

Self-adaptive Particle Swarm Optimization with Human-in-the-loop for Ankle Exoskeleton Control

Jinfeng Wang,¹ Biwei Tang,^{2*} Muye Pang,² Kui Xiang,² and Zhaojie Ju³

¹Department of Information, Wuhan Huaxia University of Technology, Wuhan 430070, China

²Intelligent System Research Institute, School of Automation, Wuhan University of Technology,
Wuhan 430070, China

³School of Computing, University of Portsmouth, Portsmouth, PO1 3HE, UK

(Received December 17, 2020; accepted June 15, 2021; online published June 25, 2021)

Keywords: ankle exoskeleton, human-in-the-loop, particle swarm optimization, muscle activity, electromyography signal

Ankle exoskeletons have recently aroused increasing research interest owing to their potential in enhancing human locomotion. Nevertheless, inter-subject variability makes the control of human–exoskeleton interaction complicated. To handle this problem, we designed a human-in-the-loop (HIL) approach to optimization control for an ankle exoskeleton during walking based on an improved self-adaptive particle swarm optimization (ISAPSO) algorithm and the iterative learning control (ILC) algorithm. As part of the development, a self-adaptive updating strategy was first proposed to tune the three key parameters of each particle to obtain a better trade-off between the global and local search abilities of ISAPSO. Moreover, since the performance of the proposed ISAPSO heavily relies on its convergence property, we provided a convergence-guaranteed parameter setting rule for the proposed optimizer after analytically investigating its convergence. Finally, the developed HIL optimization approach was verified via experimental tests on eight subjects. The experimental results revealed that the proposed method reduced the soleus muscle activities of the eight subjects by 23.46 ± 10.21 , 47.04 ± 13.54 , 28.52 ± 8.14 , and $8.58 \pm 3.82\%$ compared with those for the static assistance condition, zero-torque model, normal walking condition, and the case optimized by standard particle swarm optimization, respectively. Thus, the proposed method can be regarded as an alternative in the field of exoskeleton HIL optimization control.

1. Introduction

Owing to their great potential in improving human mobility, ankle exoskeletons have aroused increasing research interest over the last few decades.^(1–4) To achieve this potential, it is paramount to set suitable control parameters for an ankle exoskeleton since they have a profound impact on its performance.^(5–7) Traditionally, the control parameters have either been hand-tuned or set on the basis of average measured biomechanical properties of a given population group.^(8,9) However, these two methods are time-consuming and tedious owing to inter-subject

*Corresponding author: e-mail: tangbiwei@whut.edu.cn

<https://doi.org/10.18494/SAM.2021.3227>

differences.^(10,11) The drawbacks of these two methods may limit the applications of an exoskeleton for different users. Thus, to increase the suitability of the exoskeleton for different users, it is necessary to develop methods for automatically determining its control parameters.⁽¹²⁾ However, the complexities and uncertainties of both humans and the human–exoskeleton interaction impose great challenges regarding this issue.⁽¹³⁾

Despite major efforts to resolve this issue, few of them achieved successes until recent breakthroughs in human-in-the-loop (HIL) optimization for wearable robotic devices.^(14–16) The HIL optimization method can automatically optimize the control parameters of a device based on the measured physiological signals of each user.^(17–19) Therefore, this method can not only surmount the flaws of the two aforementioned parameter setting approaches, but also broaden the application of a given exoskeleton for different users.⁽²⁰⁾

Two key issues need to be addressed in exoskeleton HIL optimization. The first is the establishment of the objective function. Currently, building the objective function via the metabolic cost of each subject is one of the most commonly used approaches. As an alternative approach, the application of electromyography signals (EMGs) of the lower-limb muscles of each subject to build the objective function has been proposed due to the easy measurement and implementation of EMGs. For example, in addition to the metabolic cost, Zhang *et al.* have also tested and verified the feasibility of using the soleus muscle activity of each subject to establish the objective function for human walking.⁽¹¹⁾ Some similar works using one- or two-legged EMGs to indicate the optimization metric in exoskeleton HIL optimization can be found in Refs. 21 and 22.

The selection of the optimizer is the second key issue in exoskeleton HIL optimization.⁽²³⁾ Direct methods, such as the response surface method⁽²⁴⁾ and gradient descent method, were the first approaches used in this field. However, this type of method may be sensitive to noise, which would reduce its optimization performance for a low signal-to-noise ratio.⁽²⁵⁾ By virtue of their swarm-based nature, evolutionary algorithms (EAs) have shown excellent optimization performance in problems with a low signal-to-noise ratio. As a result, EAs such as covariance matrix adaptation (CMA-ES)^(11,26) and the genetic algorithm (GA)⁽²⁷⁾ have been recently employed in exoskeleton HIL optimization.

As one of the most popular EAs, the particle swarm optimization (PSO) algorithm has been applied to optimize exoskeleton control parameters in recent years due to its simplicity and high convergence speed.^(28,29) Nevertheless, standard PSO has been proven to be a divergent algorithm and suffer from difficulties in adjusting its global and local search abilities.^(30–33) These deficiencies would greatly constrain the performance of standard PSO in exoskeleton HIL optimization.

To enhance human walking mobility by employing PSO and the iterative learning control (ILC) algorithm,^(34,35) a two-level HIL optimization control framework was developed in this study to identify the assistive strategies that the ankle exoskeleton provides to humans. The high level implements PSO to optimize the four torque pattern control parameters, i.e., the peak torque, peak time, rise time, and fall time of the ankle exoskeleton, while minimizing the soleus muscle activity of each subject. The low level implements ILC to govern the movements of the ankle exoskeleton, so that it can provide walking assistance to each subject by following the desired torque trajectory generated by the high-level optimizer.

To enhance the optimization efficiency, we propose an improved PSO algorithm, which we called self-adaptive PSO (ISAPSO), in the high-level optimizer in the developed HIL framework. To achieve this goal, a self-adaptive updating strategy is first developed to adjust the three key coefficients (inertia weight and cognitive and social acceleration parameters) of each particle in ISAPSO. Subsequently, after analytically investigating the convergence of ISAPSO, a convergence-guaranteed parameter setting rule is provided for this optimizer so as to sufficiently guarantee the convergence of the proposed PSO.

It is important to note that the idea of using the HIL framework to optimize the assistive strategies of an ankle exoskeleton is not original. In this study, we simply referred to the framework developed in previous studies.^(11,14,15,17) The main contributions of this study involve the development of the optimization algorithm in the high level of the HIL framework and are summarized as follows:

- (1) An improved self-adaptive PSO algorithm, named ISAPSO, is proposed for the high level of the HIL framework.
- (2) To ensure a good balance between the global and local search powers of ISAPSO, a novel self-adaptive updating strategy is developed to tune the three key coefficients of each particle in this algorithm.
- (3) A convergence-guaranteed parameter setting rule is developed to sufficiently ensure the convergence of ISAPSO, which is followed by an investigation of the convergence of this PSO.

To evaluate the proposed method, we conducted five different experimental tests on eight subjects. The experimental results showed that the proposed method reduced the soleus muscle activity of the eight subjects by 23.46 ± 10.21 , 47.04 ± 13.54 , 28.52 ± 8.14 , and $8.58 \pm 3.82\%$ (mean \pm standard deviation), compared with those for the static assistance condition,⁽¹¹⁾ zero-torque model,⁽¹¹⁾ normal walking condition,⁽¹¹⁾ and the case optimized by standard PSO, respectively. Thus, our proposed method can be considered as an alternative in the field of exoskeleton HIL optimization.

The rest of this paper is organized as follows. The research problem is described and formulated in Sect. 2. Section 3 mainly discusses the proposed ISAPSO and the convergence-guaranteed parameter setting rule for this algorithm. The ISAPSO-based HIL framework is depicted in Sect. 4. The experiments, results, and analysis are reported in Sect. 5. Section 6 completes this paper by drawing conclusions and suggesting several future directions of study. Finally, the analytical investigation of the convergence of ISAPSO is given in the Appendix.

2. Problem Statement and Formulation

As shown in Fig. 1, a human–exoskeleton interaction system is considered in this paper. In such a system, an ankle exoskeleton is unilaterally worn on a human ankle to assist walking. The dynamics of this system can be mathematically represented as follows:

$$\tau_a + \tau_h = M(q)\ddot{q} + H(q, \dot{q}), \quad (1)$$

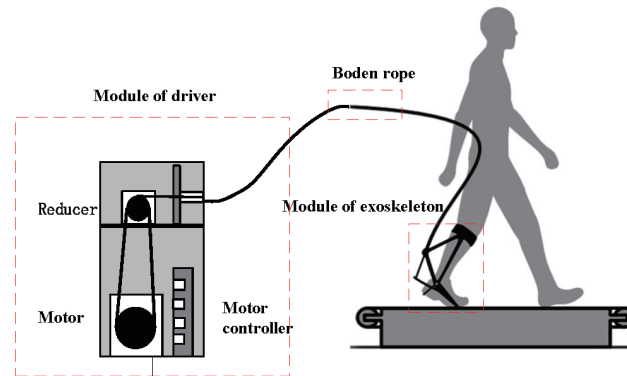


Fig. 1. (Color online) Global scheme of the human–exoskeleton interaction system.

where τ_a is the assistive torque provided by the exoskeleton. τ_h is the actual torque exerted by the human ankle, $M(q)$ and $H(q, \dot{q})$ are the inertia moment and Coriolis dynamics of the device, and q , \dot{q} , and \ddot{q} represent the exoskeleton's position, velocity, and acceleration, respectively.

Constrained by the complexities of both humans and the exoskeleton dynamics, $M(q)$ and $H(q, \dot{q})$ cannot be directly obtained in the above system. This leads to major challenges in designing the controller of the exoskeleton. Moreover, the inter-subject variations could lead to different responses of different subjects to a given control strategy. This also significantly increases the difficulty of the exoskeleton controller design, particularly in terms of individualized designs for different users.

To address these challenges, as shown in Figs. 2 and 3, we have designed a two-level HIL optimization control framework by integrating ISAPSO and ILC. The high-level optimizer uses ISAPSO to optimize the peak torque, peak time, rise time, and fall time of the device in each stride of the subject while optimizing the subject's soleus muscle activity. The four optimized torque pattern control parameters are then used to determine the desired ankle torque, which is regarded as the input of the low-level controller. The low-level controller subsequently tracks the desired ankle torque on the basis of ILC, so that the device can yield assistive torque to the ankle.

Since the desired ankle torque is optimized individually in the high-level optimizer on the basis of the subject's soleus muscle activity, the HIL framework can not only reduce the difficulty of designing the low-level controller, but also broaden the application of the exoskeleton to different users. The proposed ISAPSO and the design of the HIL framework are described in the following sections.

3. Proposed ISAPSO

3.1 Review of standard PSO

Inspired by birds flocking and homing, standard PSO was first proposed by Kennedy and Eberhart in 1995. Each agent in the PSO file is regarded as a particle and represents a candidate

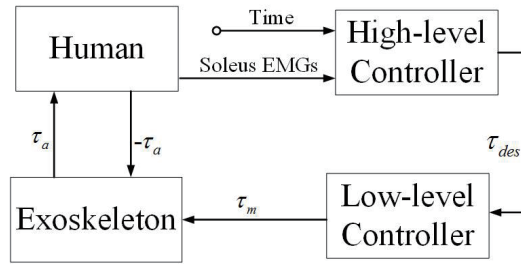


Fig. 2. Control flow chart of the developed HIL framework.

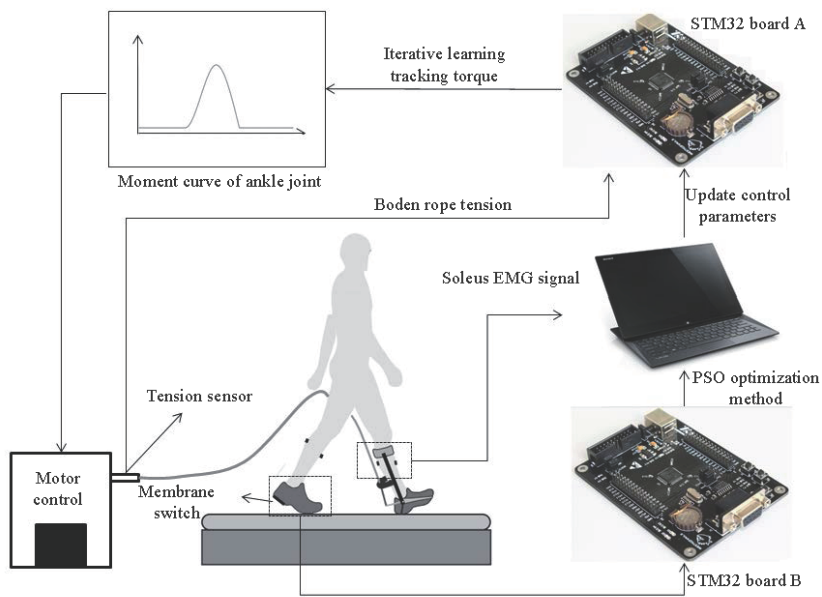


Fig. 3. (Color online) Schematic diagram the developed HIL framework.

solution to an optimization problem. During the search for solutions, each particle dynamically updates its search information based on its own experience and those of its companions as follows:⁽³⁶⁾

$$\begin{cases} V_m^{k+1} = \omega_m V_m^k + c_{1m} r_1 (pbest_m^k - X_m^k) + c_{2m} r_2 (gbest^k - X_m^k) \\ X_m^{k+1} = X_m^k + V_m^{k+1} \end{cases}, \quad (2)$$

where V_m^k and X_m^k are the velocity and position of particle m at iteration k , $pbest_m^k$ and $gbest^k$ are the personal best position of particle m and the global best position of the swarm at iteration k , ω_m is the inertia weight of each particle, c_{1m} and c_{2m} are two positive real parameters representing the cognitive and social acceleration coefficients of each particle, and r_1 and r_2 are two random numbers uniformly distributed in $[0, 1]$, respectively.

Originally, standard PSO was proposed to solve some nonlinear optimization problems through social interactions among particles. As shown in Eq. (2), each particle regulates its velocity and position information through its own flight memory and those of other particles. Note that V_m^k and X_m^k are dimensionless and simply used to model social interactions among particles. Thus, they can be directly added in Eq. (2). For more details of standard PSO, please refer to Ref. 36.

3.2 Self-adaptive updating strategy proposed in ISAPSO

The global and local search abilities of PSO heavily rely on the particle's inertia weight and the social and cognitive acceleration parameters.⁽³³⁾ The basic principles regarding how these three parameters affect the global and local search abilities of PSO can be summarized as follows: (1) a large inertia weight enhances the global search ability, whereas a small inertia weight enhances the local search ability; (2) a large cognitive acceleration parameter compared with the social acceleration parameter facilitates the global search; (3) a large social component compared with the cognitive component augments the local search power.^(36,37)

Normally, the global search ability of PSO needs to be intensified in the early stage of the evolution, so that particles can search through the entire solution space to reduce the likelihood of missing the area containing the global optimum.^(33,37) On the other hand, the local search ability needs to be strengthened in the late stage of the evolution to encourage particles to search carefully in the area including the global optimum to increase the likelihood of finding the global optimum.^(33,37)

Since the above three parameters in standard PSO are predefined constants and there is no difference between the social and cognitive acceleration parameters, this algorithm has been found to have difficulties in obtaining a good balance between its global and local search abilities.⁽³⁷⁾ This shortcoming can greatly reduce the optimization performance of standard PSO. To increase the performance, it is necessary to remedy this deficiency of standard PSO.

The principle and analysis noted above provide a meaningful insight into how to effectively balance the global and local search abilities of PSO via changing the three key coefficients of each particle. Herein, to surmount the flaw of standard PSO, we propose a self-adaptive updating strategy to tune the three key coefficients of each particle in ISAPSO as follows:

$$\omega_m = (\omega_{max} - \omega_{min}) \exp\left(-\frac{\delta_1 k}{\beta}\right) + \omega_{min}, \quad (3)$$

$$c_{1m} = (c_{1s} - c_{1f}) \exp\left(-\frac{\delta_2 k}{\beta}\right) + c_{1f}, \quad (4)$$

$$c_{2m} = (c_{2s} - c_{2f}) \exp\left(-\frac{\delta_3 k}{\beta}\right) + c_{2f}, \quad (5)$$

$$\delta_1 = \frac{\omega_{max} - \omega_{min}}{k_{max}}, \quad (6)$$

$$\delta_2 = \frac{c_{1s} - c_{1f}}{k_{max}}, \quad (7)$$

$$\delta_3 = \frac{c_{2s} - c_{2f}}{k_{max}}, \quad (8)$$

$$\beta = \|gbest^k - pbest_m^k\|, \quad (9)$$

where ω_{max} and ω_{min} are the maximum and minimum values of the inertia weight, c_{1s} , c_{1f} , c_{2s} , and c_{2f} are the initial and final values of the cognitive and social acceleration coefficients, β is the Euclidean distance between the particle's personal best position and the swarm's global best position, and k_{max} is a predefined constant representing the maximum iteration number of the swarm, respectively. Note that $c_{1s} > c_{1f}$ and $c_{2s} < c_{2f}$ in the above updating strategy. Also, the particles in ISAPSO still use Eq. (2) to update their velocity and position information. Moreover, since the exponential function is known for its rapid growth, the three key coefficients of each particle in ISAPSO are exponentially updated using the above strategy to increase the convergence speed of ISAPSO.

3.3 Parametric analysis of ISAPSO

To ensure a good balance between the global and local search abilities of ISAPSO, a self-adaptive parameter updating rule given by Eqs. (3)–(9) is developed for this algorithm. To explain how this rule can ensure a good balance between the two abilities in principle, we conduct a parametric analysis of this algorithm based on the rule developed in this section. From Eqs. (3)–(5), it is evident that ω_m and c_{1m} decrease, whereas c_{2m} increases as the iteration number k increases. These changes in the three parameters indicate that the global search ability of this algorithm can be increased in the early stage of evolution and will be dominated by the local search ability in the later stage. Thus, on the basis of the principle stated in Sect. 3.2, the likelihood of missing the global optimal solution can be decreased early in the evolution, and the likelihood of finding the global optimum can be increased later in the evolution.

Moreover, the global and local search abilities of ISAPSO can be adapted using the introduced parameter β . It is clear from Eqs. (3)–(5) that the variations in ω_m and c_{1m} decrease while the change in c_{2m} increases with increasing β . This may imply that the global search ability of ISAPSO can be increased for a large β . From Eq. (9), a larger β indicates a larger distance between the personal best position of the particle and the global best position of the swarm. In this case, it is logical to enhance the global search ability of ISAPSO, so that particles can quickly converge toward the global best position and to reduce the likelihood of missing the

potentially vital solution space during the search. In contrast, it is desirable to strengthen the local search ability of ISAPSO for a small value of β to maximize the likelihood of finding the global optimum.

In summary, the three key coefficients of each particle in our proposed ISAPSO can be adaptively updated via the self-adaptive updating strategy defined by Eqs. (3)–(9) while complying with the principle noted in Sect. 3.2. As a result, the optimization efficacy of ISAPSO can be increased.

3.4 Convergence-guaranteed parameter setting rule for ISAPSO

As shown in the Appendix, we have proven that ISAPSO converges if and only if

$$\begin{cases} 0 < c_{1m}r_1 + c_{2m}r_2 < 2\omega_m + 2 \\ -1 < \omega_m < 1 \end{cases}, \quad (10)$$

where c_{1m} and c_{2m} are the cognitive and social acceleration coefficients, respectively, ω_m is the inertia weight of each particle, and r_1 and r_2 are two random numbers uniformly distributed in $[0, 1]$.

As shown by Eq. (10), despite obtaining the necessary and sufficient condition for the convergence of ISAPSO, the rule for setting the three coefficients under this condition to guarantee the convergence of ISAPSO still remains unknown. To address this issue, we provide a parameter setting rule to sufficiently guarantee the convergence of ISAPSO by integrating the developed rule defined by Eqs. (3)–(9) and the condition given by Eq. (10).

Lemma 1. The convergence of ISAPSO is sufficiently guaranteed only if the initial and final values of the three coefficients of each particle meet the following conditions.

$$\begin{cases} 2\omega_{min} + 2 > c_{1s} + c_{1f} \\ 1 > \omega_{max} > \omega_{min} > -1 \\ c_{1s} = c_{2f} > c_{1f} = c_{2s} > 0 \end{cases} \quad (11)$$

Proof. Since c_{1m} and c_{2m} are positive and the two stochastic numbers r_1 and r_2 are uniformly distributed in $[0, 1]$, it is trivial that $c_{1m} \geq c_{1m}r_1$ and $c_{2m} \geq c_{2m}r_2$. Thus, one can easily see that

$$\begin{cases} 2\omega_{min} + 2 > c_{1m} + c_{2m} \\ -1 < \omega_m < 1 \\ c_{1m}, c_{2m} > 0 \end{cases} \Rightarrow \begin{cases} 0 < c_{1m}r_1 + c_{2m}r_2 < 2\omega_m + 2 \\ -1 < \omega_m < 1 \end{cases}. \quad (12)$$

If $c_{1s} = c_{2f}$ and $c_{1f} = c_{2s}$ in Eq. (11), it is trivial from Eqs. (4) and (5) and Eqs. (7) and (8) that $c_{1m} + c_{2m} = c_{1s} + c_{1f}$ for any particle at any iteration in ISAPSO. Moreover, one can easily see from Eqs. (3)–(5) that $\omega_{min} \leq \omega_m \leq \omega_{max}$, $c_{1f} \leq c_{1m} \leq c_{1s}$, and $c_{2s} \leq c_{2m} \leq c_{2f}$. Therefore, we have

$$\begin{cases} 2\omega_{min} + 2 > c_{1s} + c_{1f} \\ 1 > \omega_{max} > \omega_{min} > -1 \\ c_{1s} = c_{2f} > c_{1f} = c_{2s} > 0 \end{cases} \Rightarrow \begin{cases} 2\omega_{min} + 2 > c_{1m} + c_{2m} \\ -1 < \omega_m < 1 \\ c_{1m}, c_{2m} > 0 \end{cases}. \quad (13)$$

From Eqs. (12) and (13), it is trivial that

$$\begin{cases} 2\omega_{min} + 2 > c_{1s} + c_{1f} \\ 1 > \omega_{max} > \omega_{min} > -1 \\ c_{1s} = c_{2f} > c_{1f} = c_{2s} > 0 \end{cases} \Rightarrow \begin{cases} 2\omega_{min} + 2 > c_{1m} + c_{2m} > 0 \\ -1 < \omega_m < 1 \end{cases}. \quad (14)$$

Thus, the proof of **Lemma 1** is easily completed owing to the fact that the right-hand-side inequalities in Eq. (14) are the necessary and sufficient convergence condition [refer to Eq. (10)] for ISAPSO.

Recall that ω_{min} , ω_{max} , c_{1s} , c_{1f} , c_{2s} , and c_{2f} are predefined constants in the self-adaptive updating strategy defined by Eqs. (3)–(9). Therefore, the convergence condition given by Eq. (11) can be easily satisfied. This indicates that the convergence of ISAPSO can be easily and sufficiently guaranteed by suitably setting the initial and final values of the three coefficients of the particle. To this end, we set $\omega_{max} = 0.9$, $\omega_{min} = 0.4$, $c_{1s} = c_{2f} = 2$, and $c_{1f} = c_{2s} = 0.1$ in the self-adaptive updating strategy proposed for ISAPSO based on our pilot test. With this suggested parameter setting, the convergence of the position and velocity trajectories of a particle in ISAPSO are illustrated in Fig. 4.

4. HIL Framework Developed for Ankle Exoskeleton During Human Walking

Leveraging ISAPSO and ILC, a two-level HIL optimization control framework is developed for the ankle exoskeleton during human walking. To minimize the soleus muscle activity in each gait cycle, the high-level optimizer uses ISAPSO to optimize the peak torque, peak time, rise

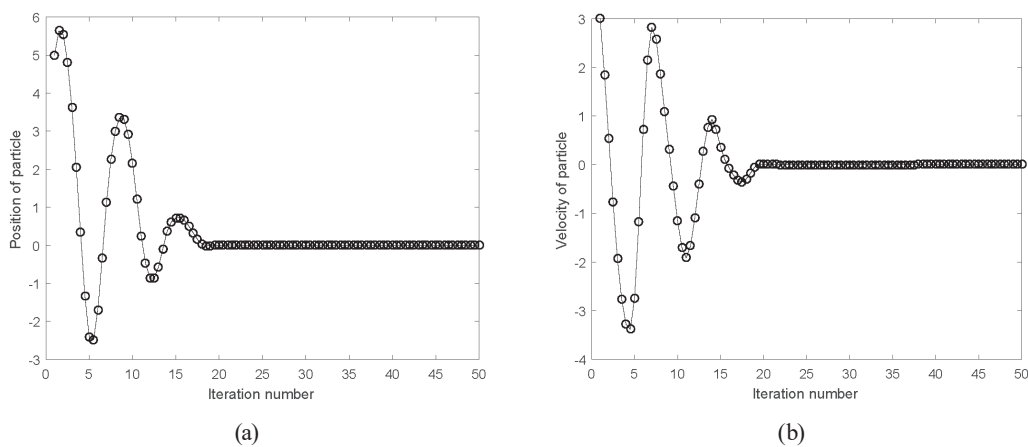


Fig. 4. Convergent position and velocity trajectories of particle in ISAPSO: (a) position and (b) velocity.

time, and fall time of the device. The low level implements ILC to control the exoskeleton to follow the desired optimized ankle torque. The formulation of the ankle torque pattern during walking and the designs of the two levels in the HIL framework are next outlined.

4.1 Formulation of ankle torque pattern during walking

As visualized in Fig. 5, during walking, the ankle torque has a hill-like shape that is primarily defined by four parameters: the peak torque, peak torque time, rise time, and fall time.^(11,24) Universally, the hill-like pattern is composed of rising and falling cubic spline sections as follows:^(11,24)

$$\begin{cases} \tau_1(t) = a_1 t^3 + b_1 t^2 + c_1 t + d_1 \\ \tau_2(t) = a_2 t^3 + b_2 t^2 + c_2 t + d_2 \end{cases} \quad (15)$$

where $\tau_1(t)$ and $\tau_2(t)$ respectively denote the rising and falling cubic splines. t represents the time elapsed since the heel strike in the current stride, and a_i , b_i , and c_i ($i = 1, 2$) are shape parameters that must be set.

The two spline sections defined by Eq. (15) satisfy^(11,24)

$$\begin{cases} \tau_1(t_p - t_r) = 2 \\ \tau_1(t_p) = \tau_p \\ \tau_2(t_p) = \tau_p \\ \tau_2(t_p + t_f) = 2 \\ \bullet \\ \tau_1(t_p) = 0 \\ \bullet \\ \tau_2(t_p) = 0 \\ \bullet \\ \tau_1(t_p - t_r) = 0 \\ \bullet \\ \tau_1(t_p + t_f) = 0 \end{cases}, \quad (16)$$

where τ_p , t_p , t_r , and t_f are the peak torque, peak torque time, rise time, and fall time, respectively.

Note that constraints given by Eq. (15) fully define the shape of the ankle torque. Once τ_p , t_p , t_r , and t_f are known, the shape parameters a_i , b_i , and c_i ($i = 1, 2$) in Eq. (15) can be obtained by solving Eq. (16). Once the shape parameters are determined, the ankle torque in Eq. (15) can be determined. Moreover, as shown in Fig. 6, different values of τ_p , t_p , t_r , and t_f determine different ankle torque patterns. In addition, Zhang *et al.* discovered that a small torque (2 Nm), rather than zero, needs to be enforced at both torque onset and removal to avoid instability caused by torque measurement errors.⁽¹¹⁾ This discovery explains why we have $\tau_1(t_p - t_r) = 2$ and $\tau_2(t_p + t_f) = 2$ in Eq. (16). See the supplementary materials of Ref. 11 for detailed information about modeling the ankle torque pattern during walking.

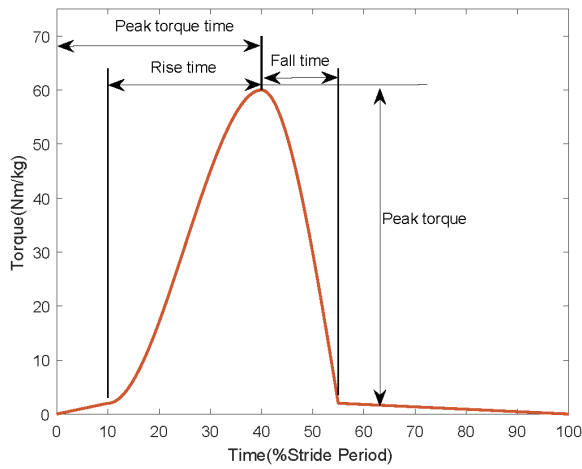


Fig. 5. (Color online) Parameterization of ankle torque during walking.

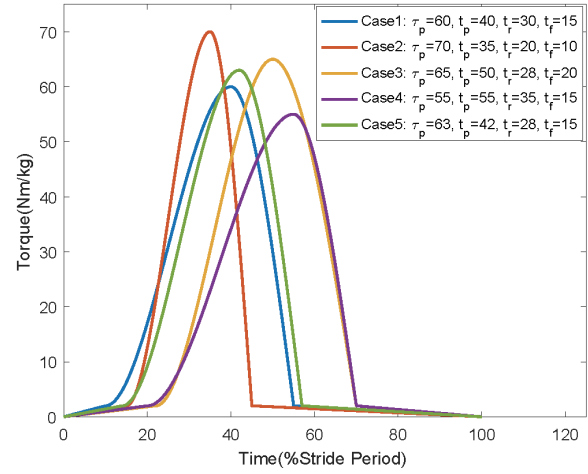


Fig. 6. (Color online) Torque patterns determined for different values of ankle torque control parameters.

4.2 Design of high-level optimizer in the HIL framework

Recall that the high-level optimizer employs ISAPSO to optimize the peak torque, peak time, rise time, and fall time of the exoskeleton in each stride. During the design of the high-level optimizer, similarly to in some previous studies,^(11,22) the root mean square (RMS) of the soleus muscle activity of each subject is used to build the objective function. Thus, the high-level optimization problem can be represented as follows:

$$\begin{cases} \text{Find : } X_m = [\tau_p, t_p, t_r, t_f] \\ \text{Minimize : } J_{ma} = E_{rms_l} + E_{rms_r} \end{cases} \quad (17)$$

subject to

$$\begin{cases} \tau_p^{min} \leq \tau_p \leq \tau_p^{max} \\ t_p^{min} \leq t_p \leq t_p^{max} \\ t_r^{min} \leq t_r \leq t_r^{max} \\ t_f^{min} \leq t_f \leq t_f^{max} \end{cases} \quad (18)$$

Here, X_m is the variable vector to be optimized. τ_p , t_p , t_r , and t_f are the peak torque (in Nm), the peak torque time (in % stride period), the rise time (in % stride period), and the fall time (in % stride period), J_{ma} is the total soleus muscle activity of each user, and E_{rms_l} and E_{rms_r} are the RMS values of the previous soleus EMGs of each user's left and right feet, respectively. The subscripts "min" and "max" in τ_p , t_p , t_r , and t_f respectively indicate the lower and upper bounds

of the corresponding parameter. In this study, we have $\tau_p^{min} = 2$, $\tau_p^{min} = Ms$ (Ms is the body weight of the user), $t_p^{min} = 30\%$, $t_p^{max} = 65\%$, $t_r^{min} = 10\%$, $t_r^{max} = 40\%$, $t_f^{min} = 5\%$, and $t_f^{max} = 20\%$.

Note that the raw soleus EMGs must be filtered, so that noises contained in the muscle can be eliminated from the calculation of the objective function in the high-level optimizer. In this paper, similar to some other studies,^(11,22,23) the raw soleus EMGs of each subject are filtered by a low-pass second-order Butterworth filter, then a high-pass second-order Butterworth filter. We follow the suggestion of Buchanan *et al.*⁽³⁸⁾ of empirically setting the cut-off frequencies of the low-pass and high-pass filters to 4 and 10 Hz, respectively. However, the frequency of the soleus muscle generally ranges from almost zero to hundreds of Hz. The soleus EMGs can still be powerful with a cut-off frequency greater than 10 Hz or smaller than 4 Hz. Thus, it is necessary to further investigate the effect of the cut-off frequencies on the EMG data processing in the near future.

After establishing the optimization model defined by Eqs. (17) and (18), the high-level optimizer then applies ISAPSO to optimize the peak torque τ_p , peak torque time t_p , rise time t_r , and fall time t_f with the goal of minimizing the objective function defined by Eq. (17). During the optimization process, the position vector of each particle in ISAPSO is coded by these four ankle torque pattern parameters. To guarantee that the position vector of each particle satisfies the constraints given by Eq. (18), we use the following saturation mechanism to modify each element of the position vector of each particle in ISAPSO:

$$x_{m_i} = \begin{cases} x_{m_i}^{max}, & \text{if } x_{m_i} > x_{m_i}^{max} \\ x_{m_i}^{min}, & \text{if } x_{m_i} < x_{m_i}^{min} \\ x_{m_i}, & \text{otherwise} \end{cases}, \quad (19)$$

where x_{m_i} denotes the i th ($i = 1, 2, 3, 4$) element in the position vector of particle m , and $x_{m_i}^{max}$ and $x_{m_i}^{min}$ denote the upper and lower bounds of the i th element, respectively. Note that the upper and lower bounds of each element refer to the corresponding torque pattern control parameter constraints depicted above.

During the search, the main loop of ISAPSO is not executed until the iteration number of the swarm reaches the given maximum iteration number. Also, the personal best position ($pbest_m$) of each particle and the global best solution ($gbest$) are substituted by those of the particle only if the fitness value of the particle is less than those of $pbest_m$ and $gbest$. Moreover, the updated $gbest$ is sent to the low-level controller in the HIL optimization framework. Note that the global best solution searched for by ISAPSO is not the global optimum since the proposed algorithm is a first-order convergent algorithm. Thus, the four optimized ankle torque parameters in the high-level optimizer are near optimal. The steps in using ISAPSO to optimize the above parameters are given in Table 1.

4.3 Design of low-level controller in the HIL framework

As demonstrated in Fig. 7, we have developed an ankle exoskeleton to assist human walking. The size of the exoskeleton is designed on the basis of the average foot size of a normal adult.

Table 1

Steps in ISAPSO in the high-level optimizer in the HIL framework.

-
- 1: set the required simulation parameters and initialize the swarm
 - 2: **while** not exist condition **do**
 - 3: **for** each particle m **do**
 - 4: calculate the objective function of the particle by Eq. (17)
 - 5: update the personal best position ($pbest_m$) of the particle
 - 6: update the velocity and position information of the particle by Eq. (2)
 - 7: modify the position vector of the particle by Eq. (19)
 - 8: update the three coefficients of each particle by Eqs. (3)–(9)
 - 9: **end for**
 - 10: update $gbest$ of the swarm and send it to the low-level controller
 - 11: increase the iteration number by 1 and check the exist condition
 - 12: **end while**
-

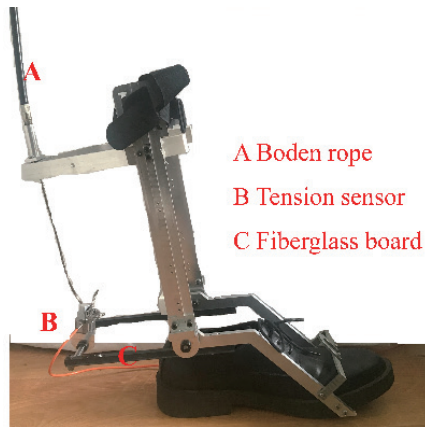


Fig. 7. (Color online) Developed ankle exoskeleton.

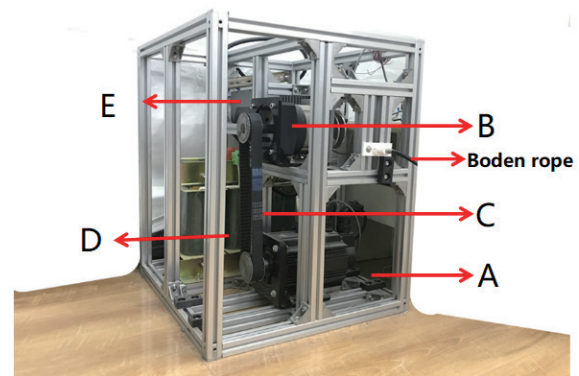


Fig. 8. (Color online) Motor used for the ankle exoskeleton.

The distance between the heel and the toe is 230 mm. The vertical height and horizontal distance between the ankle joint and the heel are both 80 mm. Two high-strength fiberglass boards connected by carbon steel are attached at the joint of the exoskeleton. The exoskeleton is connected to an off-board commercial motor (Maxshine 130MSL10025) by a Boden rope. A tension sensor is connected to the end of the Boden rope.

The designed exoskeleton is driven by a motor controller with a rated power of 2.6 kW and a 5:1 planetary reducer. The motor used for the exoskeleton is shown in Fig. 8, where A, B, C, D, and E respectively denote the servo motor, planetary reducer, pulley, transformer, and motor controller. The motor driver changes its voltage output on the basis of the desired ankle torque. Different motor voltage outputs result in different tensions that the Boden rope transfers to the exoskeleton device, and consequently different assistance torques are provided by the device to the ankle.

Owing to its simplicity and promising control stability, the ILC algorithm is used to update the desired motor torque for the next step in our developed HIL framework as follows:⁽³³⁾

$$\tau_d(t, n+1) = \tau_d(t, n) + k_g(\tau_d(t, n) - \tau_a(t, n)), \quad (20)$$

where n and $n + 1$ denote the current and next steps, $\tau_d(t, n + 1)$ and $\tau_d(t, n)$ denote the desired motor torque at steps $n + 1$ and n steps, and k_g is the constant iterative learning gain and has a value of less than 1, respectively. $\tau_a(t, n)$ denotes the actual torque of the exoskeleton and t is the time index as a percentage of the stride period.

By using Eq. (20), ILC updates the desired motor torque for the next step based on the ankle torque errors of the current step in the HIL framework. The iterative learning gain has profound effects on the performance of ILC. A greater gain accelerates the convergence of ILC, although this will generate a relatively large control error according to the results of our pilot test. Thus, we empirically set the iterative learning gain to 0.5 as a compromise between the convergence rate and tracking performance.

Since we cannot access the proprietary controller used in the Maxshine motor driver, the precise relationship between its input and output remains unknown. However, we can still assume that the input and output of the motor driver have a linear relationship when the motor runs at a moderate and steady speed. Therefore, the proportional coefficient between the input and output of the motor can be obtained using system identification technology. In this study, the motor control in the lower level adopts the following torque control model:

$$v(t) = \frac{\tau_d(t, n+1)}{k_v}, \quad (21)$$

where $v(t)$ is the output voltage of the motor, $\tau_d(t, n + 1)$ is the desired motor torque, and k_v is the proportional coefficient between the input and output of the Maxshine motor. We use $k_v = 12.69$ in this paper, which was obtained using system identification technology. In addition, the sampling and control frequencies of the low-level motor controller are 100 and 3000 Hz, respectively.

In the low-level controller system, the motion of the motor is transferred to the exoskeleton device through the Boden rope to provide assistive torque to the ankle. In this study, the actual ankle torque provided by the exoskeleton is calculated as follows:

$$\tau_a(t, n) = f_b(t)R, \quad (22)$$

where $\tau_a(t, n)$ is the actual ankle torque of the ankle exoskeleton and R is the length of the tension and is determined by the mechanical structure of the ankle exoskeleton ($R = 0.11$ m). $f_b(t)$ denotes the tension of the Boden rope and can be determined as

$$f_b(t) = k_1 v(t), \quad (23)$$

where $v(t)$ is the voltage output of the motor driver and can be obtained from Eq. (21). $k_1 = 0.2686$ is the coefficient between the tension of the Boden rope and the voltage output of the motor driver.

5. Experiments, Results, and Analysis

5.1 Experimental protocol

In this paper, eight healthy subjects (age: 24 ± 1.2 years; mass: 63 ± 5.1 kg; height: 168 ± 3.5 cm, average \pm standard deviation) were recruited to experimentally verify the developed method. All subjects had no prior experience in using the exoskeleton. Prior to the experiments, all subjects provided written informed consent for the experimental study approved by Wuhan University of Technology.

The experimental platform is displayed in Fig. 9 and contains (1) a membrane switch to distinguish the gait cycle of the subject, (2) an ankle exoskeleton worn on each subject's left foot, (3) a motor (Maxshine 130MSL10025), (4) a laptop (ThinkPad E531), (5) two STM32 single chips, (6) an electromyograph (ELONXI EMG 100-Ch-Y-RA) to record muscle EMGs, (7) a force sensor (DYMH-103), and (8) a treadmill (JET300).

For rigorous evaluation, we conducted five tests on each subject under the normal walking model, the zero-torque model, the static assistance condition, and the cases optimized by standard PSO and ISAPSO in our HIL framework. See Ref. 12 for more detailed information about the normal walking model, the zero-torque model, and the static assistance condition. For convenience, the five tests are denoted by Case1 to Case5. To reduce the effect of randomness, the five tests were conducted in random order for each subject. Note that each test lasted 20 min with a 5 min break allowed for each subject between any two tests to avoid muscle fatigue. As described in Sect. 4.2, the soleus muscle activity of each subject was recorded for the final 3 min of each test. We then compared the obtained muscle activity in each test.

For Case4 and Case5, the swarm size and maximum iteration number of standard PSO and ISAPSO were respectively set to 30 and 50 for each subject. The simulation parameters of standard PSO were set to $\omega = 1$ and $c_1 = c_2 = 2.5$ in accordance with the original literature. The simulation parameters of ISAPSO were set to $\omega_{max} = 0.9$, $\omega_{min} = 0.4$, $c_{1s} = c_{2f} = 2$, and $c_{1f} = c_{2s} = 0.1$ on the basis of the analysis results in Sect. 3.4.

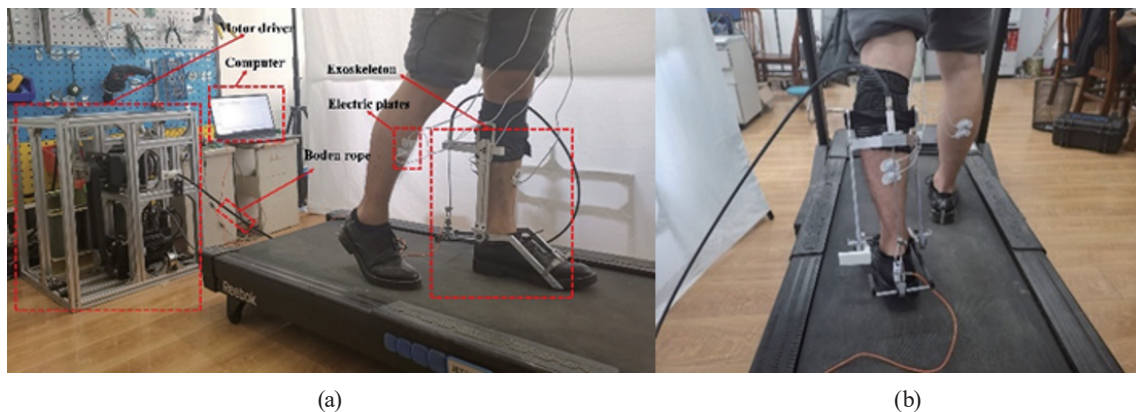


Fig. 9. (Color online) HIL experimental platform: (a) side view and (b) back view.

5.2 Experimental results and analysis

Table 2 shows the numerical results of each test for each subject, where the best results among the five tests for each subject are highlighted in bold. Note that the obtained torque pattern control parameters under Case1 and Case2 are represented by “NA” in Table 2 due to the fact that the exoskeleton provides no assistance to each subject for these two cases. The average muscle activity and the soleus EMGs of each subject are visualized in Figs. 10 and 11,

Table 2
Results for each subject under different test conditions.

Subject	Condition	Obtained [τ_p , t_p , t_r , t_f]	Muscle activity of soleus		
		[Nm, %Stride, %Stride, %Stride]	Left foot	Right foot	Total
1	Case1: Normal walking	NA	0.3019	0.3045	0.6064
	Case2: Zero torque	NA	0.3497	0.3358	0.6855
	Case3: Static assistance	[20.25, 56.13, 19.32, 18.99]	0.2583	0.2690	0.5273
	Case4: Optimized by standard PSO	[15.35, 61.01, 30.36, 19.26]	0.2308	0.2284	0.4592
	Case5: Optimized by ISAPSO	[12.53, 58.42, 28.53, 17.54]	0.2147	0.2053	0.4220
2	Case1: Normal walking	NA	0.1917	0.1799	0.3716
	Case2: Zero torque	NA	0.2488	0.2101	0.4589
	Case3: Static assistance	[18.35, 60.45, 21.21, 16.78]	0.1963	0.1570	0.3533
	Case4: Optimized by standard PSO	[14.58, 61.65, 17.73, 15.44]	0.1783	0.1234	0.3017
	Case5: Optimized by ISAPSO	[15.35, 63.54, 20.45, 11.35]	0.1621	0.1112	0.2733
3	Case1: Normal walking	NA	0.2285	0.2673	0.4958
	Case2: Zero torque	NA	0.2645	0.3377	0.6022
	Case3: Static assistance	[19.56, 62.91, 24.46, 15.31]	0.1921	0.2631	0.4552
	Case4: Optimized by standard PSO	[18.80, 57.53, 25.10, 12.25]	0.1714	0.2504	0.4218
	Case5: Optimized by ISAPSO	[17.54, 61.53, 22.54, 14.43]	0.1643	0.2395	0.4038
4	Case1: Normal walking	NA	0.3541	0.3284	0.6825
	Case2: Zero torque	NA	0.3823	0.3528	0.7351
	Case3: Static assistance	[19.25, 62.03, 19.47, 20.38]	0.3141	0.3183	0.6324
	Case4: Optimized by standard PSO	[16.58, 61.94, 17.38, 19.57]	0.3005	0.2963	0.5968
	Case5: Optimized by ISAPSO	[18.42, 57.04, 12.94, 17.52]	0.2862	0.2816	0.5678
5	Case1: Normal walking	NA	0.2853	0.2963	0.5816
	Case2: Zero torque	NA	0.3274	0.3173	0.6447
	Case3: Static assistance	[19.56, 71.37, 20.47, 19.53]	0.2953	0.2734	0.5687
	Case4: Optimized by standard PSO	[17.58, 59.64, 21.02, 17.45]	0.2483	0.2531	0.5014
	Case5: Optimized by ISAPSO	[13.76, 62.92, 20.53, 18.53]	0.2301	0.2396	0.4597
6	Case1: Normal walking	NA	0.3207	0.3421	0.6628
	Case2: Zero torque	NA	0.3486	0.3702	0.7188
	Case3: Static assistance	[20.48, 62.73, 22.67, 20.48]	0.3194	0.3012	0.6206
	Case4: Optimized by standard PSO	[19.57, 61.45, 22.47, 18.04]	0.2704	0.2993	0.5697
	Case5: Optimized by ISAPSO	[17.47, 63.04, 19.87, 17.94]	0.2563	0.2801	0.5364
7	Case1: Normal walking	NA	0.2563	0.2751	0.5314
	Case2: Zero torque	NA	0.2753	0.3173	0.5926
	Case3: Static assistance	[17.91, 61.36, 25.73, 19.31]	0.2474	0.2845	0.5319
	Case4: Optimized by standard PSO	[15.49, 62.94, 24.57, 16.46]	0.2193	0.2283	0.4476
	Case5: Optimized by ISAPSO	[16.58, 56.28, 19.58, 15.40]	0.1946	0.2118	0.4064
8	Case1: Normal walking	NA	0.2080	0.2410	0.4490
	Case2: Zero torque	NA	0.2337	0.2963	0.5300
	Case3: Static assistance	[21.56, 64.45, 26.89, 19.11]	0.2235	0.2766	0.5001
	Case4: Optimized by standard PSO	[19.37, 63.38, 25.46, 16.29]	0.1874	0.2246	0.4120
	Case5: Optimized by ISAPSO	[16.54, 57.43, 24.34, 17.53]	0.1692	0.1853	0.3545

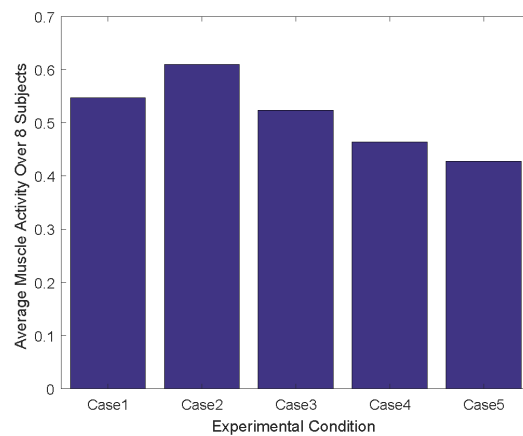


Fig. 10. (Color online) Average muscle activity of eight subjects in different tests.

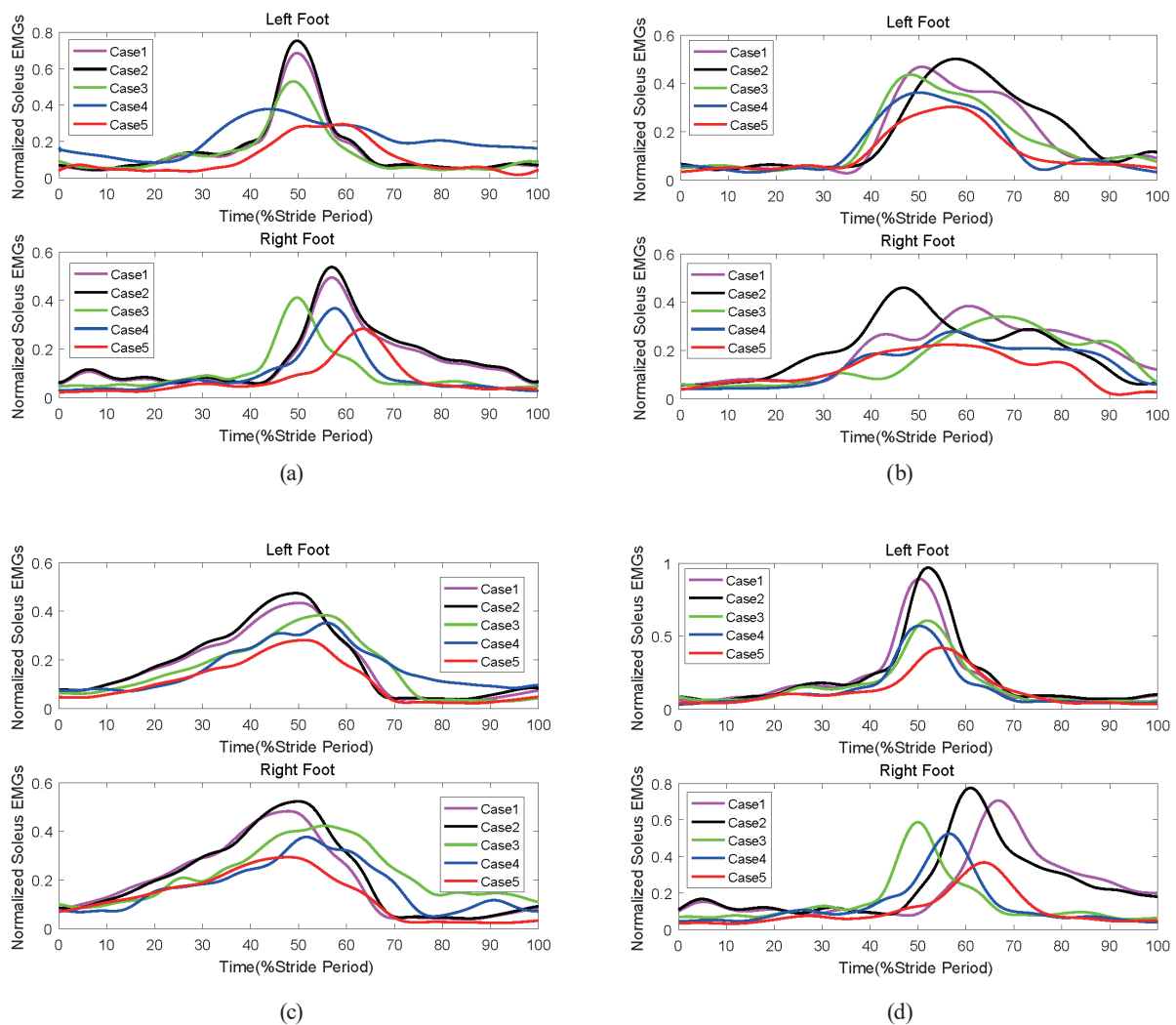


Fig. 11. (Color online) Obtained soleus EMGs of each subject's feet under each test condition. (a)–(d) correspond to subjects 1–4, respectively.

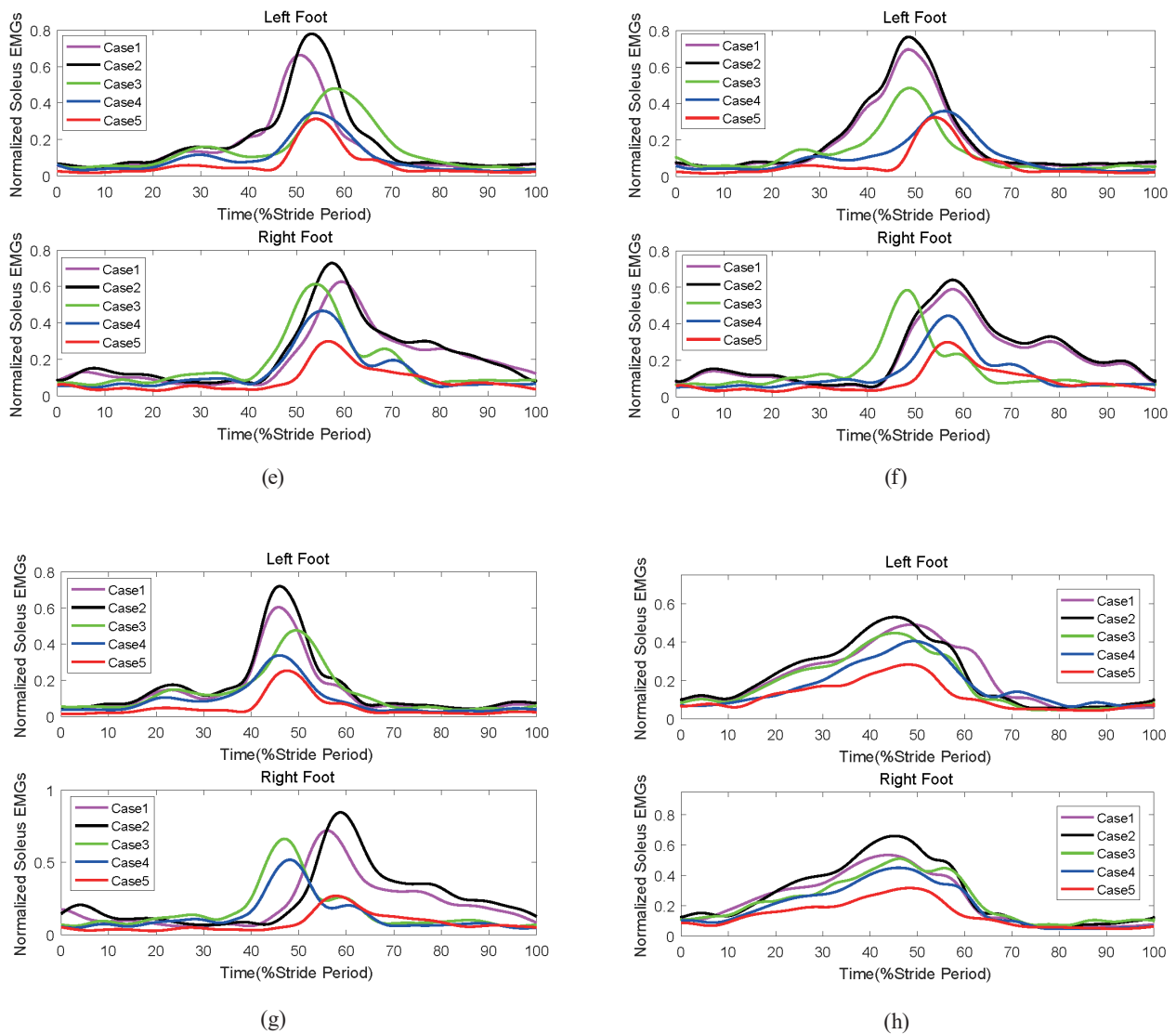


Fig. 11. (Continued) (Color online) Obtained soleus EMGs of each subject's feet under each test condition. (e)–(h) correspond to subjects 5–8, respectively.

respectively. Figure 12 shows the fitness value curves optimized by standard PSO and ISAPSO for each subject. Table 3 shows the percentage improvements in total soleus muscle activity of the eight subjects for Case5 compared with the other four cases.

It can be clearly observed from Table 2 that Case5 has the lowest muscle activity of each subject, followed by Case4, Case3, Case1, and Case2. Also, from Fig. 10, it is evident that Case5 outperforms the other four cases in terms of the average muscle activity for the eight subjects. These two observations indicate that our proposed method has superior performance to the other methods considered in this paper. It is also evident from Fig. 11 that Case5 can significantly reduce the magnitude of the soleus EMGs of each subject among the five considered cases. This also suggests the effectiveness and superiority of our proposed method.

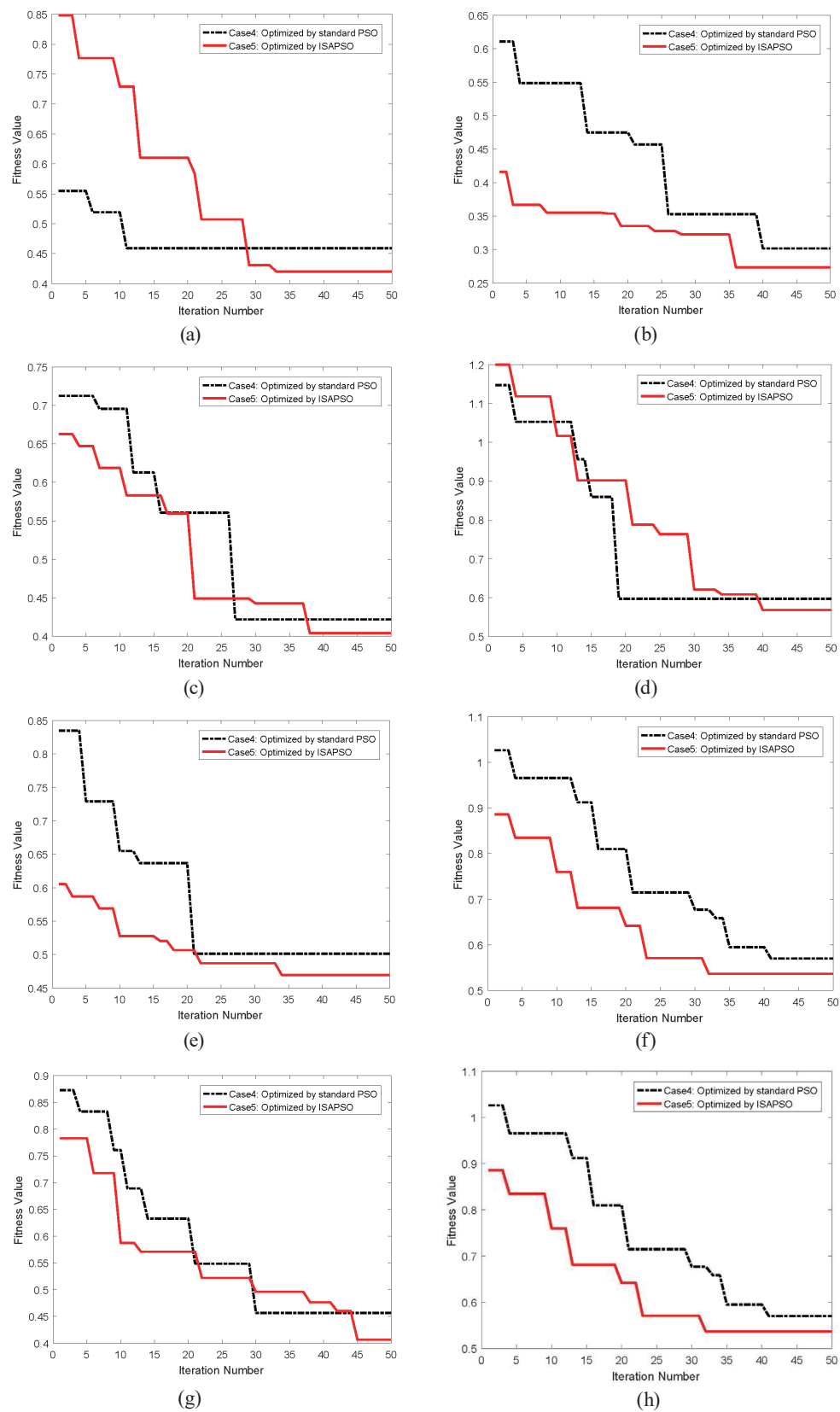


Fig. 12. (Color online) Fitness value curves optimized by standard PSO and ISAPSO for different subjects. (a)–(h) correspond to Subjects 1–8, respectively.

Table 3

Total soleus muscle activity improvements over eight subjects in Case5 compared with the other cases (in %).

	Case5 to Case1	Case5 to Case2	Case5 to Case3	Case5 to Case4
Best	44.38	67.91	41.07	16.22
Worst	20.20	29.46	11.37	4.46
Average	28.52	47.04	23.46	8.58
Std.	89.14	13.54	10.21	3.82

Moreover, it can be seen from Fig. 12 that, compared with those for Case4, the fitness value curves under Case5 for most of the eight subjects continue to fall in the later stage of the evolution. This may imply that, compared with standard PSO, the proposed ISAPSO has a better trade-off between global and local search abilities. This can probably be explained by the fact that the self-adaptive parameter updating strategy proposed in ISAPSO can better balance the global and local search abilities. This can reflect the effectiveness of the self-adaptive parameter updating strategy proposed in ISAPSO to some extent.

Note that the above analysis can only show the general efficacy and superiority of the proposed method. The concrete differences between our proposed method and the other four methods for the compared cases still remain unknown. To this end, a paired t-test was conducted on the statistical results in Table 3 to concretely identify the differences between Case5 and the other four cases. Since eight subjects performed five tests in this study, the t-test analysis was executed with $N = 8$ at a confidence level of 95%. It was found that compared with Case4, Case 3, Case2, and Case1, Case5 had average reductions of 8.58 ± 3.82 , 23.46 ± 10.21 , 47.04 ± 13.54 , and $28.52 \pm 8.14\%$ (mean \pm standard deviation), respectively, in the average total soleus muscle activity of the eight subjects at a confidence level of 95%.

The results of our analysis allow us to conclude that our proposed method significantly outperforms the other methods compared in this paper. Thus, the proposed method can be considered as an effective alternative in HIL optimization for the ankle exoskeleton during human walking.

6. Conclusions and Future Work

With the goal of enhancing human walking mobility, we designed an HIL optimization control framework for an ankle exoskeleton by employing a newly developed ISAPSO algorithm and the ILC algorithm. To obtain a good balance between the global and local search abilities of ISAPSO, we proposed a self-adaptive parameter updating strategy to adjust the three key coefficients of the particles in the PSO optimizer. Moreover, the convergence and a convergence-guaranteed parameter setting rule concerning ISAPSO were derived. Then, an HIL optimization framework was established by using ISAPSO and the ILC algorithm to individually and automatically identify the assistive strategies that the exoskeleton provides to each subject. The experimental results show that the proposed method can significantly reduce the soleus muscle activity of each subject compared with the other four cases considered.

The results and methods reported in this paper raise several issues that deserve future study. Firstly, the convergence speed of ISAPSO needs to be further improved so as to avoid muscle

fatigue of the experimenter. Secondly, although the objective function determined by the soleus muscle activity implemented in this paper is a commonly accepted index, we are still considering the possibility of mixing soleus EMGs with other muscle EMGs, such as those of the main muscles of the human hip, to establish a more comprehensive objective function. Thirdly, despite confirming the efficiency and superiority of the proposed method by comparing its performance with those of the normal walking model, static assistance condition, zero-torque model, and the case optimized by standard PSO, its performance should also be compared with those of some other well-established EAs. Finally, we are also considering the possibility of developing more advanced bionic controlling algorithms to design the low-level controller in a forthcoming study to further reduce the walking cost consumed by the user.

Acknowledgments

We would like to thank the editor for his efforts in coordinating the review process. The anonymous reviewers are sincerely appreciated for their useful and valuable comments. This work was financially supported by National Science Foundation of China under Grant Nos. 61903286 and 52075530, and by the AiBle project co-financed by the European Regional Development Fund. The authors declare no conflicts of interest regarding this work.

References

- 1 A. Agrawal, O. Harib, A. Hereid, S. Finet, M. Masselin, L. Praly, A. D. Ames, K. Sreenath, and J. W. Grizzle: IEEE Access **5** (2017) 99. <https://doi.org/10.1109/ACCESS.2017.2690407>
- 2 C. Bayo'n, O. Ram'irez, J. I. Serrano, M. D. Castillo, A. Pe'rez-Somarriba, J. M. Belda-Lois, I. Mart'inez-Caballero, S. Lerma-Lara, C. Cifuentes, A. Frizera, and E. Rocon: Rob. Auton. Syst. **91** (2017) 101. <https://doi.org/10.1016/j.robot.2016.12.015>
- 3 Y. Hua, J. Fan, G. Liu, X. Zhang, J. I. E. Zhao, and Y. Zhu: IEEE Access **7** (2019) 1. <https://doi.org/10.1109/ACCESS.2019.2904709>
- 4 W. Dm and D. Wei: IEEE Access **7** (2019) 72343. <https://doi.org/10.1109/ACCESS.2019.2918721>
- 5 L. Mooney and H. Herr: J. Neuroeng. Rehabil. **1** (2016) 1. <https://doi.org/10.1186/s12984-016-0111-3>
- 6 F. Panizzolo, I. Galiana A. T. Asbeck, C. Sivi, K. Schmidt, K. Holt, and C. Walsh: J. Neuroeng. Rehabil. **13** (2016). <https://doi.org/10.1186/s12984-016-0150-9>
- 7 K. Yin, K. Xiang, M. Pang, J. Chen, P. Anderson, L. Yang: IEEE Access **7** (2019) 7222. <https://doi.org/10.1109/ACCESS.2019.2920134>
- 8 Y. Ding, F. Panizzolo, C. Sivi, P. Malcolm, I. Galiana, K. Holt, and C. Walsh: J. Neuroeng. Rehabil. **13** (2016) 1. <https://doi.org/10.1186/s12984-016-0196-8>
- 9 K. Seo, J. Lee, and Y. Park: IEEE Int. Conf. Rehabil. Robot. (IEEE, 2017) 246–251.
- 10 M. Kim, Y. Ying, P. Malcolm, J. Speeckaert, C. Sivi, C. Walsh J, and S. Kuindersma: PLoS One **12** (2019). <https://doi.org/10.1371/journal.pone.0184054>
- 11 J. Zhang, P. Fiers, K. Witte, R. Jackson, K. Poggensee, C. Atkeson, and S. Collin: Sci. Rob. **1284** (2017) 1280. <https://doi.org/10.1126/science.aal5054>
- 12 Y. Ding, M. Kim, S. Kuindersma, and C. Walsh: Sci. Rob. **3** (2018) 1. <https://doi.org/10.1126/scirobotics.aar5438>
- 13 B. Quinlivan, L. Sangjun, P. Malcolm, D. Rossi, M. Grimmer, C. Sivi, and C. Walsh: Sci. Rob. **4416** (2016) 1. <https://doi.org/10.1126/scirobotics.aah4416>
- 14 J. Koller, D. Gates, D. Ferris, and C. Remy: J. Appl. Physiol. **122** (2016) 242. <https://doi.org/10.1152/japplphysiol.00710>
- 15 S. Lee, J. Kim, L. Baker, A. Long, N. Karavas, N. Menard, I. Galiana, and C. Walsh: J. Neuroeng. Rehabil. **4** (2018). <https://doi.org/10.1186/s12984-018-0410-y>
- 16 B. Quinlivan, L. Sangjun, P. Malcolm, D. Rossi, M. Grimmer, C. Sivi, and C. Walsh: Sci. Rob. **4416** (2016) 1. <https://doi.org/10.1126/scirobotics.aah4416>

- 17 K. Seo, K. Kim, Y. Park, J. Cho, J. Lee, B. Choi, B. Lim, Y. Lee, and Y. Shim: 2018 IEEE Int. Conf. Robot. Autom. (2018 IEEE) 6752–6758
- 18 M. Hamaya, T. Matsubara, T. Noda, T. Teramae, and J. Morimoto: Pattern Recognit. Lett. **99** (2017) 67. <https://doi.org/10.1016/j.patrec.2017.04.007>
- 19 X. Zhang, W. Jiang, Z. Li, and S. Song: Eur. J. Control **50** (2019) 198. <https://doi.org/10.1016/j.ejcon.2019.06.003>
- 20 I. Kang, H. Hsu, and A. Young: IEEE Rob. Autom. Lett. **4** (2019) 430. <https://doi.org/10.1109/lra.2019.2890896>
- 21 H. Yan, B. Tang, and K. Xiang: 2019 IEEE 4th Int. Conf. Advanced Robotics and Mechatro (IEEE, 2019).
- 22 W. Wang, J. Chen, and Y. Ji: IEEE Trans. Ind. Inf. **99** (2020) 1. <https://doi.org/10.1109/TII.2020.2974232>
- 23 P. Malcolm, S. Galle, and D. Clercq: Sci. Rob. **356** (2017) 1230. <https://doi.org/10.1126/science.aan5367>
- 24 W. Felt, J. Selinger, J. Donelan, and C. Remy: PLoS One **10** (2015) 1. <https://doi.org/10.1371/journal.pone.0135342>
- 25 R. Koller, H. Gates, P. Ferri, and C. David: Robot. Sci. Syst. **2** (2016). <https://doi.org/10.15607/rss.2016.xii.007>
- 26 K. Yin, M. Pang, K. Xiang, and C. Jing: In: Proc. 2018 IEEE 7th Data Driven Control Learn. Syst. Conf. (IEEE, 2018) 175–179.
- 27 Y. Long, Z. Du, W. Wang, and W. Dong: Appl. Bionics. Biomech. **2016** (2016) 1. <https://doi.org/10.1155/2016/5017381>
- 28 A. Belkadi, H. Oulhadj, Y. Touati, S. Khan, and B. Daachi: Appl. Soft Comput. J. **60** (2017) 87. <https://doi.org/10.1016/j.asoc.2017.06.012>
- 29 A. Majeed, Z. Taha, A. Abidin, M. Zakaria, I. Khairuddina, M. Razman, and Z. Mohamed: Procedia Comput. Sci. **105** (2016) 183. <https://doi.org/10.1016/j.procs.2017.01.204>
- 30 M. Hasanipanah, D. Armaghani, H. Bakhshandeh, M. Majid, and M. Tahir: Neural. Comput. Appl. **28** (2017) 1043. <https://doi.org/10.1007/s00521-016-2434-1>
- 31 Z. Ma, Y. Dong, C. Wang, and X. Shao: IEEE Access **6** (2018) 1. <https://doi.org/10.1109/ACCESS.2018.2841411>
- 32 B. Tang, Z. Zhu, H. Shin, A. Tsourdos, and J. Luo: Inf. Sci. **420** (2017) 364. <https://doi.org/10.1016/j.ins.2017.08.076>
- 33 B. Tang, Z. Zhu, and J. Luo: Int. J. Adv. Robot Syst. **13** (2016) 1. <https://doi.org/10.5772/63812>
- 34 X. Tu, X. Zhou, J. Li, C. Su, X. Sun, H. Han, X. Jiang, and J. He: Cluster Comput. **20** (2017) 2855. <https://doi.org/10.1007/s10586-017-0880-x>
- 35 J. Zhang, C. Cheah, and S. Collin: Proc. IEEE Int. Conf. Robot Autom. (IEEE, 2015) 5584–5589.
- 36 R. Eberhart and J. Kennedy: MHS'95 Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (IEEE, 1995) 39–43.
- 37 R. Akbari and K. Ziarati: J. Comput. Appl. Math **235** (2011) 2694. <https://doi.org/10.1016/j.cam.2010.11.01>
- 38 T. Buchanan, L. loyd, K. Manal, and T. Besier: J. Appl. Biomech. **20** (2004) 367. <https://doi.org/10.1109/TMI.2004.835602>

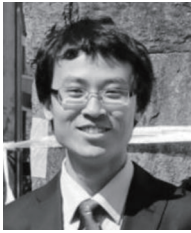
About the Authors



Jinfeng Wang received her B.E. degree from the Department of Physics, Wenzhou Normal University, Zhejiang, China, in 2002, and her M.D. degree from Zhejiang Normal University, Jinhua, Zhejiang, China, in 2008. She is currently a lecturer with the Department of Information, Wuhan Huaxia University of Technology. Her research interests include human motion detection and human–robot interaction. (jinfeng_wang@126.com)



Biwei Tang received his Ph.D. degree in aircraft design from the School of Astronautics, Northwestern Polytechnical University, China, in 2018. He is now a lecturer in the School of Automation, Wuhan University of Technology, China. His current research interests include the optimization and control of human–exoskeleton interaction. (tangbiwei@whut.edu.cn)



Muye Pang received his Ph.D. degree in intelligent mechanical engineering from Kagawa University, Kagawa, Japan, in 2015. He joined Wuhan University of Technology, Wuhan, China, as an assistant professor in 2015. His research interests include bipedal robots, push recovery, muscle reflexes, and surface electromyographic signals. (pangmuye@whut.edu.cn)



Kui Xiang received his B.E. degree in mechanical manufacture from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1999 and his M.E. degree in mechanical dynamics and his Ph.D. degree in system sciences from Zhejiang University, Hangzhou, China, in 2002 and 2006, respectively. His current research interests include human motion signal analysis, exoskeleton design and control, modeling, and statistical learning. (xkarcher@126.com)



Zhaojie Ju received his Ph.D. degree in intelligent robotics from the University of Portsmouth, UK, in 2010. His research interests include machine intelligence, pattern recognition, and their applications to human motion analysis, multifingered robotic hand control, human–robot interaction and collaboration, and robot skill learning. (zhaojie.ju@port.ac.uk)

Appendix: Convergence investigation of ISAPSO

As a stochastic algorithm, the convergence of PSO is of great importance because it profoundly affects the optimization performance of PSO in different situations. To this end, the convergence of the proposed ISAPSO is investigated in this subsection. Because the dimensions in the velocity and position vectors of the particles in this algorithm are independent, the motion rule given by Eq. (2) can be rewritten as the following 1D dynamic system:

$$\begin{pmatrix} X_m(k+1) \\ V_m(k+1) \end{pmatrix} = \begin{pmatrix} 1-c & \omega_m \\ -c & \omega_m \end{pmatrix} \begin{pmatrix} X_m(k) \\ V_m(k) \end{pmatrix} + \begin{pmatrix} c \\ c \end{pmatrix} P_e, \quad (24)$$

where

$$c = c_{1m}r_1 + c_{2m}r_2, \quad (25)$$

$$P_e = \frac{c_{1m}r_1 \cdot pbest + c_{2m}r_2 \cdot gbest}{c_{1m}r_1 + c_{2m}r_2}. \quad (26)$$

Let $\lambda_{1,2}$ be the characteristic roots of the dynamic system represented by Eq. (24). Then, we can readily obtain the characteristic equation and roots to this system as

$$\lambda^2 - (1 + \omega_m - c)\lambda + \omega_m = 0, \quad (27)$$

$$\lambda_{1,2} = \frac{1 + \omega_m - c \pm \sqrt{(1 + \omega_m - c)^2 - 4\omega_m}}{2}. \quad (28)$$

Clearly, the dynamic system represented by Eq. (23) converges if and only if⁽³³⁾

$$\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1. \quad (29)$$

Since the characteristic roots $\lambda_{1,2}$ can be real or complex, both cases are discussed in the following.

(a) When $\lambda_{1,2}$ are complex, namely, $\lambda_{1,2} \in \mathbb{C}$, where \mathbb{C} is the imaginary domain.

Lemma 2. For the dynamic system defined by Eq. (24), $\lambda_{1,2} \in \mathbb{C}$ if and only if

$$\begin{cases} 1 + \omega_m - 2\sqrt{\omega_m} < c < 1 + \omega_m + 2\sqrt{\omega_m} \\ \omega_m \geq 0 \end{cases}. \quad (30)$$

Proof. It is clear from the characteristic equation given by Eq. (27) that $\lambda_{1,2}$ are two complex roots if and only if

$$(1 + \omega_m - c)^2 - 4\omega_m < 0. \quad (31)$$

Lemma 2 is easily proved by expanding Eq. (31). ■

Lemma 3. In the case where $\lambda_{1,2} \in \mathbb{C}$, the dynamic system given by Eq. (24) converges if and only if

$$\begin{cases} 1 + \omega_m - 2\sqrt{\omega_m} < c < 1 + \omega_m + 2\sqrt{\omega_m} \\ 0 \leq \omega_m < 1 \end{cases}. \quad (32)$$

Proof. From Eq. (28), it is clear that, for $\lambda_{1,2} \in \mathbb{C}$, we have

$$\text{Max}\{|\lambda_1|, |\lambda_2|\} = |\lambda_1| = |\lambda_2| = \sqrt{\omega_m}. \quad (33)$$

Thus, in such a case, $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$ holds if and only if

$$\sqrt{\omega_m} < 1. \quad (34)$$

Considering the conditions given by **Lemma 2** and the fact that $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$, it can be easily concluded that the dynamic system expressed by Eq. (24) converges under the situation where $\lambda_{1,2} \in \mathbb{C}$ if and only if

$$\begin{cases} 1 + \omega_m - 2\sqrt{\omega_m} < c < 1 + \omega_m + 2\sqrt{\omega_m} \\ 0 \leq \omega_m < 1 \end{cases}. \quad (35)$$

This completes the proof of **Lemma 3**. ■

(b) When $\lambda_{1,2}$ are real, namely, $\lambda_{1,2} \in \mathbb{R}$.

Lemma 4. For the dynamic system given by Eq. (24), $\lambda_{1,2}$ are two real roots if and only if

$$\begin{cases} c \in \mathbb{R}, \omega_m < 0 \\ c \leq 1 + \omega_m - 2\sqrt{\omega_m} \text{ or } c \leq 1 + \omega_m + 2\sqrt{\omega_m}, \omega_m \geq 0 \end{cases}. \quad (36)$$

Proof. It is evident from Eq. (27) that $\lambda_{1,2}$ are two real roots if and only if

$$(1 + \omega_m - c)^2 - 4\omega_m \geq 0. \quad (37)$$

Lemma 4 is easily proved by expanding Eq. (37). ■

Lemma 5. For any $\lambda_{1,2} \in \mathbb{R}$, the dynamic system given by Eq. (24) converges if and only if

$$\begin{cases} 0 \leq c < 2\omega_m + 2, -1 < \omega_m < 0 \\ 0 < c \leq 1 + \omega_m - 2\sqrt{\omega_m} \text{ or } 1 + \omega_m + 2\sqrt{\omega_m} < c < 2\omega_m + 2, 0 \leq \omega_m \leq 1 \end{cases}. \quad (38)$$

Proof. For any $\lambda_{1,2} \in \mathbb{R}$, it is trivial from Eqs. (28) and (29) that $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$ if and only if

$$-1 < \frac{1 + \omega_m - c \pm \sqrt{(1 + \omega_m - c)^2 - 4\omega_m}}{2} < 1. \quad (39)$$

It is clear that, for $\lambda_{1,2} \in \mathbb{R}$, Eq. (39) can be rewritten as

$$c - \omega_m - 3 < \pm \sqrt{(1 + \omega_m - c)^2 - 4\omega_m} < c - \omega_m + 1. \quad (40)$$

By expanding Eq. (27), we find that $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$ holds in the case that $\lambda_{1,2} \in \mathbb{R}$ if and only if

$$\begin{cases} 2\omega_m + 2 - c > 0 \\ c > 0 \end{cases}. \quad (41)$$

Simultaneously considering the conditions given by **Lemma 4** and $\text{Max}\{|\lambda_1|, |\lambda_2|\} < 1$ given by Eq. (41), it can be easily proven that **Lemma 5** is satisfied when $\lambda_{1,2} \in \mathbb{R}$. ■

Lemma 6. The dynamic system given by Eq. (24) converges in any domain if and only if

$$\begin{cases} 0 < c_{1m}r_1 + c_{2m}r_2 < 2\omega_m + 2 \\ -1 < \omega_m < 1 \end{cases}. \quad (42)$$

Proof. Combining the conditions given by **Lemma 3** and **Lemma 5**, it is trivial that the dynamic system given by Eq. (24) converges in any domain if and only if

$$\begin{cases} 0 < c < 2\omega_m + 2 \\ -1 < \omega_m < 1 \end{cases}. \quad (43)$$

Substituting Eq. (24) into Eq. (42) completes the proof of **Lemma 6**. ■

The convergence domain of the dynamic system given by Eq. (24), i.e., the proposed ISAPSO, is displayed in Fig. A1. The convergence of this algorithm is necessarily and sufficiently guaranteed if any pair of selected control parameters ω_m and c ($c = c_{1m}r_1 + c_{2m}r_2$) of each particle is located within the domain shown in Fig. A1(b).

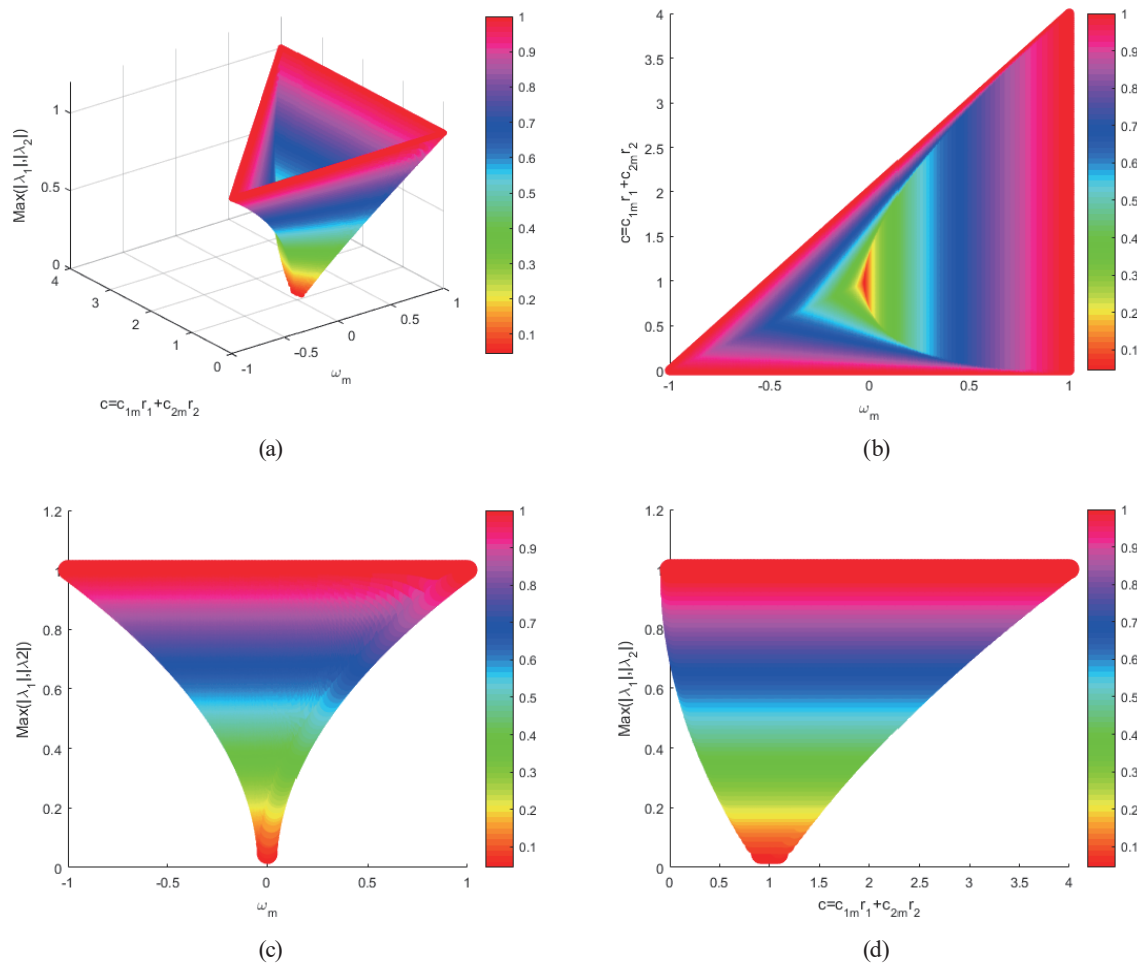


Fig. A1. (Color online) Real convergence domain of ISAPSO: (a) 3D convergence domain, (b) 2D projection of ω_m and c , (c) 2D projection of ω_m and $\text{Max}\{|\lambda_1|, |\lambda_2|\}$, and (d) 2D projection of c and $\text{Max}\{|\lambda_1|, |\lambda_2|\}$.