a trajectory generator, a trajectory learner, a desired trajectory predictor, and a soft saturation function. Each part will be explained in detail in the following subsections.

#### 2.1.1. Trajectory Generator

Since DMPs can generalize motion excellently, the trajectory generator based on DMPs can generate periodic reference trajectories for LLRRs [21]. It can be described as follows:

$$\begin{cases} \kappa \dot{z}_q = \alpha_q (\beta_q (\mathbf{g} - \mathbf{q}_r) - \mathbf{z}_q) + \mathbf{f} \\ \kappa \dot{q}_r = \mathbf{z}_q \end{cases}, \tag{1}$$

$$\kappa \dot{\phi} = 1, \tag{2}$$

$$\kappa \dot{r} = \alpha_r (r_0 - r), \tag{3}$$

$$\mathbf{f} = \frac{\sum_{i=1}^m \psi_i \mathbf{w}_i}{\sum_{i=1}^m \psi_i} \mathbf{r}, \tag{4}$$

$$\psi_i = \exp\left(\frac{\cos(\phi - c_i) - 1}{2\sigma_i^2}\right). \tag{5}$$

where  $\kappa$  is the positive temporal scaling factor;  $\alpha_q$  and  $\beta_q$  are positive constants that ensure the nonlinear system is in a critical damping state;  $\mathbf{g} \in \mathbb{R}^2$  and  $\mathbf{f} \in \mathbb{R}^2$  are the position goal and forcing term vector;  $\mathbf{q}_r \in \mathbb{R}^2$  and  $\dot{\mathbf{q}}_r \in \mathbb{R}^2$  are the reference angle and angular velocity vector;  $\mathbf{z}_q \in \mathbb{R}^2$  and  $\dot{\mathbf{z}}_q \in \mathbb{R}^2$  are the reference angular velocity and angular acceleration vector after expansion or contraction;  $r$  and  $r_0$  are the amplitude modulation factor and the initial value of amplitude modulation factor, respectively;  $\alpha_r$  is a positive constant that determines the speed of the amplitude modulation factor change;  $\phi$  is a phase variable;  $\psi_i$  is the  $i_{th}$  kernel function;  $\sigma_i$  and  $c_i$  are constants that determine the width and center of the  $i_{th}$  kernel function;  $m$  is the number of kernel functions; and  $\mathbf{w}_i \in \mathbb{R}^2$  is the weight coefficient vector corresponding to the kernel function  $\psi_i$ .

### 2.1.2. Trajectory Learner

The trajectory learner based on DMPs can enable robots to learn reference trajectories and desired trajectories online, which is particularly helpful for LLRRs in handling time-varying interaction dynamics and adapting to varying walking patterns. The trajectory learning process is as follows:

Firstly, Equation (1) can be rewritten as follows to calculate the forcing term vector:

$$\kappa^2 \ddot{\mathbf{q}}_r + \alpha_q \kappa \dot{\mathbf{q}}_r - \alpha_q (\beta_q (\mathbf{g} - \mathbf{q}_r)) = \mathbf{f}. \quad (6)$$

Given the target reference angle vector  $\mathbf{q}_r^*$ , angular velocity vector  $\dot{\mathbf{q}}_r^*$ , and angular acceleration vector  $\ddot{\mathbf{q}}_r^*$  and substituting them into Equation (6), we can obtain the target forcing term vector as follows:

$$\kappa^2 \ddot{\mathbf{q}}_r^* + \alpha_q \kappa \dot{\mathbf{q}}_r^* - \alpha_q \beta_q (\mathbf{g} - \mathbf{q}_r^*) = \mathbf{f}^*. \quad (7)$$

Secondly, the locally weighted regression method [21] is adopted to minimize the quadratic error function:

$$J_i = \sum_{n=1}^P \psi_i(n) (f_k(n) - w_{k,i} r)^2. \quad (8)$$

The solution to the above problem is as follows:

$$\begin{cases} w_{k,i} = \mathbf{s}^T \mathbf{J}_i \mathbf{f}_k^* / \mathbf{s}^T \mathbf{J}_i \mathbf{s} \\ \mathbf{s} = [r(1), r(2), \dots, r(P)]^T \\ \mathbf{J}_i = \text{diag}\{\psi_i(1), \psi_i(2), \dots, \psi_i(P)\} \\ \mathbf{f}_k^* = [f_k^*(1), f_k^*(2), \dots, f_k^*(P)]^T \end{cases}. \quad (9)$$

where  $n = 1, \dots, P$  is the sample number;  $w_{k,i}$  is the weight coefficient of the  $k_{th}$  active joint corresponding to the kernel function  $\psi_i$ ;  $f_k(n)$  is the forcing term of the  $k_{th}$  active joint at the  $n_{th}$  sampling point;  $\psi_i(n)$  is the  $i_{th}$  kernel function at the  $n_{th}$  sampling point;  $r(n)$  is the amplitude modulation factor at the  $n_{th}$  sampling point; and  $f_k^*(n)$  is the target forcing term of the  $k_{th}$  active joint at the  $n_{th}$  sampling point.

### 2.1.3. Desired Trajectory Predictor

Inspired by the motion superposition mechanism [24] and minimal jerk criterion [25] of human motion planning, the desired trajectory predictor based on the TDA is designed to smoothly generate and predict the desired trajectory through pHRI [19,20]. The desired trajectory predictor can be described as follows:

$$\begin{cases} \hat{\mathbf{q}}_{d,k}(u_k, t) = \hat{\mathbf{q}}_{d,k}(0, t) + u_k \mathbf{V}(t), t \in [t_s, t_f] \\ \mathbf{V}(t) = \delta \mathbf{H} \boldsymbol{\tau}_{I,k}(t_s) \\ \mathbf{H} = \frac{\mathbf{W} \boldsymbol{\beta}}{(\mathbf{p} + \delta) \|\mathbf{W}\|} \end{cases}, \quad (10)$$

$$\mathbf{W} = (\mathbf{I} - (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{C}^T (\mathbf{C} (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{C}^T)^{-1} \mathbf{C}) (\mathbf{Z}^T \mathbf{Z})^{-1}, \quad (11)$$

$$N = \frac{p}{\delta} + 1. \quad (12)$$

where  $t_s$  and  $t_f$  are the start and end times of trajectory deformation at the current physical interaction, respectively;  $u_k$  is the deformation factor of the  $k_{th}$  active joint;  $\boldsymbol{\tau}_{I,k}(t_s)$  is the interaction torque vector at time  $t_s$  for the  $k_{th}$  active joint;  $\hat{\mathbf{q}}_{d,k}(0, t)$  and  $\hat{\mathbf{q}}_{d,k}(u_k, t)$  are the discrete forms of the desired trajectory vector after the last iteration and deformation curve function vector in the current iteration, respectively;  $\boldsymbol{\beta} \in \mathbb{R}^N$  is the prediction vector of the interaction torque;  $p$  is the prediction time;  $N$  is the number of waypoints;  $\mathbf{Z} \in \mathbb{R}^{(N+3) \times N}$  is a finite differencing matrix;  $\mathbf{C} \in \mathbb{R}^{4 \times N}$  is a constraint matrix;  $\mathbf{I} \in \mathbb{R}^{N \times N}$  is an identity

matrix; and  $H$  and  $V(t)$  are the discrete forms of the motor primitive and vector field function, respectively.

#### 2.1.4. Soft Saturation Function

Due to abnormal interaction behavior, the desired trajectory may exceed the safe joint space. Thus, the soft saturation function is adopted to ensure that the desired trajectory of each joint is constrained in a predetermined safety range, enhancing rehabilitation training safety [26]. The soft saturation function will not affect the trajectory predictor in generating the desired trajectory through pHRIs when it is within the predetermined safety range; conversely, it will smoothly restrict the desired trajectory within the predetermined safety boundary. The soft saturation function for the  $k_{th}$  active joint can be described as follows:

$$q_{d,k} = \begin{cases} (1 - \zeta_k)U_k(1 - e^{(\zeta_k U_k - \hat{q}_{d,k}(u_k, t_s))/((1 - \zeta_k)U_k)}) + \zeta_k U_k, & \hat{q}_{d,k}(u_k, t_s) > \zeta_k U_k \\ \hat{q}_{d,k}(u_k, t_s), & \zeta_k D_k \leq \hat{q}_{d,k}(u_k, t_s) \leq \zeta_k U_k \\ (1 - \zeta_k)D_k(1 - e^{(\zeta_k D_k - \hat{q}_{d,k}(u_k, t_s))/((1 - \zeta_k)D_k)}) + \zeta_k D_k, & \hat{q}_{d,k}(u_k, t_s) < \zeta_k D_k \end{cases}, \quad (13)$$

where  $\hat{q}_{d,k}(u_k, t_s)$  and  $q_{d,k}$  are the desired angles at time  $t_s$  for the  $k_{th}$  active joint before and after adjustment;  $U_k$  and  $D_k$  are the upper and lower boundaries of the desired trajectory, respectively;  $\zeta_k$  ( $0 \leq \zeta_k < 1$ ) is the positive constant; and the closer the positive constant value is to 1, the closer the desired angle is to the predetermined safety boundary.

### 2.2. Low-Level Position Controller

In this section, the dynamics model of LLRRs in joint space is given and the triple-step nonlinear method [27] is adopted to design the triple-step controller. Moreover, the stability of the proposed controller is proven through the Lyapunov theorem. Each part will be explained in detail in the following subsections.

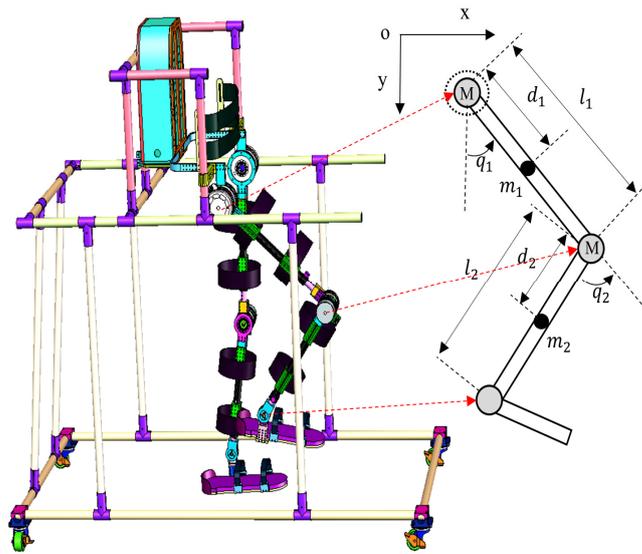
#### 2.2.1. Dynamics Model

As illustrated in Figure 2, each leg of an LLRR can be simplified into a two-link model in the sagittal plane. The dynamics model in joint space can be given as follows based on the Euler–Lagrange method [10]:

$$\hat{M}(q)\ddot{q} + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) = \tau_A + T, \quad (14)$$

$$T = \tau_I + \tau_E - f(\dot{q}) - (M(q) - \hat{M}(q))\ddot{q} - (C(q, \dot{q}) - \hat{C}(q, \dot{q}))\dot{q} - (G(q) - \hat{G}(q)). \quad (15)$$

where  $q = [q_1, q_2]^T$  is the actual angle vector;  $\dot{q}$  and  $\ddot{q}$  are the angular velocity vector and angular acceleration vector;  $M(q)$ ,  $C(q)$ , and  $G(q)$  are the inertia matrix, the centripetal and Coriolis matrix, and the gravitational vector, respectively, and can be calculated with mechanical parameters, including the mass of exoskeleton  $m_i$ , the lengths of exoskeleton  $l_i$ , the length from the center of mass to the revolute joint  $d_i$ , and the inertias of exoskeleton  $I_i$ ;  $i = 1$  and  $i = 2$  represent the thigh and calf, respectively;  $\hat{M}(q)$ ,  $\hat{C}(q)$ , and  $\hat{G}(q)$  are the nominal inertia matrix, the nominal centripetal and Coriolis matrix, and the nominal gravitational vector, respectively;  $\tau_A$  represents the control torque vector;  $f(\dot{q})$ ,  $\tau_I$ , and  $\tau_E$  are friction torque, interaction torque, and external torque vectors; and  $T$  represents total disturbances, consisting of internal and external disturbances.



**Figure 2.** The virtual prototype of an LLRR with a simplified two-linkage model of the right leg.

### 2.2.2. Triple-Step Controller

The triple-step method [27] is adopted to design the low-level position controller. This method's core concept is to divide the design process into three steps: steady-state control, feedforward control, and feedback control. The NDO and the RBFNN are integrated into the steady-state control and feedback control to enhance the accuracy and robustness of trajectory tracking.

The steady-state control is used to compensate for the effect of gravity, external disturbances, and inaccurate dynamics model, improving the steady-state performance. Firstly, a nonlinear disturbance observer, as follows, is adopted to compensate for the total disturbances [28]:

$$\begin{cases} \dot{z} = -L(q, \dot{q})z + L(q, \dot{q})(\hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) - \tau_A - p(q, \dot{q})) \\ \hat{T} = z + p(q, \dot{q}) \\ p(q, \dot{q}) = c[\dot{q}_1; \dot{q}_1 + \dot{q}_2] \\ L(q, \dot{q}) = c \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \hat{M}^{-1}(q) \\ c > m_2 l_1 d_2 \dot{q}_{2, \max} \end{cases} \quad (16)$$

where  $z$  is the auxiliary variable;  $c$  is a constant;  $m_2$  is the nominal mass of the calf exoskeleton;  $l_1$  is the nominal length of the thigh exoskeleton;  $d_2$  is the distance from the nominal center of the mass to the rotation center of the knee joint;  $\dot{q}_{2, \max}$  is the maximum angular velocity of the calf exoskeleton; and  $\hat{T}$  is the estimated value of total disturbances.

Thus, Equation (14) can be rewritten as follows:

$$\begin{cases} \hat{M}(q)\ddot{q} + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) = \tau_A + \hat{T} - \Delta T \\ \Delta T = \hat{T} - T \end{cases}, \quad (17)$$

where  $\Delta T$  is the estimated error.

By assigning zero to  $\dot{q}$ ,  $\ddot{q}$ , and  $\Delta T$  and replacing  $\tau_A$  with  $\tau_{SSC}$  in Equation (17), we can obtain the steady-state controller as follows:

$$\tau_{SSC} = \hat{G}(q) - \hat{T}. \quad (18)$$

Feedforward control is adopted to improve the response speed due to the nonlinear and time-varying characteristics of the dynamics model. By defining  $\ddot{q} = \ddot{q}_d$ ,  $\dot{q} = \dot{q}_d$ , and

$\tau_A = \tau_{ssc} + \tau_{ffc}$  and assigning zero to  $\Delta T$  in Equation (17), we can obtain the feedforward controller as follows:

$$\tau_{ffc} = \hat{M}(q)\ddot{q}_d + \hat{C}(q, \dot{q}_d)\dot{q}_d, \quad (19)$$

where  $\dot{q}_d$  and  $\ddot{q}_d$  are the derivative and the second derivative of the desired angle  $q_d$ .

The feedback control input is adopted to improve the accuracy and robustness of the controller. Equation (14) can then be rewritten as follows:

$$\ddot{q} = \hat{M}(q)^{-1}(\tau_A + T) - \hat{M}(q)^{-1}\hat{C}(q, \dot{q})\dot{q} - \hat{M}(q)^{-1}\hat{G}(q). \quad (20)$$

By letting  $\tau_A = \tau_{ssc} + \tau_{ffc} + \tau_{fbc}$  and defining the tracking error  $e = q_d - q$ , we can obtain:

$$\ddot{e} = \hat{M}(q)^{-1}\hat{C}(q, \dot{q})\dot{q} - \hat{M}(q)^{-1}\hat{C}(q, \dot{q}_d)\dot{q}_d - \hat{M}(q)^{-1}\tau_{fbc} + \hat{M}(q)^{-1}(\Delta T). \quad (21)$$

Thus, a second-order error auxiliary system can be defined as follows:

$$\ddot{e} = -k_p e - k_d \dot{e} + \Delta H - c_{er}, \quad (22)$$

where  $k_d$  and  $k_p$  are the control gain matrices and  $c_{er}$  is a compensation variable for observation errors after decoupling  $\Delta H = \hat{M}(q)^{-1}\Delta T$ .

Substituting Equation (22) into Equation (21), we can obtain a feedback controller as follows:

$$\tau_{fbc} = \hat{M}(q)(k_p e + k_d \dot{e} + c_{er}) + \hat{C}(q, \dot{q})\dot{q} - \hat{C}(q, \dot{q}_d)\dot{q}_d. \quad (23)$$

Since  $\Delta H$  can be efficiently approximated by the RBFNN, Equation (22) can be rewritten as follows:

$$\begin{cases} \ddot{e} + k_d \dot{e} + k_p e = \Delta H - c_{er} \\ c_{er} = \Delta \hat{H} + \Gamma \\ \Delta H = \Delta \hat{H}(W^*) = W^{*T} h(x) + \sigma \\ \Delta \hat{H}(\hat{W}) = \hat{W}^T h(x) \\ h_j(x) = \exp(-\pi \|x - c_j\|^2 / b_j^2), j = 1, 2, \dots, n \end{cases}. \quad (24)$$

where  $\Delta \hat{H} = [\Delta \hat{H}_1, \Delta \hat{H}_2]^T$  is the estimated value of  $\Delta H$ ;  $\Gamma = [\Gamma_1, \Gamma_2]^T$  is the robust term to be designed;  $\hat{W}$  is a weight matrix to be tuned;  $\sigma = [\sigma_1, \sigma_2]^T$  is the approximation error vector of the RBFNN when taking the optimal weight matrix;  $W^*$  is the optimal weight matrix;  $h(x)$  is the hidden layer basis vector of the RBFNN;  $h_j(x)$  is the Gaussian function of the  $j_{th}$  neuron in the hidden layer of the RBFNN;  $n$  is the number of neurons in the hidden layer; and  $c_j$  and  $b_j$  are the center vector and width of the  $j_{th}$  Gaussian function.

For the  $k_{th}$  active joint, defining  $E_k = [e_k, \dot{e}_k]^T$ , Equation (24) can be rewritten as follows:

$$\begin{cases} \dot{E}_k = \Lambda_k E_k + B_k N_k \\ N_k = (W_k^* - \hat{W}_k)^T h_k(x_k) + \sigma_k - \Gamma_k \end{cases}, \quad (25)$$

where  $\Lambda_k = \begin{bmatrix} 0 & 1 \\ -k_{p,k} & -k_{d,k} \end{bmatrix}$  and  $B_k = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ .

For a scalar  $\Delta H_k$ , the RBFNN approximation law and robust term are designed as follows:

$$\begin{cases} \Delta H_k = \hat{W}_k^T h_k(x_k) \\ \dot{\hat{W}}_k = \gamma_k E_k^T P_k B_k h_k(x_k) \end{cases}, \quad (26)$$

$$\Gamma_k = \eta_k \text{sat}(E_k^T P_k B_k), \quad (27)$$

$$\text{sat}(E_k^T P_k B_k) = \begin{cases} 1, E_k^T P_k B_k > \Delta \\ \frac{1}{\Delta} E_k^T P_k B_k, |E_k^T P_k B_k| \leq \Delta, \\ -1, E_k^T P_k B_k < -\Delta \end{cases} \quad (28)$$

$$\Lambda_k P_k + P_k \Lambda_k = -Q_k, \quad (29)$$

where  $\gamma_k$  is a positive constant;  $P_k$  is the solution of the Lyapunov equation;  $\eta_k$  is a positive constant;  $\Delta$  is the boundary layer thickness; and  $Q_k$  is a positive constant matrix.

Thus, Equation (23) can be rewritten as follows:

$$\tau_{fbc} = \hat{M}(q)(k_p e + k_d \dot{e} + \Gamma + \hat{W}^T h(x)) + \hat{C}(q, \dot{q})\dot{q} - \hat{C}(q, \dot{q}_d)\dot{q}_d. \quad (30)$$

Finally, according to Equations (18), (19), and (30), the triple-step controller can be obtained as follows:

$$\tau_A = \tau_{ssc} + \tau_{ffc} + \tau_{fbc} = \hat{M}(q)(\ddot{q}_r + k_p e + k_d \dot{e} + \Gamma + \hat{W}^T h(x)) + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) - \hat{T}. \quad (31)$$

### 2.2.3. Proof of Stability

The stability of the triple-step controller can be proven as follows using the Lyapunov theorem.

Substituting Equation (31) into Equation (14), we can obtain a second-order error auxiliary system as follows:

$$\ddot{e} + k_p e + k_d \dot{e} = \Delta H - \Gamma - \hat{W}^T h(x). \quad (32)$$

The closed-loop stability of the triple-step controller is transformed into the stability of the second-order error system.

If the RBFNN can effectively compensate for the observation errors of each joint after decoupling, i.e.,  $\Delta H - \Gamma - W^{*T} h(x) = \sigma$  ( $\sigma$  is close to zero), the control gain matrices  $k_d$  and  $k_p$  can be determined using the pole placement method.

For the  $k_{th}$  active joint, the control gain  $k_{p,k}$  and  $k_{d,k}$  are determined using:

$$(s + \omega_k)^2 = 0, \quad (33)$$

where  $s$  is the Laplace variable and  $\omega_k$  is the pole set by the user. Thus, the control gain can be set as follows:

$$k_{p,k} = \omega_k^2, k_{d,k} = 2\omega_k. \quad (34)$$

However, it is difficult to quickly converge the weight matrix to its optimal value. Moreover, the number of nodes in the RBFNN cannot be set too high to ensure computational efficiency. Therefore, compensation errors often exist, causing instability and a reduction in the robustness of the HRCCS.

If we obtain the control gain matrices using the pole placement method, the second-order error auxiliary system is a linear constant continuous system. Thus, the Lyapunov function for the  $k_{th}$  active joint can be defined as follows to guarantee closed-loop stability:

$$V = \frac{1}{2} E_k^T P_k E_k + \frac{1}{2\gamma} (W_k^* - \hat{W}_k)^T (W_k^* - \hat{W}_k). \quad (35)$$

The derivation of  $V$  can be given as follows:

$$\begin{aligned} \dot{V} &= \frac{1}{2} \dot{E}_k^T P_k E_k + \frac{1}{2} E_k^T P_k \dot{E}_k - \frac{1}{\gamma} (W_k^* - \hat{W}_k)^T \dot{\hat{W}}_k \\ &= \frac{1}{2} (\Lambda_k E_k + B_k N_k)^T P_k E_k + \frac{1}{2} E_k^T P_k (\Lambda_k E_k + B_k N_k) - \frac{1}{\gamma} (W_k^* - \hat{W}_k)^T \dot{\hat{W}}_k \\ &= \frac{1}{2} E_k^T (\Lambda_k^T P_k + P_k \Lambda_k) E_k + E_k^T P_k B_k N_k - \frac{1}{\gamma} (W_k^* - \hat{W}_k)^T \dot{\hat{W}}_k \end{aligned} \quad (36)$$

According to Equations (25) and (29), we can obtain:

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{E}_k^T\mathbf{Q}\mathbf{E}_k + \mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k((\mathbf{W}_k^* - \hat{\mathbf{W}}_k)^T\mathbf{h}(x_k) + \sigma_k - \Gamma_k) - \frac{1}{\gamma}(\mathbf{W}_k^* - \hat{\mathbf{W}}_k)^T\dot{\hat{\mathbf{W}}}_k \\ &= -\frac{1}{2}\mathbf{E}_k^T\mathbf{Q}\mathbf{E}_k + \mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k(\sigma_k - \Gamma_k) + (\mathbf{W}_k^* - \hat{\mathbf{W}}_k)^T(\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\mathbf{h}(x_k) - \frac{1}{\gamma}\dot{\hat{\mathbf{W}}}_k)\end{aligned}\quad (37)$$

Substituting Equations (26) and (27) into Equation (37), we can obtain:

$$\begin{aligned}\dot{V} &= -\frac{1}{2}\mathbf{E}_k^T\mathbf{Q}\mathbf{E}_k + \mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k(\sigma_k - \eta_k\text{sat}(\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k)) \\ &\quad + (\mathbf{W}_k^* - \hat{\mathbf{W}}_k)^T(\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\mathbf{h}(x_k) - \frac{1}{\gamma}\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\mathbf{h}(x_k)) \\ &= -\frac{1}{2}\mathbf{E}_k^T\mathbf{Q}\mathbf{E}_k + \mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\sigma_k - \eta_k\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\text{sat}(\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k) \\ &= -\frac{1}{2}\mathbf{E}_k^T\mathbf{Q}\mathbf{E}_k + \mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k\sigma_k - \eta_k|\mathbf{E}_k^T\mathbf{P}_k\mathbf{B}_k|\end{aligned}\quad (38)$$

Since  $\sigma_k$  is close to zero, it is easy to choose  $\eta_k$ , which can maintain  $\sigma_k \leq \eta_k$ . Thus, it can ensure  $\dot{V} \leq 0$  and asymptotic stability of the closed-loop system can be guaranteed.

### 3. Simulation and Results

#### 3.1. Simulation Setup

Three types of simulation experiments were carried out in MATLAB (2014a, MathWorks) to verify the superiority of the HRCCS. The reference angle of each joint was fitted from one healthy subject's gait data, and the sample time was set to 0.01 s.

##### 1. Robustness verification of the HRCCS

Three types of external torque were applied to the system, i.e., (A)  $\boldsymbol{\tau}_E = [0; 0]$ ; (B)  $\boldsymbol{\tau}_E = [15 \cos(0.4\pi t); 15 \sin(0.4\pi t)]$ ; and (C)  $\boldsymbol{\tau}_E = [30 \cos(0.4\pi t); 30 \sin(0.4\pi t)]$ . The interaction torque and deformation factor were set to  $\boldsymbol{\tau}_I = [0; 0]$  and  $\mathbf{u} = [0; 0]$ . Moreover, the control parameters remained unchanged under different external torques during the simulations.

##### 2. Compliant interaction verification of the HRCCS

The deformation factor of each joint was set as follows:

$$u_1(t) = u_2(t) = \begin{cases} 0.02, & 5 \leq t \leq 15 \\ 0.04, & 20 \leq t \leq 25 \\ 0, & \text{others} \end{cases} \quad (39)$$

Similarly, to [20], the interaction torque of each joint was set as follows to simulate the physical interaction process, including the contact transition stage. Moreover, the external torque was set to  $\boldsymbol{\tau}_E = [0; 0]$ .

$$\tau_{I,1}(t) = \begin{cases} 5, & 5 \leq t \leq 15 \\ -5, & 20 \leq t \leq 25 \\ 0, & \text{others} \end{cases} \quad (40)$$

$$\tau_{I,2}(t) = \begin{cases} -5, & 5 \leq t \leq 15 \\ 5, & 20 \leq t \leq 25 \\ 0, & \text{others} \end{cases} \quad (41)$$

##### 3. Trajectory interactive learning verification of the HRCCS

The deformation factor and external torque were set to  $\mathbf{u} = [0.04; 0.04]$  and  $\boldsymbol{\tau}_E = [0; 0]$ . The interaction torques were assumed to be periodic, and they can be given as follows:

$$\tau_{I,1}(t) = \begin{cases} 5 \sin(0.2\pi t), & 20 \leq t \leq 30 \\ -5 \sin(0.2\pi t), & 50 \leq t \leq 60 \\ 0, & \text{others} \end{cases} \quad (42)$$

$$\tau_{1,2}(t) = \begin{cases} -5 \sin(0.2\pi t), & 20 \leq t \leq 30 \\ 5 \sin(0.2\pi t), & 50 \leq t \leq 60 \\ 0, & \text{others} \end{cases} \quad (43)$$

### 3.2. Parameter Settings

The actual structural parameters were given as shown in Table 1. The nominal physical parameters were set as  $\hat{m}_i = 1.2m_i$ ,  $\hat{l}_i = 1.2l_i$ ,  $\hat{I}_i = I_i$ , and  $\hat{d}_i = d_i$ .

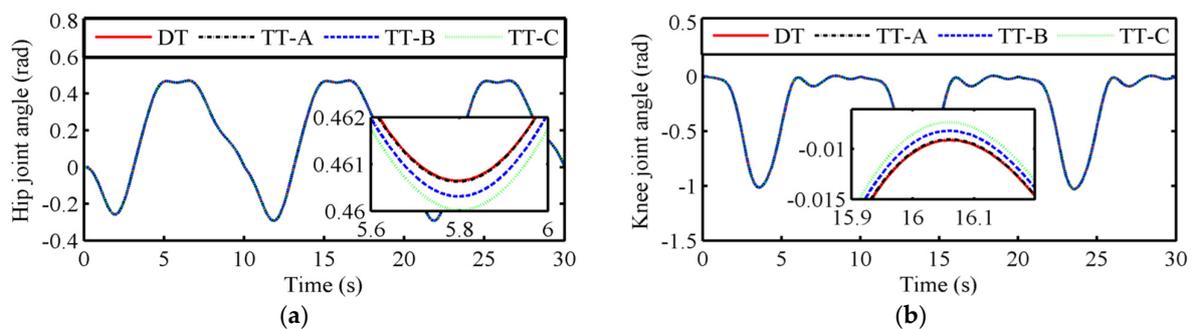
**Table 1.** Structural parameters of LLRR.

Symbol	Thigh ( $i = 1$ )	Calf ( $i = 2$ )
$m_i$ (kg)	2.582	3.192
$l_i$ (m)	0.390	0.464
$d_i$ (m)	0.328	0.355
$I_i$ (kg m <sup>2</sup> )	0.307	0.446

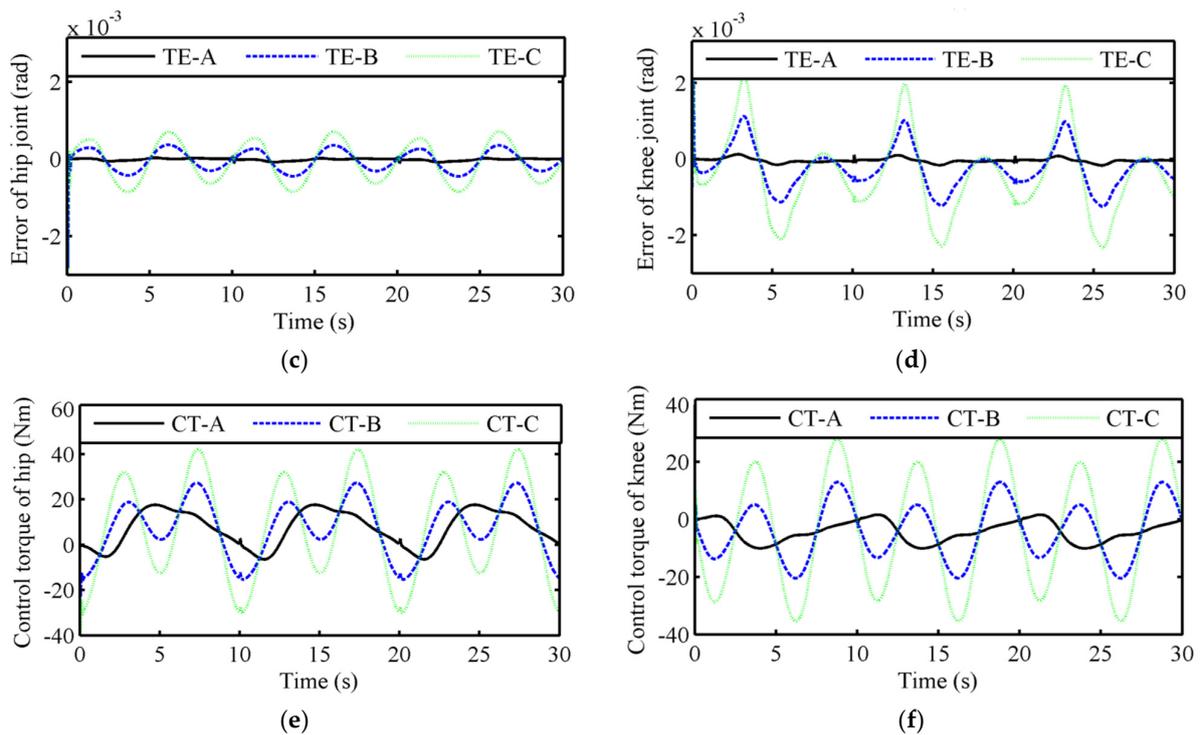
The positive constants in the trajectory generator were set as follows:  $\alpha_q = 25$ ,  $\beta_q = 6.25$ ,  $\alpha_r = 12.5$ , and  $g = 0$ . Moreover, the temporal scaling factor and the initial value of the amplitude modulation factor were set to  $\kappa = 1$  and  $r_0 = 1$ . For the trajectory predictor, the prediction time and the sample period were set to  $p = 1$  s and  $\delta = 0.01$  s. In the soft saturation function, the upper and lower boundaries of the desired trajectory were set to  $U_1 = 0.6$  rad,  $U_2 = 0.1$  rad,  $D_1 = -0.6$  rad, and  $D_2 = -1.2$  rad, and the positive constant of each joint was set to  $\zeta_1 = \zeta_2 = 0.9$ . For the RBFNN, the input was set to  $x_i = [e_i, \dot{e}_i, q_{d,i}, \dot{q}_{d,i}, \ddot{q}_{d,i}]^T$ , and the number of neurons was chosen as  $n = 5$ . The center of the RBFNN neurons were evenly designed between the lower and upper bounds of each input parameter, and we set their lower and upper bounds to  $-0.15$ – $0.15$ ,  $-0.15$ – $0.15$ ,  $-1.5$ – $1.5$ ,  $-1.5$ – $1.5$ , and  $-3.0$ – $3.0$ . According to [29], the width vector of the Gaussian function can be obtained as  $b = [5.4, 5.4, 1.7, 1.7, 1.2]^T$ . The other parameters of the triple-step controller were set as follows:  $Q_1 = Q_2 = \text{diag}\{85, 85\}$ ,  $\Delta = 1$ ,  $\eta_1 = \eta_2 = 1$ ,  $k_d = \text{diag}\{20, 20\}$ ,  $k_p = \text{diag}\{100, 100\}$ ,  $c = 50$ , and  $\gamma_1 = \gamma_2 = 400$ .

### 3.3. Simulation Results

From Figure 3a,b, the trajectory generator can continuously generate a reference trajectory for each joint and the triple-step controller can ensure that each joint follows the reference trajectory well. Figure 3c,d show that the tracking error of the hip and knee joints increases significantly with the external torque. At the same time, the tracking error is still within 0.001 rad even when the external torque reaches 30 Nm. Additionally, as shown in Figure 3e,f, the control torque exhibits periodic characteristics and shows no oscillations and apparent mutations.

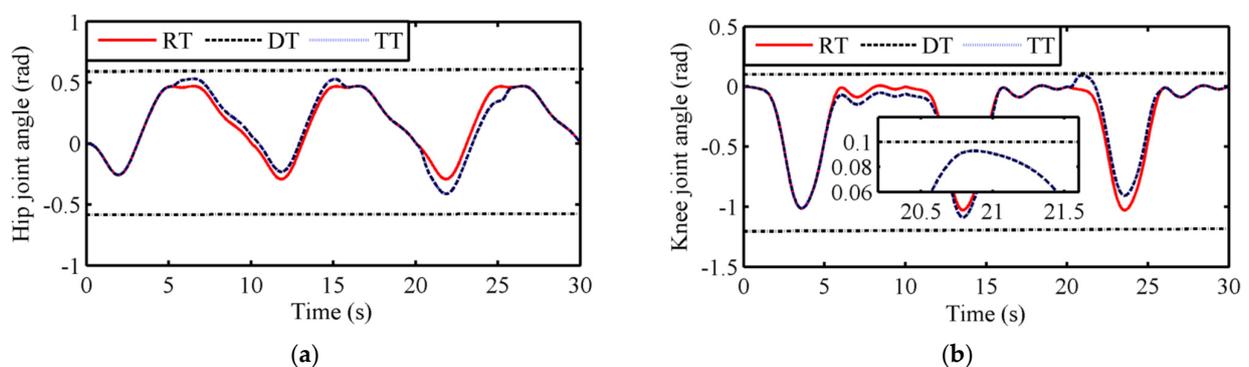


**Figure 3.** Cont.

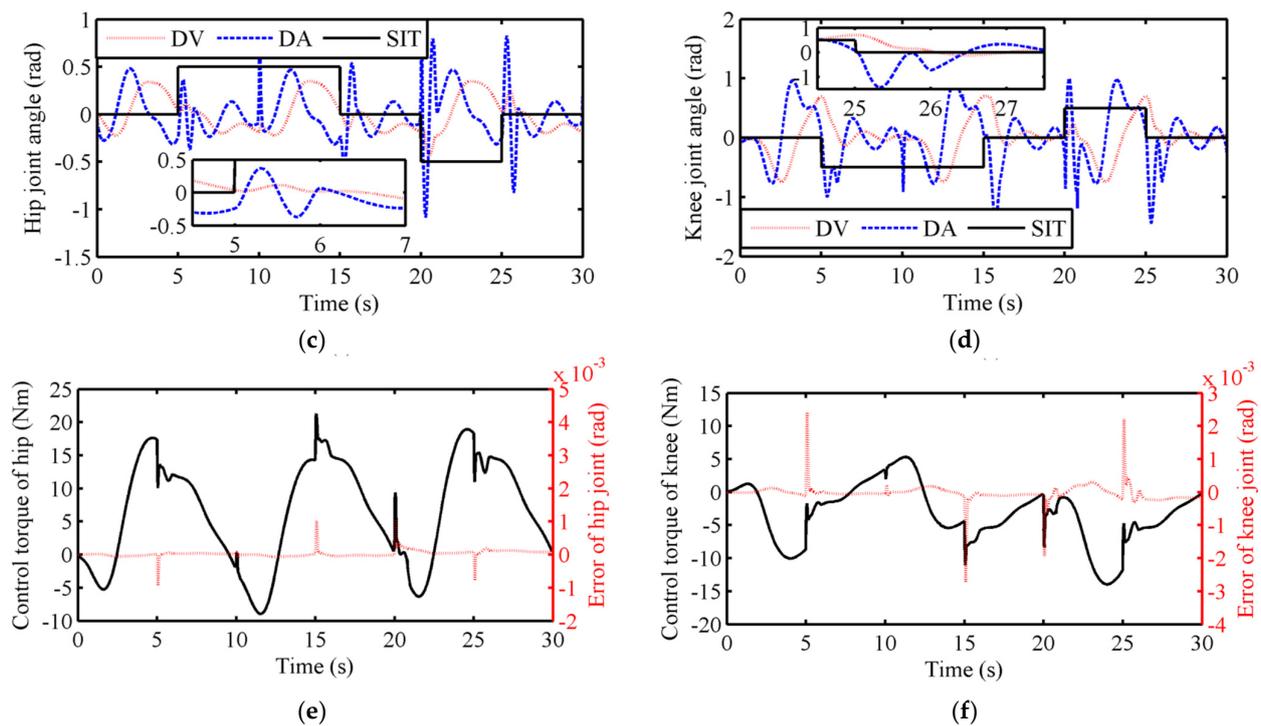


**Figure 3.** Robustness verification results of the HRCCS: (a) Hip joint angle; (b) Knee joint angle; (c) Tracking error of hip joint; (d) Tracking error of knee joint; (e) Control torque of hip joint; (f) Control torque of knee joint. RT, TT, TE, and CT represent reference trajectory, tracking trajectory, tracking error, and control torque, respectively. A, B, and C represent three types of external torques.

From Figure 4a–d, the trajectory predictor can continuously generate the desired trajectory when a pHRI occurs, and the desired trajectory gradually converges back to the reference trajectory once the pHRI stops. Although the magnitude of the interaction torque between 5–15 s and 20–25 s is the same, the trajectory deviation between the reference trajectory and the desired trajectory significantly increases because the deformation factor of the latter is twice that of the former. As shown in the partially enlarged image in Figure 4b, the soft saturation function can smoothly restrict the desired trajectory, ensuring that it is within the predetermined safety range. Furthermore, the desired angular acceleration trajectory, desired angular velocity trajectory, and desired trajectory are smooth during the transitioning stage (from non-contact to contact or contact to non-contact). As shown in Figure 4e,f, the control torque and tracking error of each joint suddenly change during the transition phase.

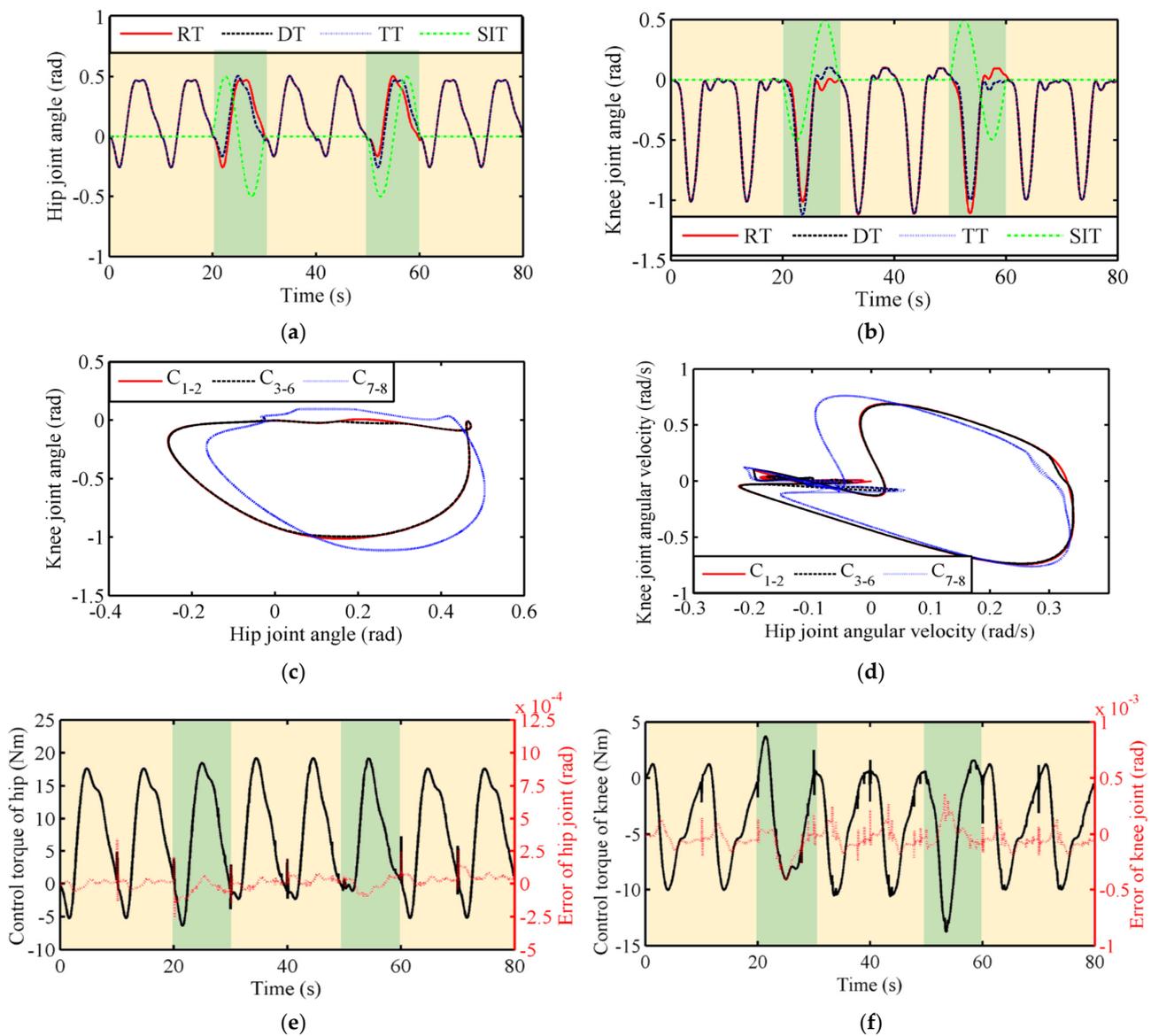


**Figure 4.** Cont.



**Figure 4.** Compliant interaction verification results of the HRCCS: (a) Hip joint angle; (b) Knee joint angle; (c) Desired angle of hip joint; (d) Desired angle of knee joint; (e) Control torque and tracking error of hip joint; (f) Control torque and tracking error of knee joint. RT, DT, and TT represent reference trajectory, desired trajectory, and tracking trajectory, respectively. DAV, DAA, and SIT represent desired angular velocity trajectory, desired angular acceleration trajectory, and scaled interaction torque, respectively.

From Figure 5a,b, the LLRR accurately tracked the reference trajectory when no interaction torque was applied during the first to second cycles. In the third cycle, the trajectory predictor smoothly generated the desired trajectory, and the LLRR tracked the desired trajectory well under sinusoidal interaction torque. In the fourth and fifth cycles, the trajectory learner learned the desired trajectory of the previous cycle and replaced the initial reference trajectory. In the sixth cycle, a sinusoidal interaction torque of the same magnitude and opposite direction was applied, and the trajectory learner smoothly generated the desired trajectory again. In the seventh and eighth cycles, the trajectory learner learned the desired trajectory from the previous cycle again and served as a new reference trajectory. As shown in Figure 5c,d, there is a significant deviation in the joint space for the tracking trajectories of the first to second and third to fifth cycles. However, the tracking trajectories of the sixth to seventh cycles coincide with those of the first to second cycles. From Figure 5e,f, the tracking error of each joint trajectory was less than 0.0005 rad during trajectory learning. Moreover, the amplitude of the control torque also underwent significant changes due to the trajectory interactive learning.



**Figure 5.** Trajectory interactive learning verification results of the HRCCS: (a) Hip joint angle; (b) Knee joint angle; (c) Hip and knee joint angle in joint space; (d) Hip and knee joint angular velocity in joint space; (e) Control torque and tracking error of hip joint; (f) Control torque and tracking error of knee joint. RT, DT, TT, and SIT represent reference trajectory, desired trajectory, tracking trajectory, and scaled interaction torque, respectively. GC1–2, GC3–6, and GC7–8 represent gait cycles from first to second, third to sixth, and seventh to eighth, respectively.

#### 4. Discussion

Compliant physical interactions, trajectory interactive learning, and robust position control are crucial for rehabilitation robots. This article effectively integrates the TDA, DMPs, and a triple-step controller into the HRCCS, including a high-level trajectory planner and a low-level position controller. Compared with the admittance control in [18], the high-level trajectory planner can predict and generate the desired trajectory based on the TDA in a restricted joint space and is more effective in increasing trajectory smoothness and robot compliance [20]. Similarly, to [30], the high-level trajectory planner can learn the desired trajectory based on the DMP using the locally weighted regression method. Moreover, the triple-step controller is designed to be a low-level position controller to ensure that each joint tracks the reference/desired trajectory. Compared with [10], disturbance observers are

used to observe and compensate for total disturbances and the RBFNN with the robust term is used to compensate for observation errors. Furthermore, the stability of the proposed controller is guaranteed through the Lyapunov theorem.

Since patients have various levels of impairment, physical therapists can adjust the control parameters of the HRCCS to meet their different rehabilitation training needs. For patients with weak residual muscle strength, the passive training mode can be used by setting the deformation factor to a small value [20]. It can achieve periodic, repetitive motion control for the affected limb under an unknown dynamic contact environment caused by the various symptoms of stroke survivors. Similarly, to the impedance parameter adjustments in [31], increasing the deformation factor can realize the switch from the passive to active training modes. In active training mode, rehabilitation robots can predict and generate smooth desired trajectories through physical interaction, allowing participants to actively participate in rehabilitation training. For patients with a high willingness and ability to actively participate in rehabilitation training, the trajectory predictor and trajectory learner of the HRCCS should be activated simultaneously. This can enable rehabilitation robots to autonomously learn the patient's movement trajectory to handle time-varying interaction dynamics and update varying walking patterns [23]. Compared with the impedance control strategy, which can provide different resistance levels for users and achieve various rehabilitation training modes [32], the HRCCS is not dependent on a high-frequency inner-loop force controller and an accurate dynamic model. Although the admittance control strategy can also solve the above problems while achieving whole-stage compliance rehabilitation training [33], it cannot learn and generalize task trajectories online. Moreover, achieving the desired trajectory prediction and ideal physical interaction performance through parameter adjustment is challenging for admittance control. Similarly, to [34], the position-constrained assist-as-needed control strategy has been proposed to establish a continuous transition between the human-dominated and robot-dominated modes to ensure seamless and safe robotic assistance [12]. However, it is challenging to enhance movement smoothness, which is a valuable performance indicator of stroke recovery [35], within the human-dominated mode due to the various symptoms of stroke survivors [36], especially when the dead zone width is set at a considerable value. Although the HRCCS cannot switch between multiple training modes, it focuses on ensuring interaction and control performance by introducing the TDA and designing robust position control [20]. Additionally, it can ensure that the desired trajectory of each joint is constrained in a predetermined safety range, which is beneficial for improving the safety of rehabilitation training [26,37].

However, this article has several limitations that should be addressed. Firstly, while the proposed control strategy can be applied in rehabilitation robots for passive, active, and interactive learning control, it has not yet been tested on LLRRs. Secondly, the proposed control strategy cannot achieve adaptive switching of multiple control modes, which would increase the operational complexity for physical therapists. In future research, we will use artificial neural networks [38,39] and neuromuscular skeletal models [40] to evaluate users' motion intention, and position errors, interaction forces, and movement smoothness will be used to evaluate users' movement performance [12,20]. Similarly, to [41], we will design and verify the multi-mode adaptive compliance strategy based on motion intention or movement performance in LLRRs. This will enable smooth switching between the passive training mode, active training mode, interactive learning training mode, and soft-stop mode, allowing LLRRs to match the varying motor abilities of patients online.

## 5. Conclusions

This article proposed an HRCCS for lower limb rehabilitation robots. The HRCCS, comprising a high-level trajectory planner designed using the TDA, DMPs, and the soft saturation function, can generate reference trajectories, predict smooth desired trajectories through pHRI, and learn the desired trajectory. A triple-step controller with NDO and RBFNN was designed to be the low-level position controller to ensure trajectory tracking

accuracy and robustness. The proposed strategy was tested through three types of simulation experiments: robustness verification, compliant interaction, and trajectory interactive learning. The results confirmed the effectiveness of the HRCCS in ensuring trajectory tracking accuracy and robustness, as well as its ability to achieve compliant pHRIs and trajectory interactive learning, and demonstrated the potential of the HRCCS in passive, active, and interactive learning control for rehabilitation robots and other fields involving pHRI.

**Author Contributions:** Conceptualization, J.Z. and Z.W.; methodology, J.Z.; software, J.Z. and Y.S.; validation, J.Z., Y.S., L.L. and W.Z.; formal analysis, J.Z.; investigation, J.Z., Y.S., L.L. and W.Z.; resources, J.Z., Y.S., L.L. and W.Z.; data curation, J.Z. and Z.W.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z. and Y.S.; visualization, J.Z. and Y.S.; supervision, J.Z. and Y.S.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Key Science and Technology Program of Liaoning Province under grant 2022020630-JH1/108; the National Natural Science Foundation of China under grant 51975386; the Science and Technology Research and Development Program of China National Railway Group Co., Ltd. under grant N2022J014; and the Natural Science Foundation of Guangdong Province under grant 2023A1515011021.

**Data Availability Statement:** All data used to support the findings of this study are included within this article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Mukherjee, D.; Patil, C.G. Epidemiology and the global burden of stroke. *World Neurosurg.* **2011**, *76*, S85–S90. [[CrossRef](#)] [[PubMed](#)]
- Ochi, M.; Wada, F.; Saeki, S.; Hachisuka, K. Gait training in subacute non-ambulatory stroke patients using a full weight-bearing gait-assistance robot: A prospective, randomized, open, blinded-endpoint trial. *J. Neurol. Sci.* **2015**, *353*, 130–136. [[CrossRef](#)]
- Rodríguez-Fernández, A.; Lobo-Prat, J.; Font-Llagunes, J.M. Systematic review on wearable lower-limb exoskeletons for gait training in neuromuscular impairments. *J. Neuroeng. Rehabil.* **2021**, *18*, 22. [[CrossRef](#)] [[PubMed](#)]
- Moon, S.B.; Ji, Y.H.; Jang, H.Y.; Hwang, S.H.; Shin, D.B.; Lee, S.C.; Han, J.S.; Han, C.S.; Lee, Y.G.; Jang, S.H.; et al. Gait analysis of hemiplegic patients in ambulatory rehabilitation training using a wearable lower-limb robot: A pilot study. *Int. J. Precis. Eng. Manuf.* **2017**, *18*, 1773–1781. [[CrossRef](#)]
- Yang, M.; Wang, X.; Zhu, Z.; Xi, R.; Wu, Q. Development and control of a robotic lower limb exoskeleton for paraplegic patients. *Proc. Inst. Mech. Eng. Part C J. Mechan.* **2019**, *233*, 1087–1098. [[CrossRef](#)]
- Dong, M.; Yuan, J.; Li, J. A lower limb rehabilitation robot with rigid-flexible characteristics and multi-mode exercises. *Machines* **2022**, *10*, 918. [[CrossRef](#)]
- Zhou, J.; Yang, R.; Lyu, Y.; Song, R. Admittance control strategy with output joint space constraints for a lower limb rehabilitation robot. In Proceedings of the International Conference on Advanced Robotics and Mechatronics, Shenzhen, China, 18–21 December 2020. [[CrossRef](#)]
- Han, S.; Wang, H.; Tian, Y.; Christov, N. Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton. *ISA Trans.* **2020**, *97*, 171–181. [[CrossRef](#)] [[PubMed](#)]
- Chen, Z.; Guo, Q.; Li, T.; Yan, Y. Output constrained control of lower limb exoskeleton based on knee motion probabilistic model with finite-time extended state observer. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 2305–2316. [[CrossRef](#)]
- Peng, H.; Zhou, J.; Song, R. A triple-step controller with linear active disturbance rejection control for a lower limb rehabilitation robot. *Front. Neurobot.* **2022**, *16*, 1053360. [[CrossRef](#)] [[PubMed](#)]
- Asl, H.J.; Narikiyo, T.; Kawanishi, M. Adaptive neural network-based saturated control of roboti exoskeletons. *Nonlinear Dynam.* **2018**, *94*, 123–139. [[CrossRef](#)]
- Cao, Y.; Chen, X.; Zhang, M.; Huang, J. Adaptive position constrained assist-as-needed control for rehabilitation robots. *IEEE Trans. Ind. Electron.* **2023**, *71*, 4059–4068. [[CrossRef](#)]
- Camardella, C.; Porcini, F.; Filippeschi, A.; Marcheschi, S.; Solazzi, M.; Frisoli, A. Gait phases blended control for enhancing transparency on lower-limb exoskeletons. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5453–5460. [[CrossRef](#)]
- Wicke, A.D.; Zitzewitz, J.V.; Caprez, A.; Lunenburger, L.; Riener, R. Path control: A method for patient-cooperative robot-aided gait rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2010**, *18*, 38–48. [[CrossRef](#)] [[PubMed](#)]
- Li, Z.; Ren, Z.; Zhao, K.; Deng, C.; Feng, Y. Human-cooperative control design of a walking exoskeleton for body weight support. *IEEE Trans. Ind. Electron.* **2020**, *16*, 2985–2996. [[CrossRef](#)]
- Martínez, A.; Lawson, B.; Goldfarb, M. A controller for guiding leg movement during overground walking with a lower limb exoskeleton. *IEEE Trans. Robot.* **2018**, *34*, 183–193. [[CrossRef](#)]

17. Shen, Z.; Zhuang, Y.; Zhou, J.; Gao, J.; Song, R. Design and test of admittance control with inner adaptive robust position control for a lower limb rehabilitation robot. *Int. J. Control Autom.* **2020**, *18*, 134–142. [[CrossRef](#)]
18. Huang, P.; Li, Z.; Zhou, M.; Li, X.; Cheng, M. Fuzzy enhanced adaptive admittance control of a wearable walking exoskeleton with step trajectory shaping. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 1541–1552. [[CrossRef](#)]
19. Losey, D.P.; O'Malley, M.K. Trajectory deformations from physical human-robot interaction. *IEEE Trans. Robot.* **2018**, *34*, 126–138. [[CrossRef](#)]
20. Zhou, J.; Li, Z.; Li, X.; Wang, X.; Song, R. Human–robot cooperation control based on trajectory deformation algorithm for a lower limb rehabilitation robot. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 3128–3138. [[CrossRef](#)]
21. Ijspeert, A.J.; Nakanishi, J.; Hoffmann, H.; Pastor, P.; Schaal, S. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Comput.* **2013**, *25*, 328–373. [[CrossRef](#)] [[PubMed](#)]
22. Zhou, J.; Peng, H.; Su, S.; Song, R. Spatiotemporal compliance control for a wearable lower limb rehabilitation robot. *IEEE Trans. Biomed. Eng.* **2023**, *70*, 1858–1868. [[CrossRef](#)] [[PubMed](#)]
23. Huang, R.; Cheng, H.; Guo, H.; Lin, X.; Zhang, J. Hierarchical learning control with physical human-exoskeleton interaction. *Inform. Sci.* **2018**, *432*, 584–595. [[CrossRef](#)]
24. Mussa-Ivaldi, F.A. Modular features of motor control and learning. *Curr. Opin. Neurobiol.* **1999**, *9*, 713–717. [[CrossRef](#)] [[PubMed](#)]
25. Corteville, B.; Aertbeliën, E.; Bruyninckx, H.J.; Schutter, D.; Brussel, H.V. Human-inspired robot assistant for fast point-to-point movements. In Proceedings of the International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007. [[CrossRef](#)]
26. He, W.; Huang, H.; Ge, S.S. Adaptive neural network control of a robotic manipulator with time-varying output constraints. *IEEE Trans. Cyber.* **2017**, *47*, 3136–3147. [[CrossRef](#)] [[PubMed](#)]
27. Gao, B.; Chen, H.; Liu, Q.; Chu, H. Position control of electric clutch actuator using a triple-step nonlinear method. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6995–7003. [[CrossRef](#)]
28. Chen, W.; Ballance, D.J.; Gawthrop, P.J.; O'Reilly, J. A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **2000**, *47*, 932–938. [[CrossRef](#)]
29. Daachi, M.E.; Madani, T.; Daachi, B.; Djouani, K. A radial basis function neural network adaptive controller to drive a powered lower limb knee joint orthosis. *Appl. Soft Comput.* **2015**, *34*, 324–336. [[CrossRef](#)]
30. Qiu, S.; Guo, W.; Caldwell, D.; Chen, F. Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *13*, 67–79. [[CrossRef](#)]
31. Jamwal, P.K.; Hussain, S.; Ghayesh, M.H.; Rogozina, S.V. Impedance control of an intrinsically compliant parallel ankle rehabilitation robot. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3638–3647. [[CrossRef](#)]
32. Akdoğan, E.; Adli, M.A. The design and control of a therapeutic exercise robot for lower limb rehabilitation: Physiotherobot. *Mechatronics* **2011**, *21*, 509–522. [[CrossRef](#)]
33. Dong, M.; Fan, W.; Li, J.; Zhou, X.; Rong, X.; Kong, Y.; Zhou, Y. A new ankle robotic system enabling whole-stage compliance rehabilitation training. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 1490–1500. [[CrossRef](#)]
34. Li, X.; Pan, Y.; Chen, G.; Yu, H. Multi-modal control scheme for rehabilitation robotic exoskeletons. *Int. J. Robot. Res.* **2017**, *36*, 759–777. [[CrossRef](#)]
35. Rohrer, B.; Fasoli, S.; Krebs, H.I.; Hughes, R.; Volpe, B.; Frontera, W.R.; Stein, J.; Hogan, N. Movement smoothness changes during stroke recovery. *J. Neurosci.* **2022**, *22*, 8297–8304. [[CrossRef](#)] [[PubMed](#)]
36. Park, J. Movement disorders following cerebrovascular lesion in the basal ganglia circuit. *J. Mov. Disord.* **2016**, *9*, 71–79. [[CrossRef](#)] [[PubMed](#)]
37. Zhou, J.; Sun, Y.; Song, R.; Wei, Z. Dynamic movement primitives modulation-based compliance control for a new sitting/lying lower limb rehabilitation robot. *IEEE Access* **2024**, *12*, 44125–44134. [[CrossRef](#)]
38. Zhang, L.; Soselia, D.; Wang, R.; Gutierrez-Farewik, E.M. Lower-limb joint torque prediction using LSTM neural networks and transfer learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 600–609. [[CrossRef](#)] [[PubMed](#)]
39. Serbest, K.; Ozkan, M.T.; Cilli, M. Estimation of joint torques using an artificial neural network model based on kinematic and anthropometric data. *Neural Comput. Appl.* **2023**, *35*, 12513–12529. [[CrossRef](#)]
40. Zhuang, Y.; Leng, Y.; Zhou, J.; Song, R.; Li, L.; Su, S.W. Voluntary control of an ankle joint exoskeleton by able-bodied individuals and stroke survivors using EMG-based admittance control scheme. *IEEE Trans. Biomed. Eng.* **2021**, *68*, 695–705. [[CrossRef](#)] [[PubMed](#)]
41. Zhou, J.; Peng, H.; Zheng, M.; Wei, Z.; Fan, T.; Song, R. Trajectory Deformation-Based Multi-Modal Adaptive Compliance Control for a Wearable Lower Limb Rehabilitation Robot. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2024**, *32*, 314–324. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.