machines

Article

Online Adaptive PID Control for a Multi-Joint Lower Extremity Exoskeleton System Using Improved Particle Swarm Optimization

Jiaqi Liu [1], Hongbin Fang [2,3,4]* and Jian Xu [1]

[1] School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China;
[2] Institute of AI and Robotics, Fudan University, Shanghai 200433, China
[3] Engineering Research Center of AI & Robotics, Ministry of Education, Fudan University, Shanghai 20043, China
[4] Shanghai Engineering Research Center of AI & Robotics, Fudan University, Shanghai 200433, China
* Correspondence: fanghongbin@fudan.edu.cn

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Machines* **2021**, *9*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

Abstract: Robotic exoskeletons have great potential in medical rehabilitation and augmentation of human performance in a variety of tasks. Proposing effective and adaptive control strategies is one of the most challenging issues for exoskeleton systems to work interactively with the user in dynamic environments and variable tasks. This research, therefore, aims to advance the state of the art of the exoskeleton adaptive control by integrating the excellent search capability of metaheuristic algorithms with the PID feedback mechanism. Specifically, this paper proposes an online adaptive PID controller for a multi-joint lower extremity exoskeleton system by making use of the particle swarm optimization (PSO) algorithm. Significant improvements, including a "leaving and re-searching mechanism", are introduced into the PSO algorithm for better and faster updating of the solution and to prevent premature convergence. In this research, a 9-DOF lower extremity exoskeleton with 7 controllable joints is adopted as a test-bench, whose first-principle dynamic model is developed, which includes as many uncertain factors as possible for generality, including human-exoskeleton interactions, environmental forces, and joint unilateral constraint forces. Based upon this, to illustrate the effectiveness of the proposed controller, the human-exoskeleton coupled system is simulated in four characteristic scenarios, in which the following factors are considered: exoskeleton parameter perturbations, human effects, walking terrain switches, and walking speed variations. The results indicate that the proposed controller is superior to the standard PSO algorithm and the conventional PID controller in achieving rapid convergence, suppressing undesired chattering of PID gains, adaptively adjusting PID coefficients when internal or external disturbances are encountered, and improving tracking accuracy in both position and velocity. We also demonstrate that the proposed controller could be used to switch the working mode of the exoskeleton for either performance or energy-saving consideration. Overall, aiming at a multi-joint lower extremity exoskeleton system, this research proposes a PSO-based online adaptive PID controller that can be easily implemented in applications. Through rich and practical case studies, the excellent anti-interference capability and environment/task adaptivity of the controller are exemplified.

Keywords: Exoskeleton robot; model-free control; adaptive control; trajectory tracking control; uncertainties and disturbances

1. Introduction

Powered robotic exoskeletons, according to their purposes, can be divided into three categories: medical, industrial, and military. In the field of medical rehabilitation [1–3], the major objective of powered exoskeletons is to provide force assistance for the elderly with muscle weakness caused by aging or to offer effective rehabilitation to incapacitated patients. In industrial applications [4–7], the exoskeleton technology is increasingly used to assist workers with repetitive tasks and heavy loads for reducing working injury and fatigue-induced errors. In terms of military uses [8–11], the value of using a powered exoskeleton is also in its ability to enhance the operational capability of individual soldiers in military activities, e.g., to increase the load-bearing and distance of cross-country marching.

Generally, the robotic exoskeleton and the human body are highly coupled and constitute a complex system. The complexities originate from three aspects. First, the robotic exoskeleton and the human body are dynamical systems with essential distinctions: the exoskeleton can be considered as a multi-rigid-body model with many degree-of-freedom (DOF), while the human body is fundamentally a rigid-flexible coupling system that is difficult to be accurately modeled. Second, the exoskeleton and the human body are always connected via flexible webbings, whose coupling effect is generally unknown and is also challenging for accurate modeling. Without this information, the force transmission between the human body and the exoskeleton becomes uncertain, and the human-machine system model is incomplete. Third, during operation, the overall human-exoskeleton system would inevitability suffer perturbations from human efforts [12], environmental changes, and other forms of external interferences [13]. All these factors are generally time-variant, indeterminate, and unpredictable, which, as a result, further complicate the system.

Note that for all intents and purposes, active control is a requisite for the powered exoskeleton to take effect. The effectiveness of the control algorithm, as well as the stability of the overall system, strongly relies on a prior dynamic model of the human-exoskeleton system. However, generally, the system is too complicated to be modeled accurately, and previous research mainly focused on developing equivalent models by more or less simplifying or neglecting certain factors for controlling purposes. For example, the human body, which is fundamentally a complex musculoskeletal system, is always modeled into a linked multi-rigid-body system [14–18]; the human-robot interaction (HRI), due to its persistent and unpredictable nature, is usually simplified in the model [19, 20]; the dynamics of the exoskeleton can also be described by a first-order transfer function [19, 21] or a data-driven model [22], while the complexity will exacerbate as the DOF increases.

Based on equivalent or simplified dynamic models, different types of control strategies have been proposed for powered exoskeletons, among which, the most fundamental one is the proportional–integral–derivative (PID) control for trajectory tracking. In general, the established models do not accurately represent the dynamics of the targeted human-exoskeleton system, and external disturbances will be encountered inevitably. To tackle these issues, researchers have proposed several techniques to guarantee control effects. For example, a nonlinear extended state observer was introduced into a state error feedback controller to improve the trajectory tracking performance [23]. In [24], a nonlinear disturbance observer was integrated into the backstepping sliding controller to effectively reduce the influence of uncertainties and external disturbances. Impedance control algorithm is proposed in [25-27] to realize the stability control of exoskeleton system. The neural network is another feasible tool to compensate for nonlinear disturbances [28–30]; however, prior knowledge of the system is always needed to determine the structure and parameters of the neural network, which makes this method easily affected by dynamic changes. The emergence of adaptive controllers based on radial basis function neural network (RBFNN) [31, 32] solves this problem. Incorporating fuzzy logic is also favorable in designing controllers to weaken the influences of external disturbances and parameter uncertainity on control effects [33–35]; however, assigning the membership functions and optimizing the fuzzy rules rely on experience or experimental results, and currently, there is a lack of mature theory. These are cutting-edge methods in exoskeleton control, therefore, the controller in this study will be reasonably compared with these methods to verify the effectiveness of our method.

On the other hand, heuristic algorithms have been integrated with conventional controllers for performance improvement. For example, by employing the genetic algorithm and the particle swarm optimization (PSO) algorithm, an intelligent PID controller is proposed, which could effectively determine the PID gains for a 4-DOF lower extremity exoskeleton [36, 37]. In [38, 39], sliding mode control technology and the genetic algorithm are combined to effectively resist uncertainties and disturbances of a lower extremity exoskeleton. Note that among various metaheuristics, the PSO algorithm possesses unique merits in that it requires no assumption about the problem to be optimized and can search very large spaces of the solution. Hence, the PSO algorithm is capable of dealing with complex high-dimensional nonlinear systems and dynamic optimization problems of various backgrounds. For example, the PSO algorithm has been applied for avoiding obstacles of swarm intelligent robots [40], tuning fuzzy logic controller of an unmanned underwater vehicle [41], and controlling positions of a nonlinear hydraulic system [42]. In spite of its potentials, the PSO algorithm still has some limitations that impede its further applications: the convergence time of the PSO algorithm may be too long to induce undesired chattering phenomena in a control scenario; the possible premature convergence issue of the PSO algorithm would make it difficult to respond to abrupt changes of the system and the environment. In [43], by improving the PSO algorithm, a PSO-based online adaptive PID controller is proposed and applied to a 2-DOF lower extremity exoskeleton for trajectory tracking, which shows significant improvement in averting optimization stagnation and accelerating the convergence. However, note that the reduction of the convergence time is achieved by substantially reducing the number of initial stochastic particle swarm, which would diminish the probability of locating the global optimal solution. Hence, this algorithm could give a satisfactory result for low-dimensional systems, while its applicability in high-dimensional nonlinear systems is questionable, and its capability in dealing with variable tasks and external disturbances is not demonstrated.

In this research, targeting the trajectory tracking control of a lower extremity exoskeleton, an online adaptive PID controller is proposed by synthesizing the excellent searchability of the PSO algorithm and the feedback control mechanism. We demonstrate that the proposed controller does not require an accurate dynamic model or precise system parameters a priori. In addition, the controller possesses unique merits including the capacity in tackling high-dimensional systems, inhibition of undesired chattering, fast convergence, and task/environment adaptability. These are demonstrated by considering an extreme case of a 9-DOF exoskeleton with 7 joints to be controlled (i.e., with 21 PID coefficients to be optimized), which significantly outstrips the current level of exoskeleton prototypes. Moreover, to manifest the disturbance resisting ability and the task/environment adaptability of the controller, simulation experiments are performed in several characteristic scenarios, including parameter uncertainties, human-exoskeleton interactions, walking speed switches, uphill and downhill, etc.

Specifically, by decreasing the inertia weight through a stochastic way, the improved PSO algorithm could converge quickly at the initial stage of control, even with a large initial population for a better global search. To enhance the adaptability and disturbance resisting ability of the controller, a so-called "leaving and re-searching" mechanism is designed and integrated into the PSO algorithm. In detail, by evaluating the evolution of the relative ratio of the mean accumulated errors, the improved PSO algorithm could automatically decide whether to leave the converged local optimum and to re-search the neighborhood. Such a mechanism enables the controller to adjust the PID gains in real-time to adapt to various changes of the working environment and the variable tracking trajectories, as well as to suppress the adverse effects induced by fast and slow-varying parameter perturbations and unknown human-exoskeleton interactions. This is a commendable improvement of the standard PSO algorithm, in which the searching ability is significantly weakened after a local convergence. The abovementioned virtues of the proposed controller are verified by comparing it with the conventional PID controller based on pre-tuned gains. We also provide an example to show that in addition to raising the tracking accuracy, the proposed controller could let the exoskeleton work in a low power mode for energy saving.

The rest of this paper is organized as follows. Section 2 develops a dynamic model of the 9-DOF lower extremity exoskeleton and briefly introduces the experiments for obtaining the reference joint angle signals. This is followed by constructing the online adaptive PID controller by integrating the improved PSO algorithm and the feedback control mechanism. In Section 4, by carrying out several simulation experiments in characteristic scenarios, the effectiveness of the proposed controller and its superiority over the conventional PID controller is demonstrated. Section 5 further discusses the effects of using the proposed controller in a lower power mode for energy saving. Conclusions of this research and a few heuristic discussions are presented in Section 6.

2. Lower extremity exoskeleton

2.1. System description

Figure 1(a) shows a human-exoskeleton system, in which the lower extremity exoskeleton consists of a trunk and two legs, and each leg is further divided into a thigh, a shank, and a foot. They connect to each other through the hip, knee, and ankle joints. Generally, the exoskeleton integrates with the human body through flexible webbings. When a human walk along a straight line, only the motion in the sagittal plane is considered, and the motions in the coronal plane and the horizontal plane are neglected due to their small range [44]. Hence, each hip/knee/ankle joints of the exoskeleton has only one DOF of flexion/extension, with the rotation axis being perpendicular to the sagittal plane. In addition, the trunk can pitch/tilt, and the exoskeleton as a whole has 2 DOF along the x and y directions in the sagittal plane; they apply to the rotation axis of the hip joints. In sum, the lower extremity exoskeleton can be described via a 2-dimensional (2D) model with a total of 9 DOF (Fig. 1(b)); the trunk, the tights, and the shanks are simplified into rigid rods, and the foot is represented by a tetrahedron, which is a triangle in the sagittal plane.

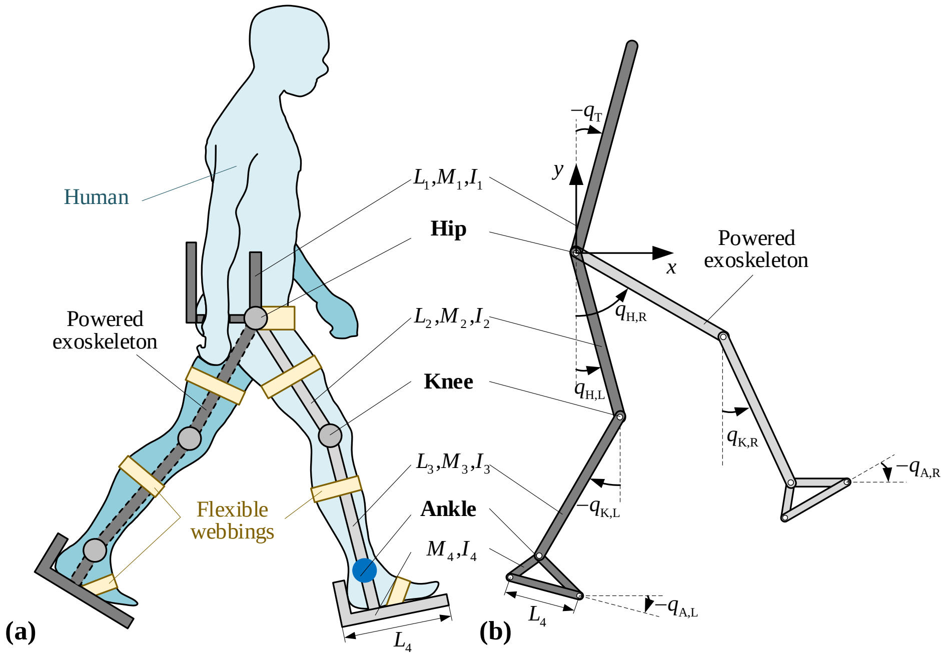

Fig.1. (a) Illustration of a human-exoskeleton system. (b) Dynamic model of the exoskeleton.

For each part, the center of gravity is assumed to locate at the geometric center. For the trunk, the length, mass, and the moment of inertia about the center of gravity are denoted by L_B, M_B, and I_B, respectively. For each thigh, shank, and foot of the exoskeleton, we denote their lengths by L_{T}, L_{S}, and L_{F}, respectively, their masses by M_{T}, M_{S} and M_{F}, respectively, and the moments of inertia with respect to the corresponding centers of gravity by I_{T}, I_{S} and I_{F}, respectively. The angles of the trunk (q_{B}), the thighs (q_{TL} and q_{TR}), and the shanks (q_{SL} and q_{SR}) are measured from the vertical axis, and the angles of the feet (q_{FL} and q_{FR}) are measured from the horizontal axis, with the counterclockwise direction defined as positive. In the above parameters, the subscripts 'B', 'T', 'S', and 'F' denote the trunk, the thighs, the shanks, and the feet, respectively, and the subscripts 'L' and 'R' denote the left and the right side, respectively. In this research, without loss of generality, (q_{TL}, q_{TR}), (q_{SL}, q_{SR}), and (q_{FL}, q_{FR}) are also used to denote the angles of the hip, the knee, and the ankle joints of the exoskeleton, respectively.

Note that the studied 9-DOF exoskeleton possesses 7 joints to be controlled, which exceeds the technological level of the current powered lower extremity exoskeleton prototypes. The reason for studying such a hypothetical example is to demonstrate the capability of the proposed controller in tackling high-dimensional systems through a real-time manner, which will be discussed in Section 3.

2.2. Dynamic modeling

The controller to be constructed in this research does not depend on a prior dynamic model of the human-exoskeleton system. The purpose of modeling in this subsection is to provide a platform for the dynamic simulation experiments, based on which the effectiveness of the proposed controller can be examined, and the unique merits can be uncovered by comparing with the conventional PID control. Based on such purposes, the basic philosophy for modeling is to put as many uncertainty factors as possible into the model, providing that the developed model is numerically processible.

In previous research, the movement of the exoskeleton is always divided into the swinging phase and the stance phase, and they are modeled separately [18, 45–47] so that it is not necessary to directly describe the foot-ground interaction and the constraint force of the knee joint after the leg is straightened. In this research, the dynamics of the exoskeleton in a complete gait cycle is considered, and two strong nonlinear factors, i.e., the foot-ground interaction and the constraint force at the knee joints, are modeled and incorporated. In detail, the exoskeleton shown in Fig. 1(b) possesses 9 DOF, with the generalized coordinate vector $\mathbf{q}$ being assigned as $\mathbf{q}=(x, y, q_{\mathrm{B}}, q_{\mathrm{TL}}, q_{\mathrm{TR}}, q_{\mathrm{SL}}, q_{\mathrm{SR}}, q_{\mathrm{FL}}, q_{\mathrm{FR}})^{\mathrm{T}}$. The governing equation of the exoskeleton is derived via the Lagrange Equation of the second kind,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{N}(\mathbf{q})=\boldsymbol{\tau}_{\mathrm{PID}}+\boldsymbol{\tau}_{\mathrm{H}}+\mathbf{F}_{\mathrm{E}}+\hat{\mathbf{N}}. \tag{1}$$

On the left-hand side of the governing equation, $\mathbf{M}(\mathbf{q})\in\mathbb{R}^{9\times 9}$ is the mass matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\in\mathbb{R}^{9\times 9}$ is the matrix for the Coriolis forces, the centrifugal forces, and the damping forces, $\mathbf{N}(\mathbf{q})\in\mathbb{R}^{9}$ is a vector for the gravitational forces and the elastic restoring forces at the joints; on the right-hand side, $\boldsymbol{\tau}_{\mathrm{PID}}\in\mathbb{R}^{9}$ denotes the control torques, $\boldsymbol{\tau}_{\mathrm{H}}\in\mathbb{R}^{9}$ indicates the human-exoskeleton interactions, $\hat{\mathbf{N}}\in\mathbb{R}^{9}$ denotes the gravity compensation to modify the PID control, and $\mathbf{F}_{\mathrm{E}}\in\mathbb{R}^{9}$ is the generalized force vector corresponding to the foot-ground interaction forces and the unilateral constraint forces of the knee joints, which can be expressed as

$$\mathbf{F}_{\mathrm{E}}=\mathbf{J}^{T}\left(F_{\mathrm{N,heel}}, F_{\mathrm{N,toe}}, F_{f,\mathrm{heel}}, F_{f,\mathrm{toe}}, F_{\mathrm{K,L}}, F_{\mathrm{K,R}}\right)^{T}. \tag{2}$$

Here, $F_{\mathrm{N},i}$ and $F_{f,i}$ ($i=\mathrm{heel},\ \mathrm{toe}$) are the normal pressure and the friction force applied to the toe or heel of the exoskeleton foot, respectively; they together constitute the foot-ground interaction. $F_{\mathrm{K},i}$ ($i=\mathrm{L},\ \mathrm{R}$) denotes the unilateral constraint force of the left and right knee joints after the leg is straightened, which is essentially the contact force originated from the angle-limit mechanism between the tight and shank rods of the exoskeleton. $\mathbf{J}$ is the Jacobian matrix that transforms the above forces into the generalized forces.

In this research, the normal pressures $F_{\mathrm{N},i}$ ($i = \text{heel, toe}$) are described by the smoothed Kelvin-Voigt model [48]

$$F_{\mathrm{N},i} = \left(\frac{1}{2} - \frac{\arctan C_1 \delta_i}{\pi}\right)\left(K_{\mathrm{N},i}\delta_i + \chi_{\mathrm{N},i}\dot{\delta}_i\right), \tag{3}$$

and the friction forces $F_{f,i}$ ($i = \text{heel, toe}$) are described by the smoothed Coulomb dry friction [48]

$$F_{f,i} = \left(-\frac{2\arctan C_2 v_i}{\pi}\right)\mu F_{\mathrm{N},i}. \tag{4}$$

In Eq. (3) and (4), C_1 and C_2 are constants describing the degree of smoothing of the normal pressures and the friction forces; $K_{\mathrm{N},i}$ and $\chi_{\mathrm{N},i}$ are the stiffness and the damping of the contact, respectively; μ is the dry friction coefficient; v_i is the tangential velocity of the toe or the heel with respect to the ground; and δ_i denotes the virtual depth of the foot pressing into the ground, which equals to zero when the foot is off the ground. The parameters of the foot ground contact model are given according to[49]. The unilateral constraint force $F_{\mathrm{K},i}$ ($i = \mathrm{L, R}$) of the knee joint is also described by the smoothed Kelvin-Voigt model

$$F_{\mathrm{K},i} = \left(\frac{1}{2} - \frac{\arctan C_3 \delta_{\mathrm{K},i}}{\pi}\right)\left(K_{\mathrm{K},i}\delta_{\mathrm{K},i} + \chi_{\mathrm{K},i}\dot{\delta}_{\mathrm{K},i}\right). \tag{5}$$

Similarly, C_3 is the constant describing the degree of smoothing, $K_{\mathrm{K},i}$ and $\chi_{\mathrm{K},i}$ are the stiffness and damping of the contact, respectively, and $\delta_{\mathrm{K},i}$ represents the virtual indentation depth between the thigh and shank rods.

Generally, the human-exoskeleton system involves uncertainties and would inevitably suffer disturbances; they can be reflected in the dynamic equation. On one hand, stiffness and damping are assumed at the exoskeleton joints for restoring, their coefficients may suffer fast or slowly varied changes (to be introduced in Section 4). On the other hand, the human-exoskeleton interactions (i.e., the term $\boldsymbol{\tau}_{\mathrm{H}}$) are unknown in nature, and the walking environment (reflected in $F_{\mathrm{N},i}$ and $F_{f,i}$) subjects to uncertainty. We will also consider scenarios in which the walking speed changes, which is reflected in the tracked trajectory.

2.3. Human gait kinematics acquisition

The subject was fully explained of the aims and details of the protocol and voluntarily agreed in participation, with the signing of written informed consent. The purpose of the tests is to obtain the walking gait kinematics ($\mathbf{q}_{\mathrm{H}} \in \mathbb{R}^9$, the subscript 'H' denotes the human) of healthy subjects, which will be used in simulation experiments. The equipment used in the experiment includes Cosmos gaitway platform and Vicon optical motion capture system. The healthy subject is male, with 180 cm in height and 90 kg in weight. Referring to the Helen Hayes model, the healthy subjects were pasted with 18 targets. During the experiment, subjects walked on Cosmos gaitway with different slopes and different speeds. The sampling frequency of optical motion capture system is 100 Hz. The final data obtained from the experiment is the spatial coordinates of each target. Then the spatial coordinates of the hip joint and the spatial rotation angles of the hip, knee and ankle joints can be obtained by using the angle solution method of Helen Hayes model. Detailed gait kinematics acquisition and data decoding procedures can be found in [50]. Different scenarios are set for the healthy subject, including variations of the treadmill speed and the treadmill inclination angle.

Based on the acquired gait kinematics $\mathbf{q}_{\mathrm{H}}$, the human joint torques can be calculated by inversely solving the dynamic model of the human. The governing equation can also be derived based on the Lagrange equation

$$\mathbf{M}_{\mathrm{H}}(\mathbf{q}_{\mathrm{H}})\ddot{\mathbf{q}}_{\mathrm{H}} + \mathbf{C}_{\mathrm{H}}(\mathbf{q}_{\mathrm{H}}, \dot{\mathbf{q}}_{\mathrm{H}})\dot{\mathbf{q}}_{\mathrm{H}} + \mathbf{N}_{\mathrm{H}}(\mathbf{q}_{\mathrm{H}}) = \mathbf{f}. \tag{6}$$

Terms on the left-hand side share the same meaning as those of Eq. (1); $\mathbf{f}$ on the right-hand side denotes the generalized forces corresponding to the foot-ground normal pressures, the foot-ground friction forces, and the joint torques. By substituting the gait kinematics and the measured normal pressures and friction forces, the joint torques expressed in terms of the generalized coordinates can be obtained.

3. PSO based online adaptive PID controller

The exoskeleton system shown in Fig.1 is fundamentally a multiple DOF system, which possesses the following salient characteristics: parameter uncertainties, unknown human-exoskeleton interactions, external perturbations, and environment changes. Aiming at these features, a PSO-based online adaptive PID controller is proposed. Prior to this, the conventional PID tuning method and the standard PSO algorithm are introduced.

3.1. PID tuning based on parameter identification

The classical PID control law for trajectory tracking is

$$\tau_{\mathrm{PID},j}(t) = K_{\mathrm{P},j} e_j(t) + K_{\mathrm{I},j} \int_0^t \dot{e}_j(\tau)\mathrm{d}\tau + K_{\mathrm{D},j} \dot{e}_j(t), \tag{7}$$

where $K_{\mathrm{P},j}$, $K_{\mathrm{I},j}$, and $K_{\mathrm{D},j}$ are proportional, integral, and derivative gains of the PID controller, respectively. Here, the subscript j could take 'B', representing the trunk; 'TR' and 'TL', representing the right and the left thigh; 'SR' and 'SL', representing the right and the left shank; 'FR' and 'FL', representing the right and the left foot. In Eq. (7), $e_j(t)$ is the tracking error in position, and $\dot{e}_j(t)$ is the tracking error in velocity, they are given by

$$e_j(t) = \tilde{q}_j(t) - q_j(t), \quad \dot{e}_j(t) = \dot{\tilde{q}}_j(t) - \dot{q}_j(t), \tag{8}$$

where $\tilde{q}_j(t)$ and $q_j(t)$ denote the desired and the measured position, respectively, $\dot{\tilde{q}}_j(t)$ and $\dot{q}_j(t)$ denote the desired and the measured velocity, respectively.

Note that although the dynamic model of the exoskeleton is strongly nonlinear, the behaviors of the closed-loop system with the PID control are similar to the transient responses of a linear system [51]. By applying the PID control, each joint (corresponding to a generalized coordinate) of the exoskeleton robot can be characterized by a single input single output (SISO) system. Under the range of motion (ROM) condition, by letting one joint move and freezing the other joints [52] each joint can be approximated via a second-order linear system, which can be written as a transfer function in the frequency domain:

$$\frac{q_j(s)}{\tau_{\mathrm{PID},j}(s)} = \frac{K_j}{T_j^2 s^2 + 2\xi_j T_j s + 1}. \tag{9}$$

Applying a step input, the plant gain K_j, the damping ratio ξ_j, and the time constant T_j can be identified. Hence, based on tuning formulas similar to the Ziegler–Nichols and Cohen–Coon methods [51], the PID gains can be obtained.

3.2. Standard PSO algorithm

Particle swarm optimization (PSO) is a population-based optimization technique inspired by swarm behavior such as bird flocking and schooling in nature. It is a metaheuristic that possesses the following merits: PSO requires few or no assumptions about the problem to be optimized, differentiability is not required by the optimization problem, and the solution space that can be searched can be very large. Because of these virtues, PSO is a powerful tool for multidimensional complex optimization problems.

In the standard PSO (SPSO) algorithm, the potential solutions, called particles, move in the search-space following the current optimum particle. Basically, each particle ($\mathbf{p}_i$) in the swarm (with m particles) at the *k*-th iteration has two attributes, the position $\mathbf{X}_i^k$ and the velocity $\mathbf{V}_i^k$. The movement of each particle is determined by a velocity term, which represents the attraction of its own best-known position ($\mathbf{P}_{\mathbf{b}i}$) as well as the entire swarm's best-known position ($\mathbf{P}_\mathbf{g}$) in the search space. Specifically, the velocity of each particle is updated based on Eq. (10), in which c_1 and c_2 are the social learning factors (or, acceleration constants), w is the prescribed inertia weight, r_1 and r_2 are random numbers between 0 and 1 in each iteration. The position of the particle can then be updated according to Eq. (11). The iteration terminates when the prescribed number of iterations is reached, or the critical value of the objective function $f(t)$ is achieved.

$$\mathbf{V}_i^k = w\mathbf{V}_i^{k-1} + c_1 r_1 \cdot \left(\mathbf{P}_{\mathbf{b}i} - \mathbf{X}_i^{k-1}\right) + c_2 r_2 \cdot \left(\mathbf{P}_\mathbf{g} - \mathbf{X}_i^{k-1}\right), \tag{10}$$

$$\mathbf{X}_i^k = \mathbf{X}_i^{k-1} + \mathbf{V}_i^{k-1}. \tag{11}$$

3.3. PSO-based online PID control

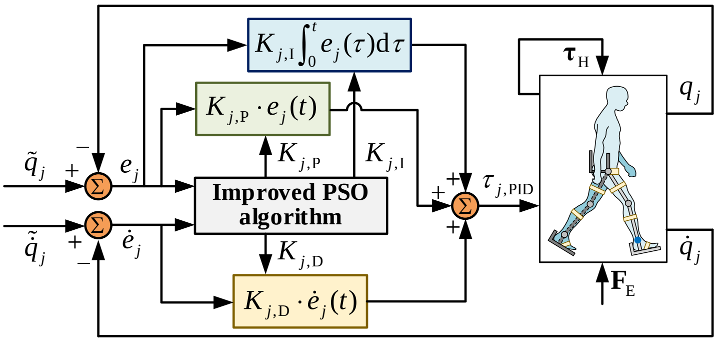

Fig.2. Block diagram of the PSO-based online adaptive PID controller.

As introduced in subsection 2.1, the exoskeleton has 7 joints to be controlled for trajectory tracking. This can be achieved via the PSO-based online PID control. Figure 2 displays the block diagram of the controller. The core idea of the controller is to online optimize the PID coefficients via the PSO algorithm so that the tracking errors in both the position and the velocity can be converged in a short time. To this end, the objective function of the optimization problem is defined as

$$f(t_i) = \frac{1}{t_i}\sum_{t=0}^{t_i}\left(\sqrt{\sum_j e_j^2(t)} + \sqrt{\sum_j \dot{e}_j^2(t)}\right), \tag{12}$$

which measures the mean accumulated error (MAE) in both the position and the velocity up to time t_i. Here, the subscript j could take 'B', 'TR', 'TL', 'SR', 'SL', 'FR', and 'FL'.

Note that with the PID control (see the control law in Eq. (7)), there are 21 parameters (i.e., proportional, integral and derivative gains corresponding to the 7 joints) to be optimized. PSO is an ideal tool for tackling such a high-dimensional problem. Hence, if with PSO, the search space is 21 dimensional, and each particle $\mathbf{p}_i$ in the swarm (with m particles) to be optimized is fundamentally a 21-dimensional vector of PID gains (i.e., $\mathbf{p}_i \in \mathbb{R}^{21}$). Algorithm 1 provides the pseudo-code of using the standard PSO (SPSO) algorithm to online optimize the PID coefficients. To ensure that the controller could online update the PID coefficients, rather than evaluating all particles of the swarm in an offline manner, each particle (i.e., a 21-dimensional vector of the PID gains) is directly injected into the human-exoskeleton system for control. Hence, within each time step, there is only one particle that serves as the PID gains to control the exoskeleton. In other words, it is not necessary to wait for the convergence to the best solution. This ensures the real-time implementation of the controller in a real exoskeleton system. When all the particles in the swarm (in this research, the population size is 100) are traversed, they will be updated based on their own best-known positions and the swarm's best-known position. This is also a simple calculation that can be handled very soon by common control boards. Moreover, the controller can accurately track the desired trajectory by incorporating only the input ($\tilde{q}_j, \dot{\tilde{q}}_j$) and output ($q_j, \dot{q}_j$) of the system, without having to know the dynamic model and relevant prior knowledge of the system.

Although the SPSO algorithm is relatively simple and easy to implement, there are some limitations when applying it for nonlinear system control, especially, when the system involves non-negligible uncertainties and unknown disturbances.

a) Since the particles are generated randomly and the inertia weight remains unchanged for good global searching, the algorithm takes long time to converge. During this convergence period, the control torques may be highly fluctuated due to the unstable PID gains, which would induce undesired vibrations of the exoskeleton and cause harm to the human body.

b) The SPSO algorithm may encounter premature convergence. That is, when the algorithm converges, the SPSO algorithm lacks the adequate capability to let the particles leave the local optimum and to re-search for better solutions. This is particularly vital when the system suffers abrupt changes (e.g., parameter perturbations, unknown human dynamics, and environmental disturbances, etc.) and the current solution is no longer the optimum.

Algorithm 1 Online PID control based on the standard PSO (SPSO) algorithm

```
Start
Initialize $m$
Initialize the swarm to random values
while the experiment does not end do
    for $i = 1:m$ do
        Inject a particle $\mathbf{p}_i$ (with position $\mathbf{X}_i$ and velocity $\mathbf{V}_i$)
        Calculate control torques $\boldsymbol{\tau}_{\mathrm{PID}}(t)$ using Eq. (7)
        Apply $\boldsymbol{\tau}_{\mathrm{PID}}(t)$ to the human-exoskeleton system
        Evaluate the objective function $f(t)$
        Save $\mathbf{p}_i$ and the corresponding value of $f(t)$
        Update $\mathbf{P}_{\mathbf{b}i}$ and $\mathbf{P}_{\mathbf{g}}$
    end for
    Update the swarm (with $m$ particles) by Eq. (10) and (11)
end while
End
```

In the following subsection, improvements are incorporated into the PSO algorithm, for tackling the above two issues.

3.4. Online adaptive PID control based on improved PSO algorithm

Generally, the human-exoskeleton system is subject to uncertainties and disturbances, including unknown human-exoskeleton interactions, uncertain joint stiffness and damping, environment changes, and variations of the tracked trajectory, etc. These factors would significantly diminish the effectiveness of the standard PSO (SPSO) algorithm due to the abovementioned limitations. To overcome these issues, an improved PSO (IPSO) algorithm is proposed and incorporated into the controller. Algorithm 2 displays the pseudo-code of the online adaptive PID controller based on the IPSO algorithm. Improvements of the PSO algorithm mainly fall into the following two aspects:

Algorithm 2 Online adaptive PID control based on improved PSO (IPSO) algorithm

```
 1: Start
 2: Initialize $m$ and $l$
 3: Initialize the swarm to random values; the particle corresponding to pre-tuned PID gains is forced to involve
 4: for $i = 1:m$ do
 5:     Inject a particle $\mathbf{p}_i$ (with position $\mathbf{X}_i$ and velocity $\mathbf{V}_i$)
 6:     Calculate control torques $\boldsymbol{\tau}_{\text{PID}}(t)$ using Eq. (7)
 7:     Apply $\boldsymbol{\tau}_{\text{PID}}(t)$ to the human-exoskeleton system
 8:     Evaluate the objective function $f(t)$ by Eq. (12)
 9:     Save $\mathbf{p}_i$ and the corresponding value of $f(t)$
10:     Update $\mathbf{P}_{\mathbf{b}i}$ and $\mathbf{P}_{\mathbf{g}}$
11: end for
12: Update the swarm (with $m$ particles) by Eq. (10) and (11)
13: Update the inertia weight $w$
14: while the experiment does not end do
15:     if (conditions given by Eq. (13) and (14) hold) then
16:         Randomly select $l$ particles; initialize them randomly based on Eq. (15)
17:         for $i = 1:l$ do
18:             $5 \to 10$
19:         end for
20:         Update the sub-swarm (with $l$ particles) by Eq. (10) and (11)
21:         Update the local inertia weight $w_{\text{Loc}}$
22:         if (condition given by Eq. (14) hold) then
23:             GOTO 17:
24:         else
25:             All particles in the swarm (with $m$ particles) are replaced by $\mathbf{P}_{\mathbf{g}}$
26:             $4 \to 13$
27:         end if
28:     else
29:         $4 \to 13$
30:     end if
31: end while
32: End
```

First, to achieve a balance between the global and local search capability, the inertia weight w is asked to decreases stochastically in the initial stage (say, 2 seconds). Specifically, the inertia weight w follows a normal distribution with the mean decreasing linearly from 2 to 0.6, and the standard deviation equaling to 2. After the initial stage, the inertia weight w stays at a relatively low value of 0.6 to ensure good local search capability. In addition, to improve searching efficiency, the particle (i.e., the PID gains) obtained through the classical tuning method (introduced in subsection 3.1) is forced to be involved into the swarm.

Second, and the most significant, to prevent premature convergence in the context of perturbations and disturbances, a "leaving and re-searching" mechanism is introduced into the controller. In detail, two critical conditions are included. The first is

$$\sigma < \sigma_{\min}, \tag{13}$$

where σ is the standard deviation of the particle velocities, quantifying the velocity discrepancy among particles. When σ drops below a prescribed small value $\sigma_{\min}$, we can presume that the swarm is converged. The second condition is

$$f(t_i) / f(t_{i-1}) > \bar{r}_f, \tag{14}$$

which quantifies the variation tendency of the mean accumulated error. If the ratio is higher than a prescribed value $\bar{r}_f$, which is a constant greater than one, we can presume that the control applied at time t_i is not optimum. This can be induced by certain interferences originating from the system itself or the environment.

If either condition (13) or (14) does not hold (i.e., the swarm has not converged (condition (13) is violated) and/or the control applied at time t_i is the optimum (condition (14) is violated)), steps $4 \rightarrow 13$ in Algorithm 2 (which is similar to the online PID control based on the SPSO algorithm, i.e., steps $5 \rightarrow 13$ in Algorithm 1, except that the inertia weight w is updated each time after the iteration) will be executed. On the other hand, if both conditions are satisfied, i.e., the solution after convergence is not the optimum one, the "leaving and re-searching" mechanism (steps $16 \rightarrow 23$ in Algorithm 2) is executed. Specifically, we randomly select l particles (e.g., $l = 0.1 \times m$) and let them re-search in the neighborhood of the current converged values to decrease the objective function (i.e., to locate a new optimum solution).
The domain for local search is a 21-dimensional supra-sphere with radius R, and the positions of the selected particles are initialized via

$$\mathbf{X}_j^{t_{i+1}} = \mathbf{X}_j^{t_i} + \boldsymbol{\rho} \cdot R, \ (j = 1, ..., l), \tag{15}$$

where $\boldsymbol{\rho} \in \mathbb{R}^{21}$ is a vector of random numbers between -1 and 1. Here, the random numbers corresponding to $K_{\mathrm{P},j}$, $K_{\mathrm{I},j}$, and $K_{\mathrm{D},j}$ are assumed to be identical, respectively. Each time when the l particles are evaluated, the corresponding best-known position of particle j ($\mathbf{P}_{\mathbf{b}j}$) and the best-known position of the entire swarm ($\mathbf{P}_g$) will be updated. Due to the relatively small number of particles and the limited search space, as well as with a stochastically decreasing inertia weight w_{local}, the convergence condition (13) can always be satisfied. However, the local search will continue until condition (14) is no longer met. Thereupon, the positions of all particles in the whole swarm (with m particles) will be forcefully replaced by $\mathbf{P}_g$, and the evolution returns to steps $4 \rightarrow 13$.

Algorithm 2 also indicates that the improved PSO algorithm is a linear time algorithm with time complexity $O(m)$. This means that the running time increases at most linearly with the number of particles m in the swarm. Besides, an increase in the DOF of the controlled system does not affect the time complexity of the algorithm. In sum, it can be hypothesized that the resources required by the proposed algorithm are low, which makes it feasible to deploy on an online controller.

4. Results

This section examines the performance of the proposed controller for trajectory tracking in various scenarios. Four cases are studied in detail: (1) walking on a horizontal plane with unknown human-exoskeleton interactions, (2) walking on a horizontal plane with variable speed, (3) walking uphill and downhill, and (4) walking uphill with variable speed. Model parameter uncertainties are also put into considerations. We assume that the stiffness and damping of the 7 exoskeleton joints subject to fast varying (with frequency 10Hz, stiffness amplitude 50N/rad, and damping amplitude $2\ \mathrm{N \cdot s/rad}$) and slow varying (with frequency 0.4Hz, stiffness amplitude 100N/rad, and damping amplitude 15 $\mathrm{N \cdot s/rad}$) perturbations (Fig. 3(a)), given by

$$\begin{aligned} k &= 150 + 50\mathrm{sign}\left(\sin(5\pi t)\right) + 100\,\mathrm{sign}\left(\sin(\pi(t-3)/5)\right), \\ c &= 5 + 2\mathrm{sign}\left(\sin(5\pi t)\right) + 15\,\mathrm{sign}\left(\sin(\pi(t-3)/5)\right). \end{aligned} \tag{16}$$

Unless otherwise stated, this assumption applies to all four cases to demonstrate the capability of the proposed controller in coping with non-negligible parameter uncertainties.

For comparison purposes, the conventional PID control with pre-tuned gains will also be tested in the abovementioned cases. We will demonstrate that the proposed controller based on the IPSO algorithm could significantly improve the tracking accuracy in both the position and the velocity. Moreover, with the introduced improvements, limitations of the SPSO algorithm can be overcome.

The four cases will be examined through numerical simulations. Specifically, the tracking signals are obtained through experiments introduced in subsection 2.3; when the control torques are generated via the controller, they will be injected into the dynamic model governed by Eq. (1). The dynamic model will be solved in MATLAB 2018b via the forward Euler method with step size 0.0001s, which mimics the behavior of a real human-exoskeleton system. The sampling rate of the controller is 100Hz, which can be easily achieved by common control boards. In the IPSO algorithm, the number of particles $m = 100$. Comparing with the traditional offline PSO, the population size is small. This is because, in this research, the PSO is implemented via a real time manner. Each particle is used as the PID gains to control the exoskeleton in each time step, and the iteration is carried out after all particles in the population are traversed. To ensure the sensitivity of the controller to interference, the population should not be too large. In the "leaving and re-searching" mechanism, the critical values are set as $\sigma_{\min}=0.15$, $\overline{r}_f=1.05$, and the number of particles for local search is $l = 10$. The saturation torque of the exoskeleton is set as $\pm 300\mathrm{N}\cdot\mathrm{m}$.

4.1. *Case I: Walking on a horizontal plane with unknown human-exoskeleton interactions*

In the first case, the controller is expected to track the position and velocity of the human when walking on a horizontal treadmill with a speed of 1.0 m/s. The human efforts, which could be either resistive or assistive to the exoskeleton, are put into consideration. They are expressed as the joint torques in terms of the generalized coordinates (obtained via the method introduced in Subsection 2.3). Fundamentally, they are unknown and unpredictable *a priori*. In this case, for simulation purposes, we assume that the human body provides assistive efforts to the exoskeleton during 20~30s, with 75% of the human joint torques (including the torques at the hip, the knee, and the ankle joints) being applied to the exoskeleton joints; during 40~50s, we assume that the human body resists the exoskeleton's motion, with 75% of the reverse human joint torques being applied to the exoskeleton joints. The applied joint torques are plotted in Fig. 3(b), they need to be transformed to the generalized forces before being substituted into the governing equation.

This case is tackled via the proposed online adaptive PID control based on the IPSO algorithm. To demonstrate its merits, another two control methods, i.e., the conventional PID control with pre-tuned gains and the online PID control based on the SPSO algorithm, are also tested and compared. For both the SPSO and the IPSO algorithms, the search-space is determined based on the pre-tuned PID gains ($K_{\mathrm{P\_Tuned}}$, $K_{\mathrm{I\_Tuned}}$, and $K_{\mathrm{D\_Tuned}}$). Specifically, for the knee and the ankle joints, we have

$$\begin{aligned} &K_{\mathrm{P\_PSO}} \in \left[K_{\mathrm{P\_Tuned}} - 200, K_{\mathrm{P\_Tuned}} + 200\right], \\ &K_{\mathrm{I\_PSO}} \in \left[0, 2K_{\mathrm{I\_Tuned}}\right], \\ &K_{\mathrm{D\_PSO}} \in \left[35, 2K_{\mathrm{D\_Tuned}}\right]. \end{aligned} \tag{17}$$

Here, $K_{\mathrm{D\_PSO}}$ is not allowed to take values lower than 35 to prevent possible instability. For the hip joints, the derivative gain is adjusted into

$$K_{\mathrm{D\_PSO}} \in \left[K_{\mathrm{D\_Tuned}} - 50, K_{\mathrm{D\_Tuned}} + 50\right], \tag{18}$$

the proportional and integral gains remain the same as Eq. (18). In what follows, unless otherwise stated, the search-space defined in Eq. (17) and (18) also apply to the other cases. Figure 3(c) and (d) demonstrate how the PID coefficients evolve with respect to the time when the three controllers are applied, respectively. Overall, at the initial stage of simulation, both the SPSO and the IPSO algorithms have shown their abilities to dynamically adjust the PID gains. Although the PID coefficients vary sharply in the convergence stage,

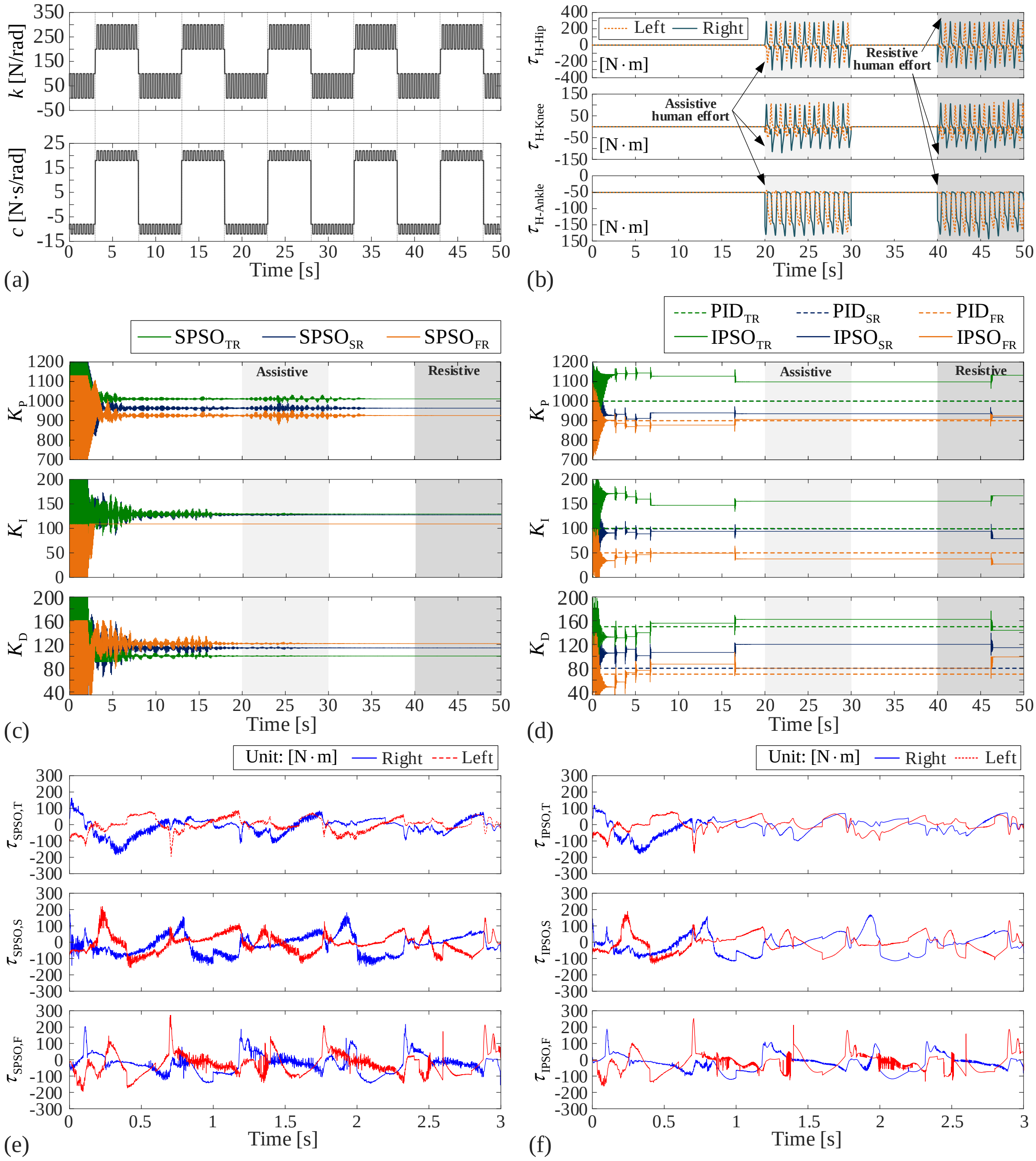

Fig. 3. Tracking results of Case I. (a) Fast and slow variations of the stiffness and damping of the exoskeleton joints. (b) Positive and negative human effects applied to the exoskeleton. (c) and (d) show evolutions of the PID gains corresponding to the SPSO and the IPSO algorithms, respectively; for reference, the pre-tuned PID gains are also plotted in (d). (e) and (f) show the control torques of the exoskeleton joints during 0~3s corresponding to the SPSO and the IPSO algorithms, respectively.

the control performance is largely good (Figure 4). This is because the search space of the particles (i.e., the search space of the PID coefficients) is determined according to the pre-tuned results, and stability can be ensured when the particles locate within the search space. After convergence, the converged PID gains are different from the pre-tuned values, indicating a performance improvement of the trajectory tracking brought by the PSO algorithm. However, in terms of the convergence speed and the chattering phenomena, the IPSO algorithm is significantly superior to the SPSO algorithm. With a stochastically decreasing inertia weight w, the IPSO algorithm converges quickly in about 2 seconds, while the SPSO algorithm needs more than 5 seconds to reach a full convergence. Besides, Fig. 3(c) shows clearly that the SPSO algorithm induces noteworthy chattering of the PID gains during the converging stage and even during the middle stage (e.g., from 20s to 35s for K_p), while the IPSO algorithm could effectively suppress such chattering.

Furthermore, with the "leaving and re-searching" mechanism, after the convergence (say, 0s to 2s), the IPSO algorithm will constantly initiate local searches to restrain the increase rate of the objective function value induced by fast and slow variations of the model parameters and human-exoskeleton interactions. For examples, adjustment of the

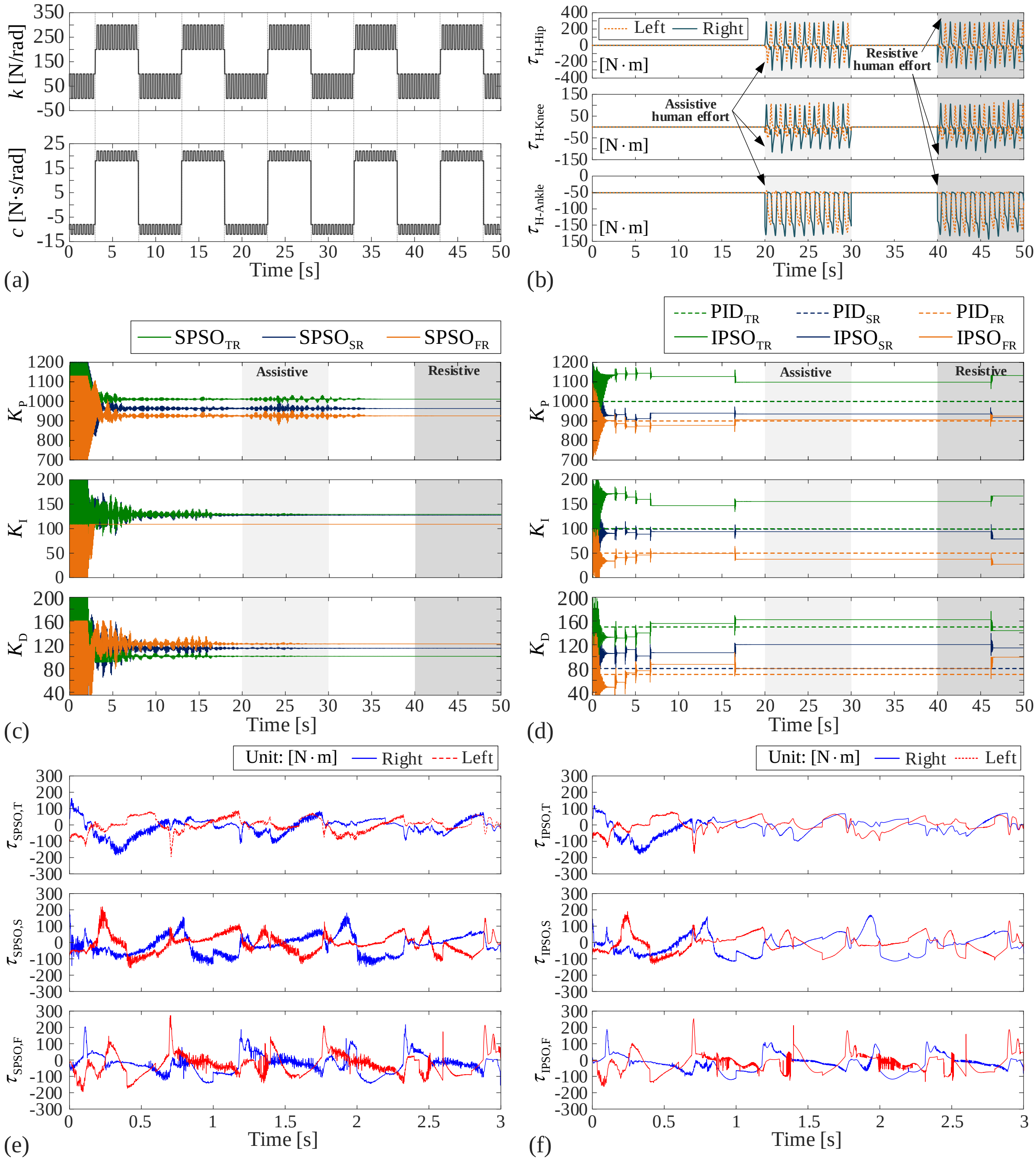

Fig. 3. Tracking results of Case I. (a) Fast and slow variations of the stiffness and damping of the exoskeleton joints. (b) Positive and negative human effects applied to the exoskeleton. (c) and (d) show evolutions of the PID gains corresponding to the SPSO and the IPSO algorithms, respectively; for reference, the pre-tuned PID gains are also plotted in (d). (e) and (f) show the control torques of the exoskeleton joints during 0~3s corresponding to the SPSO and the IPSO algorithms, respectively.

PID gains happened at 16.5s is induced by the increase of the error change rate caused by tracking error accumulation due to the fast and slow varying of the parameters. This PID gain adjustment does not relate to the assistive human efforts during 20-30s. Adjustment of the PID gains at 46.2s is also induced by the increase of the error change rate caused by tracking error accumulation. However, here, the errors come from both the resistive human efforts and the parameter changes. On the contrary the SPSO algorithm fails to adaptively adjust the PID gains even when substantial parameter variations or human-exoskeleton interactions are encountered, it would easily stagnate in a local optimum after convergence.

Correspondingly, Fig. 3(e) and (f) display the time histories (0~3s) of the control torques of the exoskeleton corresponding to the SPSO and the IPSO algorithms, respectively. When the SPSO algorithm is employed, significant chattering phenomena are observed during the converging stage (0~3s). However, when the IPSO algorithm is employed, such chattering is notably suppressed; for the hip and the knee joints, chattering is restrained in less than 1 second.

To demonstrate the effects of the proposed online adaptive PID control based on the IPSO algorithm, Fig. 4(a)~(f) displays the tracking errors in both the position and the velocity corresponding to the right hip, the right knee, and the right ankle. It reveals that the proposed controller could ensure good tracking performance in the stages when the model parameters change sharply or assistive/resistive human effects are applied. Fig. 4(a)~(f) indicates that the tracking errors are greatly affected by the model parameter variations, which cannot be completely suppressed by adjusting the PID coefficients via the IPSO algorithm. Moreover, during the stage with resistive human effects, the exoskeleton joints output higher torques to oppose the resistive effects and to ensure tracking performance.

To further quantify the accuracy, the objective function, i.e., the MAE, is examined. The first stable stage (10~20s), the stage with assistive human effect (30~40s), the stage with resistive human effect (40~50s), and the whole process (0~50s) are studied separately. Comparison of the results indicate that the PID coefficients obtained via the model-based tuning method perform well in angle tracking, and the SPSO algorithm could provide a low level (<5%) reduction of the MAE in position. Due to the torque chattering caused by SPSO (which will cause great damage to the motor), Therefore, considering comprehensively, it cannot be used as a real-time control algorithm. With the IPSO algorithm, the decrease of the MAE in position becomes more significant. On the other hand, in terms of velocity tracking, the merits of incorporating the PSO are obvious. Particularly, in all the examined stages, the IPSO algorithm could achieve more than 16% decline of the MAE in velocity. Reduction of the MAE is the highest when both model parameter variation and human interference exist, manifesting the capability of the proposed controller in coping with strong perturbations.

4.2. Case II: Walking on a horizontal plane with variable speed

In Case II, the controller is expected to track the position and velocity of the human walking on a horizontal treadmill with varying speed. Specifically, the human first walks on a horizontal treadmill at 1.2m/s for 5 seconds, and then jogs at 1.8m/s. Stiffness and damping coefficients of the exoskeleton joint still subject to fast and slow variations determined by Eq. (16), shown in Fig. 5(a). This case is studied via two methods: the conventional PID control with pre-tuned coefficients and the proposed online adaptive PID control based on the IPSO algorithm.

With the proposed controller, the human-exoskeleton system could effectively track the given signals. Fig.5(b) displays a stroboscopic image of the exoskeleton motion before and after the speed change in 7 gait cycles. The stride length after the speed change is obviously longer than that before the speed change. Fig. 5(c) shows the real-time evolution of the PID coefficients during the simulation process. It reveals that the IPSO algorithm

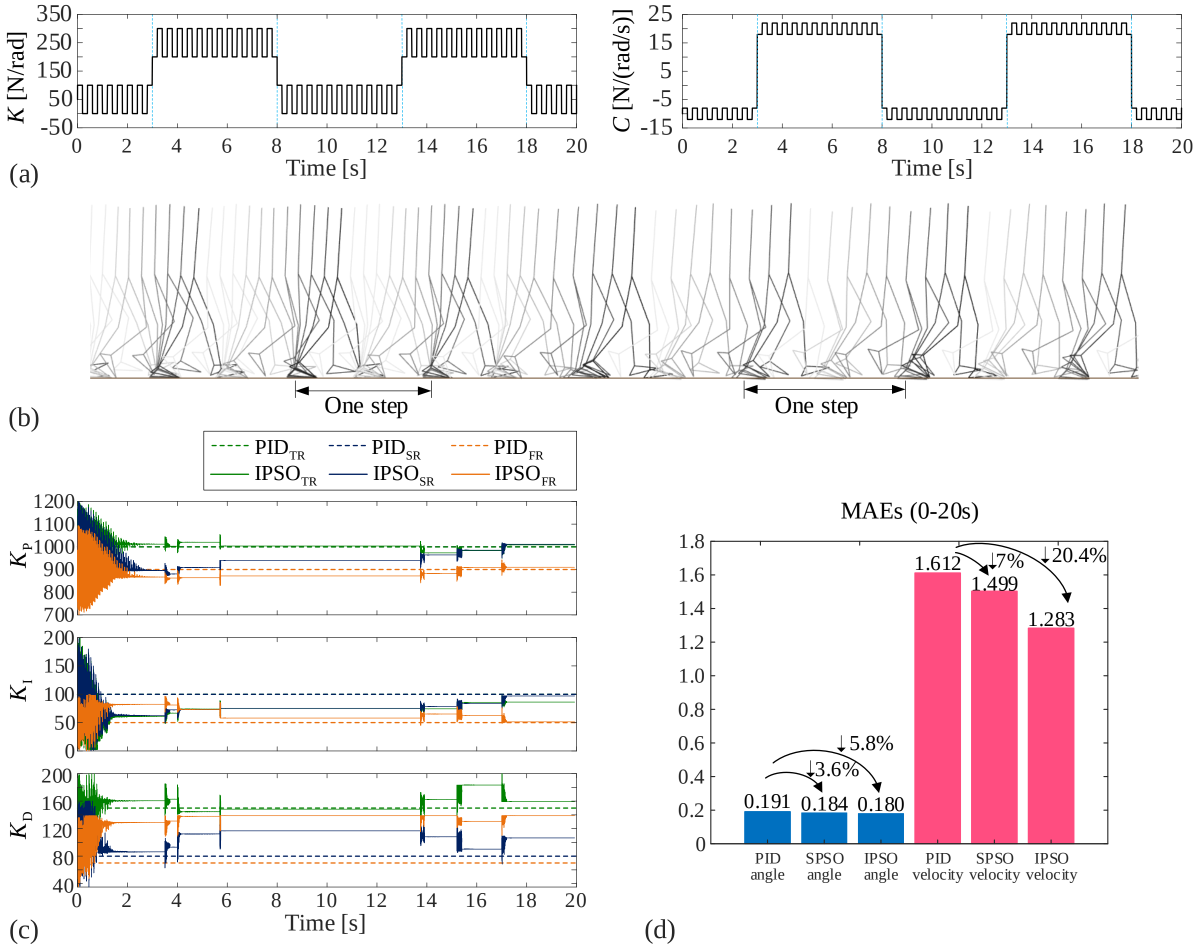

Fig.5. Tracking results of Case II. (a) Fast and slow variations of the stiffness and damping of the exoskeleton joints. (b) Stroboscopic view of dynamic walking with speed varies from 1.2m/s to 1.8m/s. (c) Evolutions of the PID gains corresponding to the IPSO algorithm; for reference, the pre-tuned PID gains are also plotted. (d) Mean accumulated errors (MAEs) in the whole process (0~20s). The reductions brought by the SPSO and the IPSO algorithms are denoted.

could locate an optimal solution that is different from the pre-tuned one in about 2 seconds. After that, during 3~8s, the model parameters are substantially increased, and the walking speed changes from 1.2 to 1.8m/s, these two factors would significantly increase the MAE values. Consequently, the "leaving and re-searching" mechanism is activated several times during this stage to locally search the PID coefficients so that the increase of the objective function can be held back, and the tracking accuracy can be ensured. Activation of the "leaving and re-searching" mechanism is also observed during 13~18s due to the substantial rise of the model parameters. Figure 5(d) quantifies the MAEs in both the position and the velocity corresponding to the three control methods. It shows that the SPSO algorithm could also improve (or equate with) the tracking performance based on conventional PID, although the improvement is limited when compared with the IPSO algorithm. For example, the SPSO algorithm acquires a 3.6% reduction of MAE in position, which is lower than that for the IPSO algorithm (5.8%). In terms of the MAE in velocity, the SPSO induces a 7% reduction, while the IPSO algorithm could achieve a 20.4% reduction. These results indicate that the IPSO algorithm outstrips the SPSO algorithm. Moreover, the IPSO algorithm could effectively restrain the chattering of the control torques, which makes it becomes a better choice than the SPSO algorithm.

4.3. Case III: Walking uphill and downhill

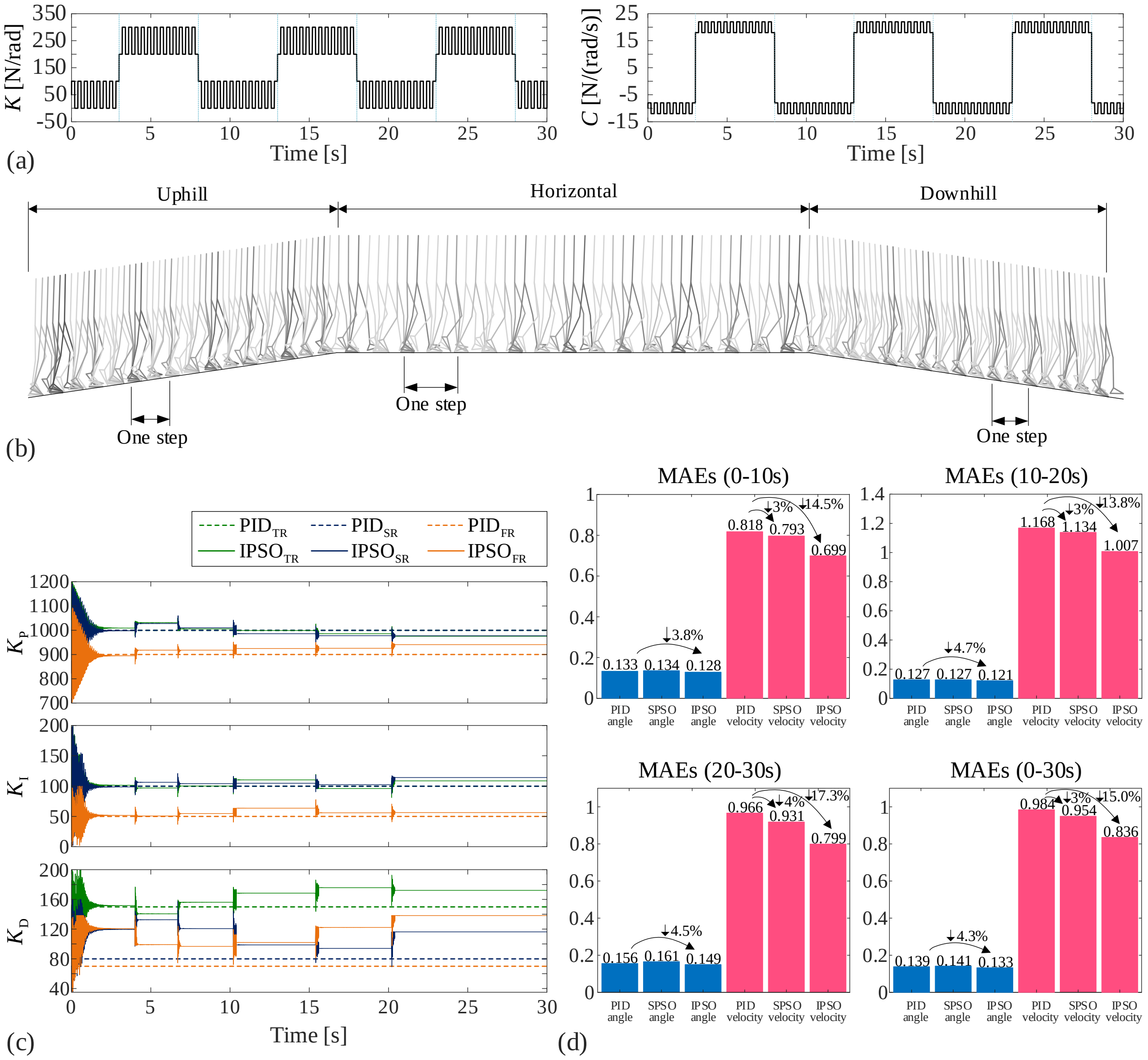

Fig.6. Tracking results of Case III. (a) Fast and slow variations of the stiffness and damping of the exoskeleton joints. (b) Stroboscopic view of dynamic walking in different terrains, including upslope (0~10s), level road (10~20s), and downslope (20~30s). (c) Evolutions of the PID gains corresponding to the IPSO algorithm; for reference, the pre-tuned PID gains are also plotted. (d) Mean accumulated errors (MAEs) in the three stages and the whole process (0~30s). The reductions brought by SPSO and the IPSO algorithm are denoted.

In Case III, during 0-10s, the human walks uphill at 0.6m/s on a 10% gradient treadmill; during 20-30s, the human walks at 1m/s on a horizontal treadmill, and during 20-30s, the human walks downhill at 0.6m/s on a -10% gradient treadmill. Hence, in this case, the environment change is reflected in both the tracked trajectory and the contact force between the foot and the ground (the models of the normal pressure and the friction remain unchanged, but the force directions change). Again, the stiffness and damping coefficients are assumed to suffer fast and slow variations determined by Eq. (16), shown in Fig. 6(a). Case III is examined by both the conventional PID control with pre-tuned gains and the online adaptive PID control based on the IPSO algorithm.

Numerical simulations reveal that the proposed controller could effectively track the signals for uphill, horizontal, and downhill walking. Figure 6(b) displays a stroboscopic image of the exoskeleton motion during the whole process, in which the posture changes and the stride length differences among the three stages can be observed. During the 0~10s uphill stage, the exoskeleton body leans forward, and the stride length is short; during 20-30s, the stride length is longest; and during the 20-30s downhill stage, the exoskeleton body leans backward, and the stride length is short again. Figure 6(c) demonstrates how the PID coefficients are adjusted by the IPSO algorithm. With the "leaving and re-searching" mechanism, after convergence at about 2s, local search of the PID gains is activated twice during 3~8s due to the drastic variations of the model parameters. At about 10s and 20s, local search is activated again, which are caused by the increases of the MAE induced by environment changes (i.e., the terrain changes from uphill to level at 10s, and changes from level to downhill at about 20s) and speed changes.

We also evaluate the MAEs during the three stages and the whole process corresponding to the two control methods. Figure 6 (d) reveals again that the IPSO algorithm has a significant effect in improving the tracking accuracy of the angular velocity. During the whole process, a 15.0% reduction of the MAE in velocity is achieved when compared with the conventional PID control.

4.4. Case IV: Walking uphill with variable speed

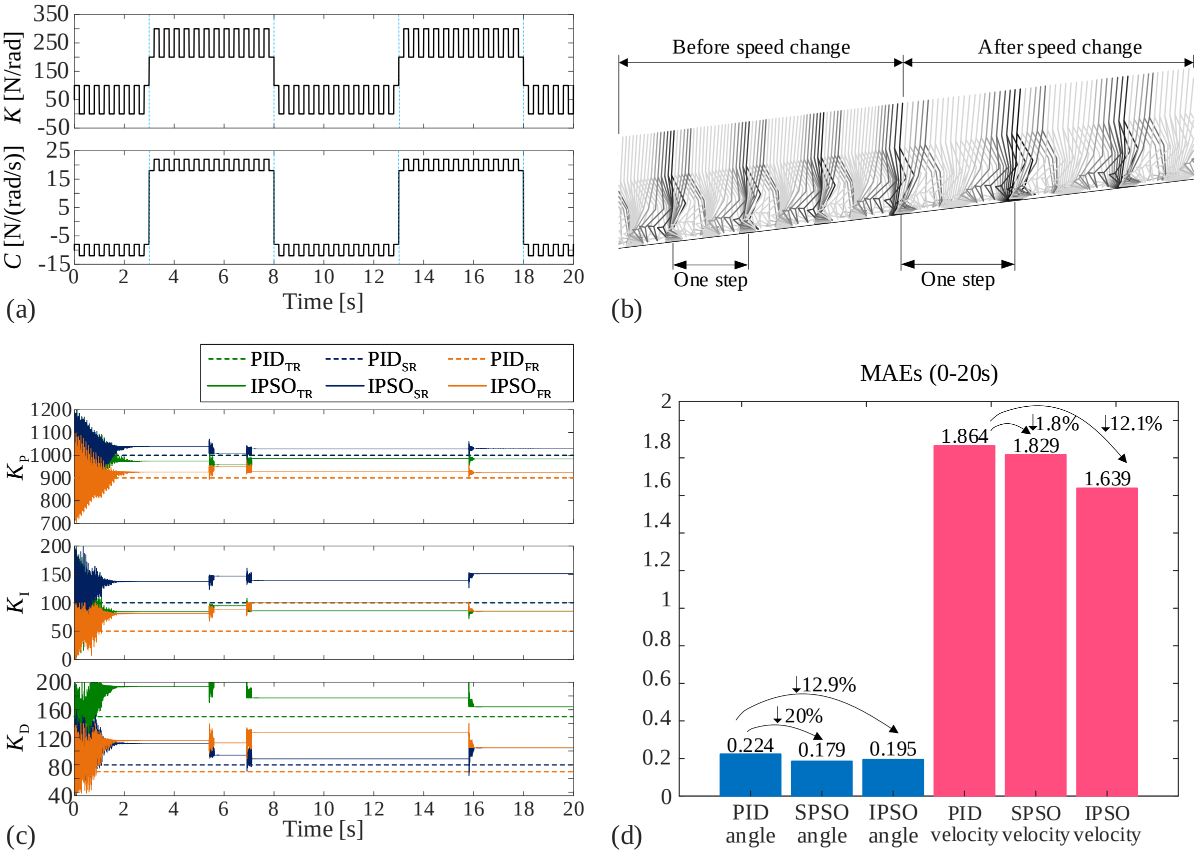

Fig.7. Tracking results of Case IV. (a) Fast and slow variations of the stiffness and damping of the exoskeleton joints. (b) Stroboscopic view of dynamic walking in upslope with variable speed. (c) Evolutions of the PID gains corresponding to the IPSO algorithm; for reference, the pre-tuned PID gains are also plotted. (d) Mean accumulated errors (MAEs) in the whole process (0~20s). The reductions brought by the SPSO and the IPSO algorithms are denoted.

In Case IV, a scenario that the human walks uphill (10% gradient treadmill) with variable speed from 1.2m/s to 1.8m/s is evaluated. Variations of the stiffness and damping coefficients are still assumed in this case, shown in Fig. 7(a). Case IV is examined by both the conventional PID control with pre-tuned gains and the online adaptive PID control based on the IPSO algorithm. Similarly, through the stroboscopic image (Fig. 7(b)), the evolution of the PID coefficients (Fig. 7(c)), and the bar charts of the MAEs (Fig. 7(d)), the effect of the proposed controller can be verified. During the whole process, the "leaving and re-searching mechanism" is activated several times due to the sharp parameter changes and the speed change, which therefore lowers the MAE in position (by 12.9%) and MAE in velocity (by 12.1%) and guarantee the tracking accuracy.

4.5. Summary

Based on the dynamic model of the human-exoskeleton system (established in Section 2), the performance of the proposed control method (i.e., the online adaptive PID control based on the IPSO algorithm) has been examined in four typical cases. Characteristic scenarios, including fast and slow variations of the model parameters (i.e., model uncertainties, in all cases), unknown human-exoskeleton interactions (Case I), environment changes (i.e., external disturbances, Case III), and switch of motion states (i.e., changes of tracked signals, Case II and IV) have been considered. Simulation results indicate that the proposed control method possesses two unique merits over the conventional PID control with pre-tuned gains and the PID control based on the SPSO algorithm. On one hand, by asking the inertia weight to decrease stochastically in the initial stage, the incorporated IPSO algorithm significantly outstrips the SPSO algorithm in achieving fast convergence and suppressing undesirable chattering phenomena (demonstrated in Case I). On the other hand, by activating local searches of the PID gains based on the "leaving and re-searching" mechanism, the IPSO algorithm could effectively resist the increase of the objective function (i.e., the MAE) originated from the unknown uncertainties and perturbations. The proposed controller also surpasses the conventional PID control with pre-tuned gains and the PID control based on the SPSO algorithm in terms of the tracking accuracy, which are evaluated by the MAEs in angle and angular velocity. Overall, through the four case studies in this section, the distinct advantages are uncovered. The proposed controller does not call for prior knowledge of the system; it could adjust the control gains effectively and timely to adapt to new environments and new tracked signals when the human-exoskeleton system is subjected to various model uncertainties, internal/external disturbance, and motion state changes.

5. Using the controller to achieve lower power mode

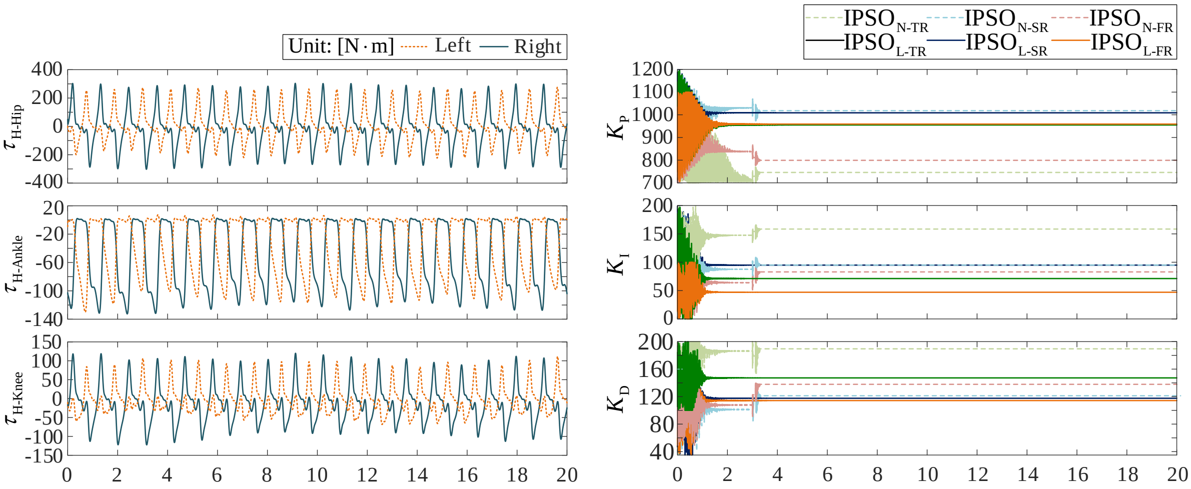

Fig.8. Comparison between the normal mode and the low power mode. (a) Human effects applied to the exoskeleton in the low power mode. (b) Evolution of the PID gains corresponding to the normal mode (denoted by subscripts 'N') and the low power mode (denoted by subscripts 'L'), they converge to significantly different values.

After years of technology accumulation and development, many advanced exoskeleton robots have been proposed for rehabilitation or augmentation, however, they generally consume a lot of energy to achieve the functional objectives [53]. Frequently, the battery life required by long-term high-intensity work is insufficient. Even on a level road, it is still a challenge to let the exoskeleton robot consistently assist human walking for a long time. Hence, we consider if the proposed controller could reduce the output power, and let the exoskeleton robot work in a lower power mode, providing that the tracking performance is not degraded. Here, to evaluate the control performance, the average output power (AOP) of the exoskeleton is defined and examined

$$\bar{p}_{\text{Output}} = \frac{1}{t_i}\sum_{t=0}^{t_i}\sum_{j}\left|\tau_{\text{IPSO},j}(t)\cdot\dot{q}_j(t)\right|, \tag{19}$$

in which j could take 'B', 'TR', 'TL', 'SR', 'SL', 'FR', and 'FL'.

In the normal mode, the motion of the human-exoskeleton system is completely generated by the exoskeleton joints; in this mode, significant energy of the exoskeleton will be consumed, while the user will be fully relaxed under ideal conditions. In the low power mode, both the human joints and the exoskeleton joints contribute positively to the motion; in this mode, the human is assumed to move as in the unequipped state (i.e., human joints output 100% torque as the exoskeleton was not equipped), and the exoskeleton consumes less energy to just follow the human's motion.

To demonstrate the performance difference, the normal mode and the lower power mode are evaluated and compared numerically. Here, zero joint stiffness and damping are assumed. In the normal mode, the simulation is performed without applying human joint torques, while in the lower power mode, 100% human joint torques (including the torques at the hip, the knee, and the ankle joints, obtained from the human gait test corresponding to walking on a horizontal treadmill with speed 1.0m/s) are applied to the exoskeleton joints (Fig. 8(a)). The torques are transformed into the generalized forces for being substituted into the governing equation. Figure 8(b) displays the evolutions of the PID coefficients (within 20s) corresponding to the normal mode and the low power mode under the IPSO algorithm. It shows that in the low power mode, the PID coefficients are converged to completely different values from those in the normal mode. After convergence, the PID gains remain stable since the system does not suffer any external perturbations. As a consequence, the AOP of the exoskeleton in the low power mode is about 25.0% lower (194.35W) than that in the normal mode (259.26W), which indicates effective energy saving. With this, the user would be able to switch the working mode of the exoskeleton according to his/her physical condition, the mission requirement, and the battery life.

6. Discussion

The discussion part mainly covers the following three aspects: tracking accuracy comparison, control system stability, and the triggering of the "leaving and re-searching" mechanism.

(1) To demonstrate the effectiveness of the proposed controller, the tracking performance are compared with several other control methods in the discussion part, in terms of the position tracking error level. In Khamar et al. [24], the sliding mode control and an observer were integrated for a knee exoskeleton, which receives satisfactory tracking effect. Jabbari et al. [29] used neural network to compensate the unknown nonlinear dynamics of a knee exoskeleton. Oktaviani et al. [35] designed a fuzzy active disturbance rejection controller for a hip and knee exoskeleton, and Liu et al. [54] proposed an impedance controller with variable gains for a knee exoskeleton. The tracking performance are provided in Table 1. For comparison purposes, additional situational experiment is performed, in which the human-exoskeleton system is asked to follow the signals of human walking on horizontal plane with constant speed 1.0 m/s, and the stiffness and damping parameters subject to small perturbations, given by

$$\begin{aligned} k &= 150 + 50\text{sign}\left(\sin(5\pi t)\right), \\ c &= 5 + 2\text{sign}\left(\sin(5\pi t)\right). \end{aligned} \tag{20}$$

Table 1 reveal that with small parameter perturbations, our IPSO-based PID controller are comparable with other methods, with the position tracking error ranges between for the hip, and for the knee. The error for the ankle is a little higher than the hip and knee because of the foot-ground reaction forces. However, reports on ankle exoskeletons are rare, and a comparison cannot be made. With fast and slow varying perturbations of the stiffness and damping parameters (Eq. (16)), as well as assistive or resistive human efforts (i.e., Case I), the tracking errors are increased, while the tracking performance are still acceptable. Table 1, thus, verifies the effectiveness and the accuracy of the proposed IPSO-based PID controller.

TABLE1. Tracking accuracy of different methods

Articles	Control methods	joint	Test conditions	Position tracking error level
Khamar et al. [24]	sliding mode control	knee	Sit and swing legs	-0.1~0.1 rad
Jabbari et al. [29]	neural network	knee	Sit and swing legs	-0.05~0.05 rad
Oktaviani et al. [35]	fuzzy control	hip & knee	swing legs without environment effects	Knee: -0.03~0.03 rad
Liu et al. [54]	impedance control	knee	Sit and swing legs	<0.06 rad
Our method	IPSO based online adaptive PID	all	walking with **environment effects** and **fast/slow varying parameters**	Hip: -0.15~0.15 rad Knee: -0.2~0.2 rad Ankle: -0.35~0.2 rad
Our method	IPSO based online adaptive PID	all	walking with **environment effects** and **parameter perturbation**	Hip: -0.05~0.05 rad Knee: -0.02~0.04 rad Ankle: -0.12~0.04 rad

In addition to the comparable tracking error level, the proposed controller possesses some unique merits. The controller does not require an accurate dynamic model or precise system parameters a priori; the controller could deal with high-dimensional systems; the controller could also suppress undesired chattering and adapt to various tasks/environments. Moreover, due to the simplicity and the real-time feature, the controller is promising to be deployed in a practical exoskeleton system.

(2) As for the stability of the controller, first, the particle range is obtained according to the pre-tuned PID gains obtained via the tuning formulas similar to the Ziegler–Nichols and Cohen–Coon methods [51] (introduced in Section 3.1). Based on Lyapunov stability analysis [51], PID gains within this range can make the system stable. Second, as a classical metaheuristic algorithm, the convergence of PSO algorithm has been systematically investigated in [55-57], These studies show that the particles will converge to a certain point in the search-space without divergence. However, this converged point may or may not be the optimum. Note that the convergence to the local optimum has been tackled via different methods, including the Lyapunov stability method [58] and the von Neumann stability criterion [59]. Due to the above two reasons, the stability of the controller can be well ensured. This is manifested in the five examples in this research. Note that the stability issue of the PSO itself falls out of the scope of this research, and thus is not provided in the manuscript.

(3) With the IPSO-based PID controller, adjustments of the PID gains after convergence is caused by the "leaving and re-searching" mechanism. The triggering of this mechanism is determined by the variation tendency of the mean accumulated error but is not directly related to a specific walking condition change. Specifically, if the critical condition $f(t_i)/f(t_{i-1}) > \bar{r}_f$, (i.e., Eq. (14)) is satisfied, we can presume that the control applied at the time t_i is not optimum. Satisfaction of the critical condition can be induced by two kinds of reasons: (1) a sudden surge of the error caused by sharp parameter changes, large external disturbances, or significant working condition variations; (2) an increase of the error change rate induced by error accumulation.

In order to apply the IPSO-based PID controller in real-time scenario, the critical value $\overline{r}_f$ need to be carefully prescribed. An overlarge $\overline{r}_f$ would let the mechanism insensitive so that the algorithm would lose adequate adaptability and trap into a local optimum; otherwise, if $\overline{r}_f$ is very close to 1, the algorithm would become overly sensitive, which would induce undesired chattering of the PID gains. Therefore, rather than a constant value, $\overline{r}_f$ should be selected case-by-case.

7. Conclusion

An IPSO-based model-independent adaptive PID controller is proposed in this paper, based on which, the human-exoskeleton system could operate effectively for trajectory tracking, which is demonstrated via numerical simulations in a few characteristic scenarios.

The most significant algorithm improvement lies in the introduction of the "leaving and re-searching" mechanism. The essence of the mechanism is to test whether the currently converged solution is still optimum; if not, the converged particles will be asked to leave their current positions and to re-search locally for a better solution such that the objective function value decreases.

To manifest the unique merits of the proposed controller, simulation experiments in four characteristic working scenarios are performed, which are based on a 9-DOF first-principle dynamic model of the exoskeleton robot. Specifically, the control performance is compared with those obtained from the conventional PID control with pre-tuned gains and the online PID control based on the SPSO algorithm. Specifically, the IPSO based on adaptive PID controller features with the following virtues: relatively faster convergence, suppression of the chattering phenomena, real-time adjusting of the PID gains when subjecting to interferences, and significantly reduced tracking errors. We also provide an example to demonstrate that the proposed controller can be exploited for letting the exoskeleton work in an energy-saving low power mode.

Finally, we list a few directions of future work by introspecting the limitations of this research. (1) Experimental verification is not performed in this research, partly due to the lacking of a mature 7-DOF exoskeleton prototype. It is necessary to experimentally verify the effectiveness of the proposed controller in a reduced-DOF exoskeleton prototype in future research. Moreover, we believe that there is no impediment to implement the proposed controller in complex 7-DOF exoskeleton prototypes because of its low requirement of computing resources. (2) Due to the metaheuristic nature of the PSO algorithm, the obtained PID gains for each test would be necessarily different. In the simulations, we notice that such differences did not substantially deteriorate the control performance. However, when applying the proposed IPSO-based online PID controller to a practical exoskeleton, the possible destabilization should be considered, and it would be necessary to provide theoretical evidence to prove the stability. (3) The control performance also relies on the physical parameters of the exoskeleton; if the exoskeleton itself is poor in dynamic stability and energy efficiency, the IPSO algorithm would hardly improve the control performance. This, therefore, calls for efforts in exoskeleton design optimization.

Author Contributions: J.L. performed the human subject experiment, developed the model and controller, carried out the computational simulations, and drafted the manuscript. H.F. conceived this study, involved in the model development and data interpretation, critically revised the manuscript, and acquired funding for this work. J.X. supervised this research and acquired funding for this work. All authors approved the final draft.

Funding: This research is supported by the National Key Research and Development Project of China (Grant No. 2018YFB1307305) and the Shanghai Rising-Star Program under Grant No. 20QA1400800.

Data Availability Statement: All the data are shown in the tables, figures of this paper.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Bortole, M.; Venkatakrishnan, A.; Zhu, F.; Moreno, J.C.; Francisco, G.E.; Pons, J.L.; Contreras-vidal, J.L. The H2 robotic exoskeleton for gait rehabilitation after stroke : early findings from a clinical study. J. Neuroeng. Rehabil. 2015, 12, 1–14, doi:10.1186/s12984-015-0048-y.
2. Jayaraman, P.T.A.; Rymer, W.Z. Exoskeletons for Rehabilitation and Personal Mobility : Creating Clinical Evidence. Biosyst. Biorobotics 2017, 16, 21–24, doi:10.1007/978-3-319-46532-6.
3. Mitzner, T.L.; Sanford, J.A.; Rogers, W.A. Closing the Capacity-Ability Gap : Using Technology to Support Aging With Disability. Innov. Aging 2018, 2, 1–8, doi:10.1093/geroni/igy008.
4. Zhang, T.; Huang, H.H. A Lower-Back Robotic Exoskeleton: Industrial Handling Augmentation Used to Provide Spinal Support. IEEE Robot. Autom. Mag. 2018, 25, 95–106, doi:10.1109/MRA.2018.2815083.
5. Barjuei, E.S.; Caldwell, D.G.; Ortiz, J. Bond graph modeling and kalman filter observer design for an industrial back-support exoskeleton. Designs 2020, 4, 1–20, doi:10.3390/designs4040053.
6. Huysamen, K.; de Looze, M.; Bosch, T.; Ortiz, J.; Toxiri, S.; O'Sullivan, L.W. Assessment of an active industrial exoskeleton to aid dynamic lifting and lowering manual handling tasks. Appl. Ergon. 2018, 68, 125–131, doi:10.1016/j.apergo.2017.11.004.
7. Kim, H.J.; Noh, J.; Yang, W. Knee-assistive robotic exoskeleton (KARE-1) using a conditionally singular mechanism for industrial field applications. Appl. Sci. 2020, 10, doi:10.3390/app10155141.
8. Chiara, M.; Silvestro, C.; Jos, M.; Symposium, I.; Robotics, W. HeSA, Hip Exoskeleton for Superior Assistance. Biosyst. Biorobotics 2017, 16, 319–323, doi:10.1007/978-3-319-46532-6.
9. Dong, W.; Liu, C.; Zhang, Q.; Xiong, C. Design and Evaluation of an Active Ankle Exoskeleton in Gait Assistance. 2019 IEEE/ASME Int. Conf. Adv. Intell. Mechatronics 2019, 318–322.
10. Hong, M.B.; Kim, G.T.; Yoon, Y.H. ACE-Ankle: A Novel Sensorized RCM (Remote-Center-of-Motion) Ankle Mechanism for Military Purpose Exoskeleton. Robotica 2019, 37, 2209–2228, doi:10.1017/S0263574719000845.
11. Van Diik, W.; Van De Wijdeven, T.; Hölscher, M.M.; Barents, R.; Könemann, R.; Krause, F.; Koerhuis, C.L. Exobuddy - A Non-Anthropomorphic Quasi-Passive Exoskeleton for Load Carrying Assistance. 2018 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics 2018, 336–341, doi:10.1109/BIOROB.2018.8487794.
12. Silva, P.C.; Silva, M.T.; Martins, J.M. Evaluation of the contact forces developed in the lower limb/orthosis interface for comfort design. Multibody Syst. Dyn. 2010, 24, 367–388, doi:10.1007/s11044-010-9219-6.
13. Arcos-Legarda, J.; Cortes-Romero, J.; Beltran-Pulido, A.; Tovar, A. Hybrid disturbance rejection control of dynamic bipedal robots. Multibody Syst. Dyn. 2019, 46, 281–306, doi:10.1007/s11044-019-09667-3.
14. Aliman, N.; Ramli, R.; Haris, S.M. Modeling and co-simulation of actuator control for lower limb exoskeleton. 2018 3rd Int. Conf. Control Robot. Eng. ICCRE 2018 2018, 94–98, doi:10.1109/ICCRE.2018.8376441.
15. Nabipour, M.; Moosavian, S.A.A. Dynamics Modeling and Performance Analysis of RoboWalk. Proc. 6th RSI Int. Conf. Robot. Mechatronics, IcRoM 2018 2018, 445–450, doi:10.1109/ICRoM.2018.8657593.
16. S. Ali A. Moosavian, Mahdi Nabipour, Farshid Absalan, V.A. RoboWalk: Explicit Augmented Human-Robot Dynamics Modeling for Design Optimization. Br. J. Psychiatry 2019, 111, 1009–1010, doi:10.1192/bjp.111.479.1009-a.
17. Paparisabet, M.A.; Dehghani, R.; Ahmadi, A.R. Knee and torso kinematics in generation of optimum gait pattern based on human-like motion for a seven-link biped robot. Multibody Syst. Dyn. 2019, 47, 117–136, doi:10.1007/s11044-019-09679-z.
18. Miripour Fard, B.; Bruijn, S.M. On the manipulability of swing foot and stability of human locomotion. Multibody Syst. Dyn. 2019, 46, 109–125, doi:10.1007/s11044-018-09664-y.
19. Taherifar, A.; Vossoughi, G.; Selk Ghafari, A. Optimal target impedance selection of the robot interacting with human. Adv. Robot. 2017, 31, 428–440, doi:10.1080/01691864.2016.1272491.

20. Taherifar, A.; Vossoughi, G.; Ghafari, A.S. Assistive-compliant control of wearable robots for partially disabled individuals. Control Eng. Pract. 2018, 74, 177–190, doi:10.1016/j.conengprac.2018.02.004.
21. Aguirre-Ollinger, G.; Nagarajan, U.; Goswami, A. An admittance shaping controller for exoskeleton assistance of the lower extremities. Auton. Robots 2016, 40, 701–728, doi:10.1007/s10514-015-9490-8.
22. Rosenberg, M.C.; Banjanin, B.S.; Burden, S.A.; Steele, K.M. Predicting walking response to ankle exoskeletons using data-driven models: Predicting walking response to ankle exoskeletons using data-driven models. J. R. Soc. Interface 2020, 17, doi:10.1098/rsif.2020.0487rsif20200487.
23. Zhao, L.; Liu, X.; Wang, T. Trajectory tracking control for double-joint manipulator systems driven by pneumatic artificial muscles based on a nonlinear extended state observer. Mech. Syst. Signal Process. 2019, 122, 307–320, doi:10.1016/j.ymssp.2018.12.016.
24. Khamar, M.; Edrisi, M. Designing a backstepping sliding mode controller for an assistant human knee exoskeleton based on nonlinear disturbance observer. Mechatronics 2018, 54, 121–132, doi:10.1016/j.mechatronics.2018.07.010.
25. Ortlieb, A.; Baud, R.; Tracchia, T.; Denkinger, B.; Herzig, Q.; Bleuler, H.; Bouri, M. An Active Impedance Controller to Assist Gait in People with Neuromuscular Diseases: Implementation to the Hip Joint of the AUTONOMYO Exoskeleton. Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics 2018, 2018-Augus, 537–543, doi:10.1109/BIOROB.2018.8487913.
26. Qu, Z.; Wei, W.; Wang, W.; Zha, S.; Li, T.; Gu, J.; Yue, C. Research on Fuzzy Adaptive Impedance Control of Lower Extremity Exoskeleton. Proc. 2019 IEEE Int. Conf. Mechatronics Autom. ICMA 2019 2019, 939–944, doi:10.1109/ICMA.2019.8816452.
27. Chinimilli, P.T.; Qiao, Z.; Rezayat Sorkhabadi, S.M.; Jhawar, V.; Fong, I.H.; Zhang, W. Automatic virtual impedance adaptation of a knee exoskeleton for personalized walking assistance. Rob. Auton. Syst. 2019, 114, 66–76, doi:10.1016/j.robot.2019.01.013.
28. Cao, Y.; Huang, J. Neural-network-based nonlinear model predictive tracking control of a pneumatic muscle actuator-driven exoskeleton. IEEE/CAA J. Autom. Sin. 2020, 7, 1478–1488, doi:10.1109/JAS.2020.1003351.
29. Jabbari Asl, H.; Narikiyo, T.; Kawanishi, M. Neural network-based bounded control of robotic exoskeletons without velocity measurements. Control Eng. Pract. 2018, 80, 94–104, doi:10.1016/j.conengprac.2018.08.005.
30. Yang, Y.; Huang, D.; Dong, X. Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning. Neurocomputing 2019, 323, 256–264, doi:10.1016/j.neucom.2018.09.085.
31. Hao, Z.; Liu, K.; Wei, Q. Adaptive Neural Network Control of Lower Limb Exoskeleton Robots Using Disturbance Observer. 2020 5th IEEE Int. Conf. Adv. Robot. Mechatronics (ICARM) 2020, 246–251, doi:10.1109/ICARM49381.2020.9195319.
32. Zhang, L.; Hu, Y.; Su, H.; Li, J.; Ovur, S.E. Adaptive Neural Network Control for a Lower-Limb Exoskeleton using Variable Stiffness Transferring. 2020 5th IEEE Int. Conf. Adv. Robot. Mechatronics 2020, 240–245, doi:10.1109/ICARM49381.2020.9195302.
33. Yin, K.; Xiang, K.; Pang, M.; Chen, J.; Anderson, P.; Yang, L. Personalised Control of Robotic Ankle Exoskeleton Through Experience-Based Adaptive Fuzzy Inference. IEEE Access 2019, 7, 72221–72233, doi:10.1109/ACCESS.2019.2920134.
34. Wu, H.; Lang, L.; An, H.; Wei, Q.; Ma, H. Fuzzy cerebellar model articulation controller-based adaptive tracking control for load-carrying exoskeleton. Meas. Control (United Kingdom) 2020, 53, 1472–1481, doi:10.1177/0020294020944962.
35. Oktaviani.J Fuzzy Active Disturbance Rejection Control for Direct Driven Exoskeleton of Swing Leg. 2020 5th Int. Conf. Adv. Robot. Mechatronics 2020.
36. Amiri, M.S.; Ramli, R.; Ibrahim, M.F. Hybrid design of PID controller for four DoF lower limb exoskeleton. Appl. Math. Model. 2019, 72, 17–27, doi:10.1016/j.apm.2019.03.002.
37. 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, 1–16, doi:10.3390/math8112040.

38. Long, Y.; Du, Z.J.; Wang, W.D.; Dong, W. Robust Sliding Mode Control Based on GA Optimization and CMAC Compensation for Lower Limb Exoskeleton. Appl. Bionics Biomech. 2016, 2016, 13, doi:10.1155/2016/5017381.
39. Pupek, L.A.Z. Development, Testing, and Implementation of An Optimized Sliding Control Scheme for Unstable Under-Actuated Mechanical Systems. 2016.
40. Yang, J.; Wang, X.; Bauer, P. Extended PSO Based Collaborative Searching for Robotic Swarms with Practical Constraints. IEEE Access 2019, 7, 76328–76341, doi:10.1109/ACCESS.2019.2921621.
41. Mohd Aras, M.S.; Abdullah, S.S.; Jaafar, H.I.; Yusof, A.A.; Mohd Tumari, M.Z.; Yan, H.G. Optimization of Single Input Fuzzy Logic Controller Using PSO for Unmanned Underwater Vehicle. Proc. ofthe 10th Natl. Tech. Semin. Underw. Syst. Technol. 2018 2019, 538, 15–26, doi:10.1007/978-981-13-3708-6_2.
42. Zhang, H.; Liang, Y.; Zhang, W.; Xu, N.; Guo, Z.; Wu, G. Improved PSO-Based Method for Leak Detection and Localization in Liquid Pipelines. IEEE Trans. Ind. Informatics 2018, 14, 3143–3154, doi:10.1109/TII.2018.2794987.
43. 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, doi:10.1016/j.asoc.2017.06.012.
44. Fujimoto, T.I.; T.K.; K.O.; G.S.; R.O.; H. Kinematic determinants of human locomotion. J. Physiol. 1996, 494, 863–879.
45. Yan, L.J.; Shen, Q.M.; Yang, D.; Qiao, J.M.; Datseris, P. Kinematic and Dynamic Modeling and Analysis of a Lower Extremity Exoskeleton. 2019 16th Int. Conf. Ubiquitous Robot. UR 2019 2019, 712000, 625–630, doi:10.1109/URAI.2019.8768556.
46. Guan, X.; Zhong, C.H.; Huang, J.; Liao, W.H. Trajectory tracking for swing phase of the lower limb exoskeleton. IEEE Int. Conf. Robot. Biomimetics, ROBIO 2019 2019, 315–320, doi:10.1109/ROBIO49542.2019.8961410.
47. Chicoma, C.; Cieza, O.; Pujada, E.; Elias, D.A. Modeling for the design of a lower limb exoskeleton for people with gait impairments; Springer International Publishing, 2019; Vol. 71; ISBN 9783030164232.
48. Khadiv, M.; Moosavian, S.A.A.; Yousefi-Koma, A.; Sadedel, M.; Ehsani-Seresht, A.; Mansouri, S. Rigid vs compliant contact: an experimental study on biped walking. Multibody Syst. Dyn. 2019, 45, 379–401, doi:10.1007/s11044-018-09653-1.
49. Brown, P.; McPhee, J. A 3D ellipsoidal volumetric foot–ground contact model for forward dynamics. Multibody Syst. Dyn. 2018, 42, 447–467, doi:10.1007/s11044-017-9605-4.
50. Yang, L.; Hongbin, F.; Jian, X.; Qining, W.; Xiaoxu, Z. DYNAMIC MODELING AND ANALYSIS OF THE LOWER LIMB PROSTHESIS WITH FOUR-BAR LINKAGE PROSTHETIC KNEE. Chinese J. ofTheoretical Appl. Mech. 2020, 52, 1157–1173.
51. Yu, W. PID CONTROL WITH INTELLIGENT COMPENSATION FOR EXOSKELETON ROBOTS; 2018; ISBN 9780128133804.
52. Amiri, M.S.; Ramli, R.; Ibrahim, M.F. Initialized Model Reference Adaptive Control for Lower Limb Exoskeleton. IEEE Access 2019, 7, 167210–167220, doi:10.1109/ACCESS.2019.2954110.
53. Aoustin, Y.; Formalskii, A.M. Walking of biped with passive exoskeleton: evaluation of energy consumption. Multibody Syst. Dyn. 2018, 43, 71–96, doi:10.1007/s11044-017-9602-7.
54. Liu, L.; Misgeld, B.J.E.; Pomprapa, A.; Leonhardt, S. A Testable Robust Stability Framework for the Variable Impedance Control of 1-DOF Exoskeleton with Variable Stiffness Actuator. IEEE Trans. Control Syst. Technol. 2021, 29, 2728–2737, doi:10.1109/TCST.2021.3051716.
55. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. Inf. Process. Lett. 2003, 85, 317–325, doi:10.1016/S0020-0190(02)00447-7.
56. Clerc, M.; Kennedy, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 2002, 6, 58–73, doi:10.1109/4235.985692.
57. Jiang, M.; Luo, Y.P.; Yang, S.Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Inf. Process. Lett. 2007, 102, 8–16, doi:10.1016/j.ipl.2006.10.005.
58. Kadirkamanathan, V.; Selvarajah, K.; Fleming, P.J. Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Trans. Evol. Comput. 2006, 10, 245–255, doi:10.1109/TEVC.2005.857077.

59. Gopal, A.; Sultani, M.M.; Bansal, J.C. On Stability Analysis of Particle Swarm Optimization Algorithm. Arab. J. Sci. Eng. 2020, 45, 2385–2394, doi:10.1007/s13369-019-03991-8.