CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Gait multi-objectives optimization of lower limb exoskeleton robot based on BSO-EOLLFF algorithm

Peng Zhang[1,2], Junxia Zhang[1,2,*] (ID) and Ahmed Elsabbagh[3]

[1]Tianjin University of Science and Technology, Tianjin, China, [2]Tianjin Key Laboratory for Integrated Design and Online Monitor Center of Light Design and Food Engineering Machinery Equipment, Tianjin, China, and [3]School of Design and Production Engineering, Ain Shams University, Cairo, Egypt
*Corresponding author. E-mail: zjx@tust.edu.cn

Abstract
Aiming at problems of low optimization accuracy and slow convergence speed in the gait optimization algorithm of lower limb exoskeleton robot, a novel gait multi-objectives optimization strategy based on beetle swarm optimization (BSO)-elite opposition-based learning (EOL) levy flight foraging (LFF) algorithm was proposed. In order to avoid the algorithm from falling into the local optimum, the EOL strategy with global search capability, the LFF strategy with local search capability and the dynamic mutation strategy with high population diversity were introduced to improve optimization performance. The optimization was performed by establishing a multi-objectives optimization function with the robot's gait zero moment point (ZMP) stability margin and driving energy consumption. The joint comparative tests were carried out in SolidWorks, ADAMS and MATLAB software. The simulation results showed that compared with the particle swarm optimization algorithm and the BSO algorithm, the ZMP stability margin obtained by the BSO-EOLLFF algorithm was increased, and the average driving energy consumption was reduced by 25.82% and 17.26%, respectively. The human-machine experiments were conducted to verify the effectiveness and superiority. The robot could realize stable and smooth walking with less energy consumption. This research will provide support for the application of exoskeleton robot.

1. Introduction

The lower extremity exoskeleton robot could assist patients with mobility impairment [1]. Compared with the traditional artificial rehabilitation training mode, rehabilitation robots have many advantages: rehabilitation robots are more suitable for performing long-term simple and repetitive exercises; rehabilitation robots have programmable capabilities to provide personalized training [2, 3]. With the development of research, there have been many mature lower limb rehabilitation robots, such as Rewalk [4], Mina exoskeleton robot [5], MindWalker [6], and Vanderbilt [7]. Since the exoskeleton robot drives wearers to move, the gait trajectory of the robot determines whether the wearer's rehabilitation training is practical [8]. Therefore, maintaining a stable and smooth gait trajectory is one of the critical technologies in rehabilitation robots.

Patients with lower limb movement disorder have a weak ability to control their balance during walking [9]. However, it is essential to ensure walking stability. Vukobratovic proposed the zero moment point (ZMP) criterion in 1972 [10] and pointed out that when ZMP was located within the robot support polygon, the robot would not tip over. Researchers used the measured human ZMP as the reference and controlled the ZMP of the exoskeleton robot to achieve stable walking [11–14]. However, when the ZMP of the exoskeleton robot is located at the edge of the support polygon, the body with the exoskeleton robot will tilt, but it is still possible to achieve stable walking. This means that the ZMP criterion is not necessary for stable walking, which is more suitable for a quasi-static situation. Wu introduced a

ZMP dynamic stability evaluation method and tracking control algorithm to improve the stability of the coronal plane and sagittal plane [15]. Feng proposed a dynamic optimization control method based on ZMP, which used information about the ZMP position [16]. Literature [17] detected the state changes of the robot's center-of-mass (CoM) velocity and acceleration and mapped the error between the measured ZMP and the predicted ZMP for the foothold adjustment. The linear predictive control strategy was used to achieve foothold adjustment, but only adjusting the foothold has limited gait stability.

The literature [18] proposed an improved deep reinforcement learning algorithm based on the robot inverted pendulum modeling method, which could effectively deal with the stability of the high-dimensional state action space generated by robot walking while solving problems of large computation and high storage requirements caused by too many robot parameters. Still, the algorithm simply conducted the action space discretization when optimizing related problems with continuous domains, leading to a partial loss of action information. The literature [19] solved the value of each joint variable corresponding to each center of gravity position point by genetic algorithm based on the robot center of gravity constraint and geometric constraint methods. The robot center of gravity trajectory found by this method and the target center of gravity trajectory coincide, and the gait trajectory with the maximum stability margin was obtained. Still, the algorithm cannot use the network's feedback information in time, which leads to a relatively slow search speed, and if a more accurate solution was to be obtained, more training time was needed. The literature [20] combined virtual constraints and hybrid zero-dynamic design to optimize stable gaits. Still, the ideal virtual completed constraints cannot be obtained simply and must use the robot's accurate kinematic-dynamic mathematical model.

The goals of exoskeleton robot walking control are, first, to ensure the stability of walking; second, to make the gait conform to given conditions. When the stability index of the actual trajectory is consistent with the reference trajectory, the motion state of the robot will conform to the goal of walking control. Therefore, the goal could be described as reducing the error of stability index between the actual trajectory and the reference trajectory with less energy consumption. The main contributions of the paper are:

1. The impact of the robot ZMP on the gait stability was analyzed through robot dynamics modeling, and a novel stability criterion was proposed to keep walking stability.
2. A multi-objectives optimization function was established with the gait ZMP stability margin and the driving energy consumption to realize stable walking with less energy consumption.
3. The beetle swarm optimization (BSO) algorithm can overcome the problem of slow convergence. However, the BSO algorithm was easy to fall into the local optimum. The elite opposition-based learning (EOL) strategy, the levy flight foraging (LFF) strategy and the dynamic mutation strategy were introduced to overcome the problems and improve the optimization performance.

The rest of the paper is organized as follows: Section 2 describes exoskeleton robot system. Section 3 illustrates the gait multi-objectives function design. The parameter optimization based on BSO-EOLLFF algorithm is presented in Section 4. Section 5 describes simulation experiment results and analysis. Section 6 describes the human-machine experiment and result analysis. Finally, the conclusion of this work is given in Section 7.

2. Exoskeleton robot system

2.1. Exoskeleton robot

The developed lower limb exoskeleton is shown in Fig. 1. The exoskeleton robot designed in this paper mainly considers the motion in the sagittal plane. For each leg, it has one active degree of freedom (DoF) for the hip joint and knee joint, respectively, and one passive DoF for the ankle joint. The ankle joint has a high structural complexity, and it has an important influence on human walking stability [30–32]. Most of existing exoskeleton robots are designed with passive DoF ankle joint, such as Rewalk robot
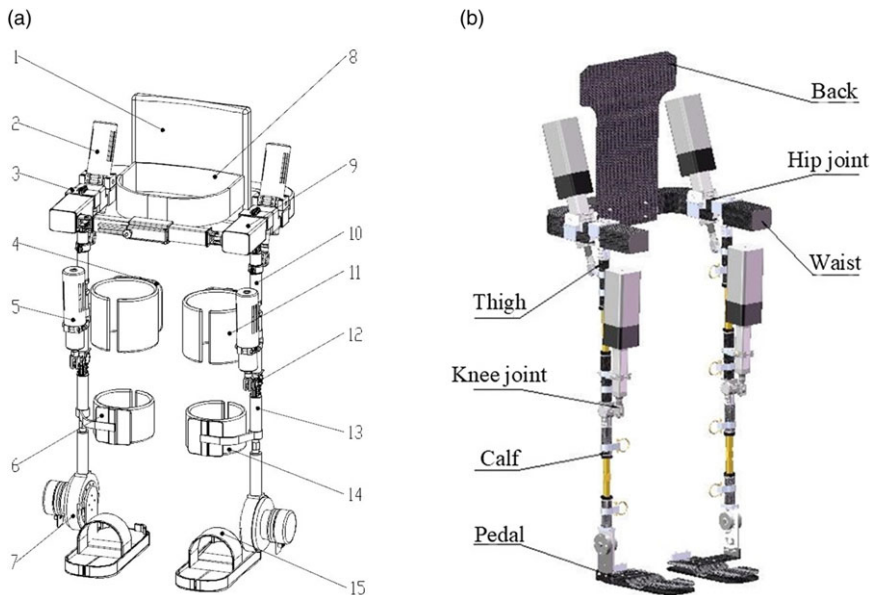
Figure 1. *(a) Overall structure diagram (1) back support; (2) actuators of hip joint; (3) hip joint component; (4) brace of thigh; (5) actuators of knee joint; (6) brace of calf; (7) ankle joint component; (8) flexible belt of waist; (9) waist component; (10) thigh component; (11) flexible belt of thigh; (12) knee joint component; (13) calf component; (14) flexible belt of calf; (15) pedal; (b). Exoskeleton mechanical diagram.*

[4] and Mina exoskeleton robot [5]. On the one hand, it is to reduce the design difficulty and cost of the exoskeleton robot; on the other hand, the study found that when the ankle joint motion is disturbed by external interference, the walking stability of human body will be greatly reduced [32]. Hence, this study makes full use of the free movable property of the ankle joint to improve the walking stability.

2.2 Dynamic modeling

Lower limb exoskeleton robots are nonlinear, strongly coupled and multiple DoFs. In order to facilitate the completion of the center of gravity calculation and spatial state description, the structure needs to be appropriately simplified. The human reference coordinate system was established based on the human anatomical posture. The exoskeleton robot was simplified to a five-link model, just as shown in Fig. 2. The white circle is the joint and the black solid circle is the CoM of the linkage.

The lower limb exoskeleton robot positional state is represented by a Cartesian coordinate system and generalized coordinates. The robot is supposed to walk in the sagittal XZ plane, and the ankle joint coordinates of the support leg and swing leg are (x_a, z_a) and (x_b, z_b), respectively. (x_{ci}, z_{ci}) is the CoM coordinate of the linkage i, $i = 1, 2,\ldots, 5$. l_i is the length of each linkage of the lower limb; d_i is the distance from the CoM of each linkage to the joint. m_i is the sum of the mass of linkage i and its corresponding human lower limb. $\theta_1, \theta_2,\ldots, \theta_5$ are the angles between each linkage and its vertical direction, and clockwise direction is positive. $M_1, M_2,\ldots, M_5$ are the joint torques of each linkage. According to the geometric relationship in Fig. 2, the CoM coordinates of each rod are

$$\begin{cases} x_{ci} = \sum_{j=1}^{i-1} \left(s_j l_j \sin \theta_j\right) + d_i \sin \theta_i + x_a \\ z_{ci} = \sum_{j=1}^{i-1} \left(s_j l_j \cos \theta_j\right) + d_i \cos \theta_i + z_a \end{cases} \tag{1}$$

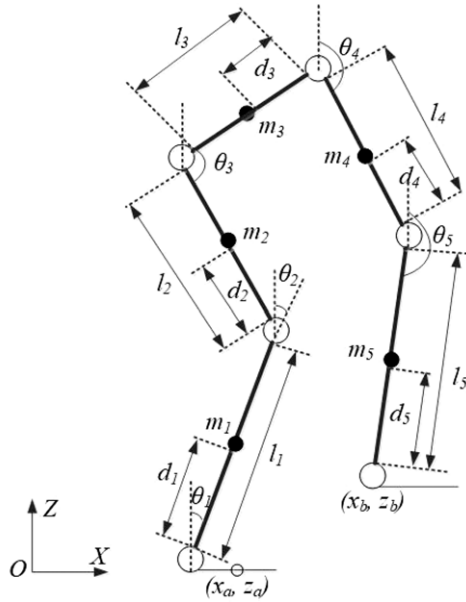where $i = 3$, $s_j = 0$; $i = 1, 2, 4, 5$, $s_j = 1$.

Figure 2. *Five-link model.*

Differentiating Eq. (1) with respect to time t, the velocity of each linkage CoM is obtained:

$$\begin{cases} \dot{x}_{ci} = \sum_{j=1}^{i-1} \left(s_j l_j \cos \theta_j \right) + d_i \dot{\theta}_i \cos \theta_i \\ \dot{z}_{ci} = \sum_{j=1}^{i-1} \left(-s_j l_j \sin \theta_j \right) - d_i \dot{\theta}_i \sin \theta_i \end{cases} \tag{2}$$

The total potential energy E_p of the lower limb exoskeleton robot system is

$$E_p = \sum_{i=1}^{5} m_i g z_{ci} = \sum_{i=1}^{5} \left\{ m_i g \left[\sum_{j=1}^{i-1} \left(s_j l_j \cos \theta_j \right) + d_i \cos \theta_i + z_a \right] \right\} \tag{3}$$

The total kinetic energy E_k of the system is

$$E_k = \sum_{i=1}^{5} \left\{ \frac{1}{2} \left[m_i \left(\dot{x}_{ci}^2 + \dot{z}_{ci}^2 \right) + I_i \dot{\theta}_i^2 \right] \right\} = \sum_{i=1}^{5} \left\{ \frac{1}{2} m_i \left[\sum_{j=1}^{i-1} \left(s_j l_j \cos \theta_j \right) + d_i \dot{\theta}_i \cos \theta_i \right]^2 \right\} +$$

$$\sum_{i=1}^{5} \left\{ \frac{1}{2} m_i \left[\sum_{j=1}^{i-1} \left(-s_j l_j \sin \theta_j \right) - d_i \dot{\theta}_i \sin \theta_i \right]^2 \right\} + \sum_{i=1}^{5} \left\{ \frac{1}{2} m_i d_{i}^2 \dot{\theta}_i^2 \sum_{j=1}^{i-1} \left[s_j l_j \dot{\theta}_j \cos \left(\theta_i - \theta_j \right) \right] \right\} \tag{4}$$

where I_i is the rotational inertia of the linkage i with respect to the CoM.

Substituting Eqs. (3) and (4) into the Lagrangian equation of the dynamical state of the lower limb exoskeleton robot system, the joint torque M_i acting on joint i during robot walking is obtained from Eq. (5).

$$M_i = \frac{d}{d_t} \frac{\partial E_k}{\partial \dot{\theta}_i} - \frac{\partial E_k}{\partial \theta_i} + \frac{\partial E_p}{\partial \theta_i}, \quad i = 1, 2, \ldots, 5 \tag{5}$$

3. Gait multi-objectives function design

In order to achieve stable walking of the lower limb exoskeleton robot, this paper abstracted the problem as a multi-objectives function optimization problem and constructed the multi-objectives function with the gait ZMP stability margin and the drive energy consumption per step. Since the hip joint is the closest
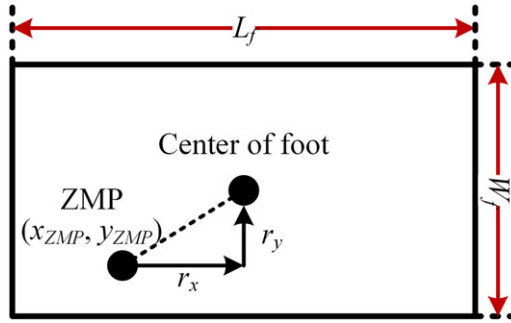
Figure 3. *ZMP stability margin diagram.*

joint to the human CoM, the change of the hip joint motion trajectory can be regarded as the change of the human CoM position, so the hip joint has a great influence on the ZMP trajectory and the gait stability of the exoskeleton robot. Assuming that the wearer's hip joint is positioned midway between the two legs during walking, the constraint along the X-axis during a single step cycle is

$$
\begin{cases}
x_{hip}(0) = -S \\
x_{hip}(T) = S \\
\dot{x}_{hip}(t) \geq 0, t \in [0, T] \\
x_l < x_{zmp} < x_r
\end{cases}
\tag{6}
$$

where x_{hip} is the trajectory of the hip joint along the X-axis, and $\dot{x}_{hip}()$ is the velocity of hip joint. T is the gait cycle, S is the step length and x_{zmp} is the component of ZMP on the X-axis. x_l and x_r are the left and right boundaries of the support polygon, respectively.

And a fifth-order polynomial is used to represent the motion trajectory of the hip joint:

$$
x_{hip} = -S + at + bt^2 + \frac{1}{2}\rho_1 t^3 + \rho_2 t^4 + \frac{1}{2}\rho_3 t^5
\tag{7}
$$

where ρ_1, ρ_2, ρ_3 are constant, $a = \frac{2S}{T} + \frac{1}{2}\rho_1 T^2 + \rho_2 T^3 + 2\rho_3 T^4$, $b = \frac{1}{2}\left(-2\rho_1 T - 4\rho_2 T^2 - 5\rho_3 T^3\right)$.

3.1. Stability criteria

In order to comprehensively measure the degree of stability during robot motion, the ZMP stability margin J_z was used to characterize the degree of stability in the horizontal direction, and the expression is shown in Eq. (8).

$$
J_z = a\sum_{i=1}^{n} r_x(i) + b\sum_{i=1}^{n} r_y(i)
\tag{8}
$$

where $r_x(i)$, $r_y(i)$ denote the distance from the ZMP point to the center of the foot in the x-axis and y-axis, respectively. n is the total number of the rod of exoskeleton robot. a and b are the weighting coefficients in the X-axis and Y-axis, respectively, and $a + b = 1$, $\frac{a}{b} = \frac{L_f}{W_f}$. L_f denotes foot length, and W_f denotes foot width, just as shown in Fig. 3.

In the absence of external disturbance, exoskeleton robots mainly rely on the frictional torque to offset the influence of the deflection torque and maintain the balance of the torque in the vertical direction. Considering that the static friction torque is complicated to model and difficult to obtain accurately, the square of the deflection torque is used to characterize the stability in the equilibrium direction.

$$
J_b = \|M_b\|_2^2
\tag{9}
$$

where M_b is the deflection torque.

The ZMP is an important criterion used to check the stability of the robot. According to the definition of ZMP, the ZMP criterion can only ensure that the horizontal component of the torque around ZMP is 0, but its vertical component is generally not 0. As the torso and body parts will generate the swaying torque around the support leg during the forward motion, when the swaying torque exceeds the maximum frictional torque on the ground, the robot will rotate around the support leg, and the deflection caused by the excessive swaying torque is more obvious, so it is necessary to consider the torque balance between horizontal and vertical directions in the gait planning process. The M_b is defined as:

$$M_b = \sum_{i=1}^{n} m_i \ddot{y}_i \left(x_i - x_{zmp} \right) - \sum_{i=1}^{n} m_i \ddot{x}_i \left(y_i - y_{zmp} \right) \tag{10}$$

$$x_{zmp} = \frac{\sum_i m_i \left(\ddot{z}_i + G_g \right) x_i - \sum_i m_i \ddot{x}_i z_i}{\sum_i m_i \left(\ddot{z}_i + G_g \right)} \tag{11}$$

$$y_{zmp} = \frac{\sum_i m_i \left(\ddot{z}_i + G_g \right) y_i - \sum_i m_i \ddot{y}_i z_i}{\sum_i m_i \left(\ddot{z}_i + G_g \right)} \tag{12}$$

where (x_i, y_i, z_i) is the coordinates of the CoM position of the i-th linkage, m_i is the mass of the i-th linkage, n is the total number of links in the robot model and (x_{zmp}, y_{zmp}) denotes the coordinates position of the ZMP.

3.2. Power consumption analysis

In terms of energy consumption, the minimum input energy of the driving joint is taken as the goal for the optimal solution of gait parameters. The average power method is an important measure of energy consumption analysis. Considering that the power of the machine is the product of motor drive torque and joint angular velocity, the average power P_{ave} of the robot during a walking cycle can be characterized as:

$$P_{ave} = \frac{1}{T} \sum_{i=1}^{5} \int_0^T \left| M_i(t) \cdot \dot{\theta}_i(t) \right| dt \tag{13}$$

where $M_i(t)$ is the torque, $\dot{\theta}_i(t)$ is the joint velocity, and T is the gait cycle.

In the process of robot motion, the exoskeleton robot may be affected by external disturbances and its own structural deformation, resulting in the instantaneous power approaching infinity, but the average power at this time may be very small. The instantaneous power peak will reduce the stability of the entire system. In response to this situation, this paper introduces the root mean square deviation of the instantaneous power, which could accurately reflect the discrete degree between instantaneous power and average power.

$$P_i = \sum_{i=1}^{5} \left| M_i(t) \cdot \dot{\theta}_i(t) \right| \tag{14}$$

$$\sigma = \sqrt{\frac{1}{(T)} \int_0^T (P_i - P_{ave})^2 \, dt} \tag{15}$$

where P_i is the instantaneous power. σ is root mean square deviation of instantaneous power.

3.3. The multi-objectives function design

We constructed the overall multi-objectives function $F(\mathbf{x})$ for optimizing the exoskeleton robot system.

$$F(\mathbf{x}) = \alpha J_z + \beta J_b + \mu \left(P_{ave} + \sigma \right) F(\mathbf{x}) = \alpha J_z + \beta J_b + \mu \left(P_{ave} + \sigma \right) \tag{16}$$

where **x** is the vector, and $\mathbf{x} = [S, \rho_1, \rho_2, \rho_3, \theta_2, \theta_3, \theta_4, \theta_5]$. α, β, $\mu \in [0,1]$, the value is set according to the proportion of gait ZMP stability margin and driving energy consumption. The smaller the $F(\mathbf{x})$, the more stable and smooth the robot gait trajectory. In this study, the ideal gait could be obtained through optimizing the gait parameters. Therefore, multi-objectives function $F(\mathbf{x})$ can be expressed as the optimized object function *GoalF* with gait parameters, and its expression is

$$GoalF = \min [F(\mathbf{x})] \tag{17}$$

4. Parameter optimization based on BSO-EOLLFF algorithm

4.1. Overview of BSO-EOLLFF algorithm

Based on the beetle antennae search algorithm and particle swarm optimization (PSO) algorithm, the BSO algorithm was designed [21, 22]. The BSO algorithm has the characteristics of fast solving speed and high accuracy and has been successfully applied in the fields of signal positioning [23] and data classification [24]. However, the BSO algorithm is easy to fall into the local optimum when searching in a high-dimensional space. In this regard, this paper introduced the EOL strategy, the LFF strategy and the dynamic mutation strategy to improve the optimization performance. In this study, we used EOL to initialize the population to promote a uniform distribution of initial population information and improve search efficiency. In addition, the LFF strategy enables the individuals to learn both their own and group experiences, allowing each individual to move purposefully and instructively and improving the convergence performance of the algorithm. Finally, a dynamic mutation strategy was introduced to increase the population diversity at the end of the iteration and prevent the algorithm from falling into the local optimum.

4.2. Population initialization based on EOL strategy

Tizhoosh introduced the concept of opposition-based learning in 2005 and showed that the inverse solution has a nearly 50% higher probability of being close to the global optimum than the current solution [25]. The main idea was that by generating reverse individuals in the area where the current individual was located and by competing for the reverse individuals with the current individuals, the best individuals will enter the next generation.

Definition 1. *(Inverse solution)*: Let there exists a real number x_o on the interval $[e, u]$, then the inverse of x is defined as $x_o' = e + u - x_o$. Based on this, it is assumed that there exists a certain n-dimensional solution point on the R-domain $p = (x_{o1}, x_{o2}, \cdots, x_{on})$, and $x_{oi} \in [e_i, u_i]$. The $p' = (x_{o1}', x_{o2}', \cdots, x_{on}')$ is defined as the inverse solution of p, where $x_{oi}' = \lambda * (e_i + u_i) - x_{oi}$, λ is a random number uniformly distributed in the interval $[0,1]$, also known as the generalization factor.

Definition 2. *(Optimization based on inverse solution)*. The problem to be optimized was a multi-objectives optimization problem, and if there exists a current solution X and its inverse solution is X', the following update mechanism is applied to X and X': (1) If $X < X'$, keep the current solution X; (2) If $X' < X$, replace X' with X; (3) If X and X' are not dominated by each other, then choose one of them at random.

Non-dominated solutions in multi-objectives optimization problems are generally considered elite individuals, usually containing more helpful information to guide the population to converge to the global optimum. The introduction of the inverse solution could expand the search area of the algorithm. Still, the search domain should be enhanced for those individuals in which original solution fitness value is greater than the inverse solution fitness value. For individuals whose original solution fitness value is smaller than the inverse solution fitness value, the value of searching the reverse area for them is higher than the value of developing their domain. Therefore, this paper took the individual whose original solution fitness value is smaller than the inverse solution fitness value as the research object and found its inverse solution, which could expand the search area and effectively avoid the time waste. If the final

algorithm could converge globally, the search area formed by the elite individuals would converge to the search area formed by the global optimal solution set. Therefore, strengthening the search of the spatial neighborhood where the privileged individuals were located will increase the algorithm's convergence speed and improve the algorithm's global convergence ability.

Definition 3. *(Elite inverse solution).* Let the inverse solution of an elite individual $X_{best} = (x_{o1}, x_{o2}, \cdots, x_{on})$ in a population p be $X'_{best} = (x'_{o1}, x'_{o2}, \cdots, x'_{on})$ in an n-dimensional search space, and the elite inverse solution could be defined as $x'_{oi} = \lambda * (de_i + du_i) - x_{oi}$. $[de_i, du_i]$ is the dynamic boundary of the population P in the i-th dimensional search space and could be calculated according to (18).

$$de_i = \min_{1 \le j \le |P|} (x_{oi}), du_i = \max_{1 \le j \le |P|} (x_{oi}) \tag{18}$$

where $|P|$ is the size of the current group.

The dynamic boundary of the search space was used instead of the fixed boundary to preserve the search experience so that the generated inverse solutions could be located in the gradually shrinking search space and promote the faster convergence of the algorithm. Since the elite inverse solution may also jump out of the boundary $[e_i, u_i]$ and become non-feasible, the method in the literature [26] was used here to reset the transgressed values, as shown in (19):

$$x'_{oi} = rand (e_i, u_i), e_i < x'_{oi} \text{ or } u_i > x'_{oi} \tag{19}$$

By forming the elite inverse solution, the detection of elite individuals' neighborhoods can be enhanced. The BSO algorithm performs reverse learning for the elite individuals of each iteration to generate the reverse population of elite individuals and participate in the evolutionary competition.

4.3. LFF strategy

In the standard BSO algorithm, the search range of beetle individuals was limited, and it is difficult to transfer the search position from the global optimal to the local optimal. Although the group search could expand the search range, there is no information exchange and feedback between the individuals, which will affect the convergence accuracy of the algorithm. The individuals in the group need to continuously learn the experience of the historical information, which plays a decisive role in the improvement of the algorithm's convergence speed. Therefore, the LFF strategy was introduced. LFF strategy was proposed by Levy in the 1930s, and Mandelbrotb described it in detail [27]. The strategy obeys the Levy distribution. LFF strategy could increase the diversity of the population, expand the search range and prevent the algorithm from falling into the local optimum. The Levy distribution satisfies:

The calculation formula of the step length s is

$$s = \frac{\psi}{|v|^{\frac{1}{\varphi}}} \tag{20}$$

where ψ and v obey normal distribution, $\mu, v \sim N\left(0, \sigma^2\right) \mu \sim N\left(0, \sigma_\mu^2\right), v \sim N\left(0, \sigma_v^2\right)$. φ is constant

$$\sigma_\psi = \left[\frac{\Gamma\left(1+\varphi\right) * \sin \frac{\pi \varphi}{2}}{\Gamma\left(\frac{1+\varphi}{2}\right) * \varphi * 2^{\frac{\varphi-1}{2}}} \right]^{\frac{1}{\varphi}} \tag{21}$$

where $\Gamma()$ is the standard gamma distribution.

The update rule of beetle direction is

$$d(t+1) = \omega * d(t) + C_1 * Levy(\theta) * (gbest(t) - X(t)) + C_2 * Levy(\theta) * (zbest(t) - X(t)) \tag{22}$$

The update rule of beetle position is

$$X(t+1) = X(t) + k_1 * s * d(t) * sign(f(X_r(t)) - f(X_l(t))) + k_2 d(t) \tag{23}$$

where *d(t)* represents the movement direction of the *t*-th generation. *X(t)* represents the beetle position of the *t*-th generation. *gbest()* represents the individual extreme value of the *t*-th generation beetle. *zbest()* represents the global extreme value. ω is the inertia weight, and C_1 and C_2 are the learning factors. *Levy(θ)* is a Levy random number. $f(x_r(t))$ and $f(x_l(t))$ are the fitness function value of the right whisker and left whisker of the *t*-th generation longhorn beetle. k_1 and k_2 are the scale factor. *sign()* is a sign function.

4.4. Dynamic mutation strategy

In the later stage of the iteration, the population diversity of the BSO algorithm will become lower, which will reduce the search ability of the algorithm. To avoid premature phenomena, a dynamic mutation strategy was introduced to increase the diversity of the population and improve convergence accuracy. At present, related scholars have proposed a variety of mutation algorithms, and the typical ones are Gaussian mutation [28] and Cauchy mutation [29]. Compared with the Gaussian mutation, the Cauchy mutation could generate a more extensive range of random numbers so that the algorithm has a greater chance to jump out of the local optimum. Meanwhile, when the peak value is small, the Cauchy mutation only takes less time to search nearby areas. The Cauchy mutation was selected for the second optimization of the population, and the mutation operation was performed on *X(t)*:

$$X^*(t) = X(t) + \zeta * C(0, 1) \tag{24}$$

$$\zeta = e^{-\gamma \frac{t}{H}} \tag{25}$$

where ζ is the weight of mutation, which will decrease with increasing iteration. *H* is the maximum number of iteration. γ is constant. *C(0,1)* is a random number generated by Cauchy operator.

4.5. The basic steps and verification of BSO-EOLLFF algorithm

Because of a large number of parameters to be optimized for the objective function in this paper, the process of gradually approximating the optimal value is complicated, the accuracy of the optimized solution is not high, and the convergence speed is relatively slow. During the search process of the traditional PSO algorithm, the position of each particle is approached toward the global optimal position in a decreasing sine wave manner. During this process, a particle converges to the currently found global optimum position, and the other particles will quickly converge to this local optimum solution. If this optimal position is only a local optimal point, then most of the particles are confined by this local optimal solution. To overcome the shortcomings of traditional algorithms, the BSO-EOLLFF optimization algorithm is proposed in this paper. According to the gait parameters to be optimized, the population individuals are selected as $x_p = [S, \rho_1, \rho_2, \rho_3, \theta_2, \theta_3, \theta_4, \theta_5]$ in the improved algorithm. Since the ankle joint has a passive DoF, θ_1 is not used as an optimization parameter. According to Eq. (16), the fitness value corresponding to the multi-objectives function $F(\mathbf{x})$ at each point x_p was calculated.

4.5.1. The steps of BSO-EOLLFF algorithm

The specific steps of the BSO-EOLLFF algorithm are:

1. Initialize the algorithm parameters: the beetle scale, iterative step length and the maximum number of iterations. Use the EOL strategy to initialize the beetle population and initialize the direction.

2. Calculate the corresponding fitness function value of the beetle and determine the individual extreme value and the global extreme value according to the fitness function value.

3. Use (22) and (23) to update the direction and position of the beetle individuals and perform cross-border processing of beetle populations.

Table I.   Psedo code of BSO-EOLLFF algorithm.

BSO-EOLLFF algorithm pseudo code:

Input: population size N, initial step length s_0, maximum number of iteration H, dimension D, inertia weight w, learning factor C_1, C_2, ratio k_1, k_2, constant γ.

Output: best BSO *zbest*

1. Initialize the BSO population $X_p(t)$ with EOL strategy, initialize the BSO toward d with random solutions, initialize Levy flight factor *Levy(θ)*;

2. Calculate fitness of $X(t)$, find global best BSO as *zbest*, find local best BSO as *gbest*;

3. **for** $i = 1: H$

4. $d(i) = d(i)/\text{norm}(d(i))$;

5. Calculate the ζ using (25);

6. **for** $j = 1:N$

7. $X_l(j)=X(j)+\text{d}(j)^*l$; $X_r(j)=X(j)-\text{d}(j)^*l$;

8. Calculate fitness of X_l and X_r;

9. Update of BSO direction using (22);

10. Update of BSO position using (23);

11. Mutation of BSO population using (24);

12. Calculate fitness of new BSO, if new BSO are better, update it in the population *zbest* and *gbest*.

End

Table II.   Parameters setting of the two algorithms.

| Parameters	Population size	Initial step length	Number of iteration(max)	Inertia weight	Learning factor	Others
Values	300	10	500	0.8	$C_1=C_2=2$	$k_1 = 0.4$ $k_2 = 0.6$ $\gamma = 10$

4. Use (24) to perform mutation operation on population.

5. Determine whether the algorithm meets the iteration termination condition. If it is satisfied, output the optimal global solution and its corresponding position, otherwise return to step 2.

The pseudo code of BSO-EOLLFF algorithm is shown in Table I.

4.5.2. Performance analysis of BSO-EOLLFF algorithm

In order to verify the performance of the BSO-EOLLFF algorithm, the Rastrigin function $z = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5\cos(2\pi x)+\cos(2\pi y)} + 20 + 2.72$ with lots of local solutions was selected to perform function optimization and convergence test. The parameter setting of the two algorithms is shown in Table II. The test results are shown in Fig. 4. The BSO-EOLLFF algorithm could accurately search for the optimal solution and would not fall into the local optimization. The algorithm's search time is 1.35 s. To verify the superiority of the BSO-EOLLFF algorithm, the PSO algorithm was used to perform parameter search for the same test function, and the experimental results are shown in Fig. 5. The experimental results showed that although the PSO algorithm can eventually obtain the global optimal
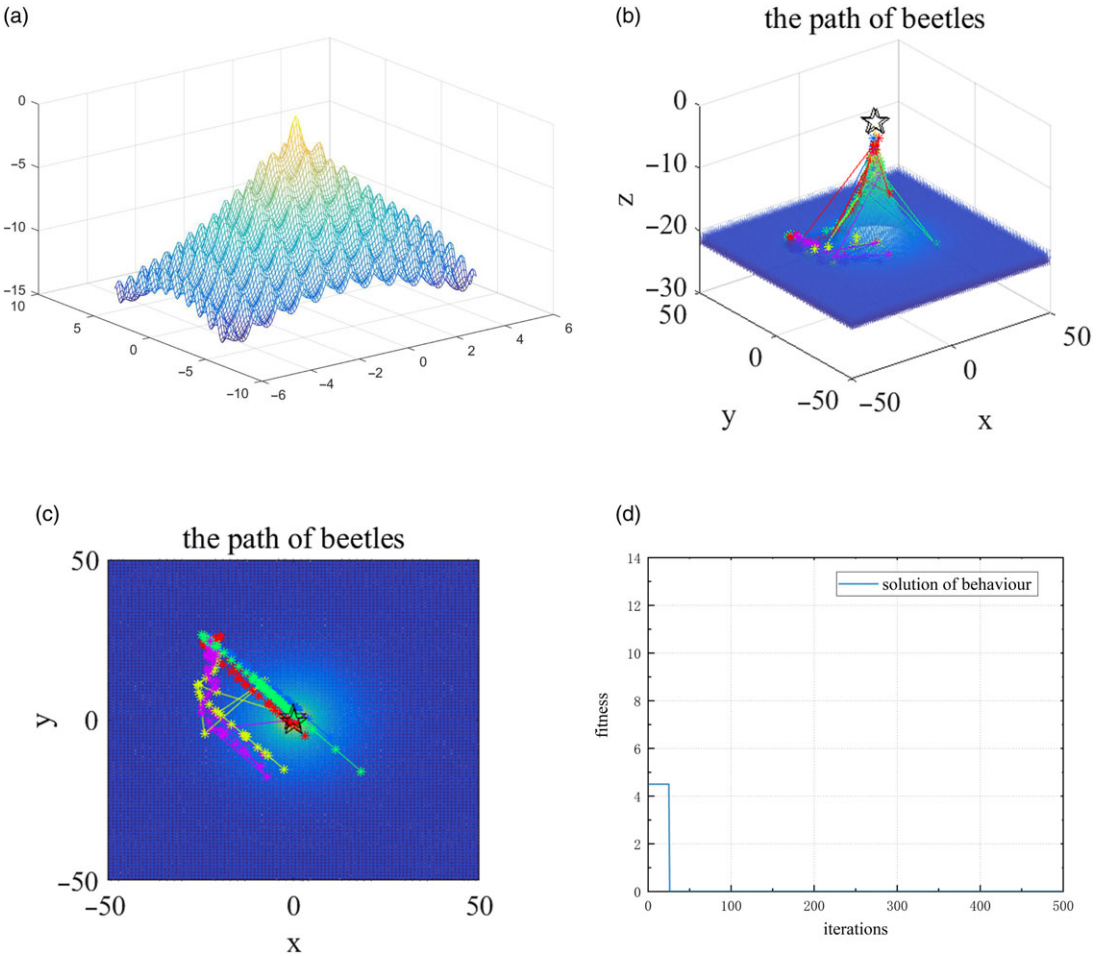
Figure 4. *Optimization results of BSO-EOLLFF algorithm for the test functions. (a) Test function diagram, (b) optimal trace of each test function, (c) the contour map of the search path and (d) the convergence curves.*
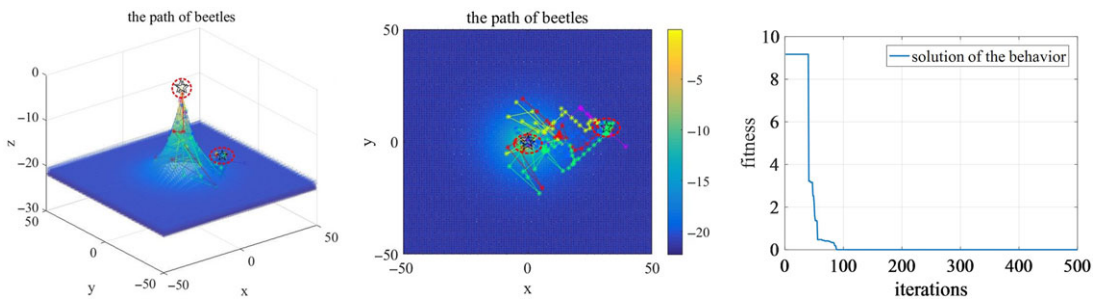
Figure 5. *Optimization results of PSO algorithm on the test functions. From left to right: optimal trace of each test function, the contour map of the search path, and the convergence curves.*

solution as well, there are four local optimal solutions in the process of finding the optimal solution, and the time to find the optimal solution is 5.73 s, which is significantly slower than the algorithm proposed in this study. The experimental results showed that the algorithm proposed in this study has a significant effect enhancement for the regular test function.
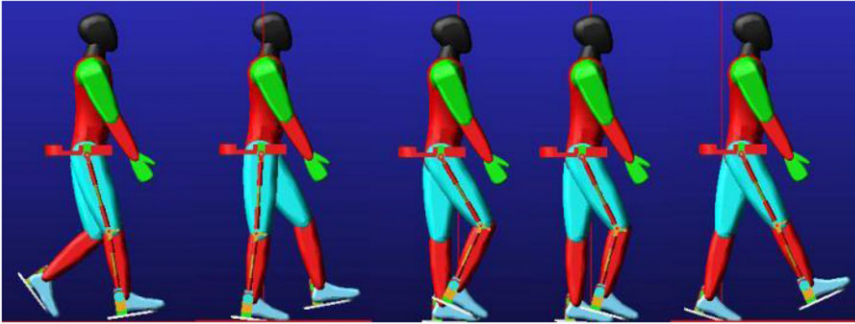
Figure 6. *The human-exoskeleton simulation process diagram.*

5. Simulation experiment analysis

SolidWorks, ADAMS and MATLAB software were used to conduct a joint comparison test to verify the performance of the optimization algorithm proposed in this paper. The 3D model of the lower limb exoskeleton robot built in SolidWorks software was imported into ADAMS software for gait characteristics analysis, and the parameter characteristics were passed to MATLAB software for gait optimization through the relevant interface. MATLAB software imported the optimized multi-objectives gait information into ADAMS software again through the control module to analyze the gait stability. The robot virtual prototype and simulation environment are shown in Fig. 6.

5.1. Parameter setting

The mechanism parameters of the lower limb exoskeleton robot are set as: $m_1 = 2\,\text{kg}$, $m_2 = 2.5\,\text{kg}$, $L_1 = 0.37$ m, $L_2 = 0.4$ m. The number of beetles n is 50, the maximum number of iterations H is 1000, inertia weight w is 0.8, individual learning factor C_1 is 1.8, and global learning factor C_2 is 1.4. The ratio k_1 and k_2 are 0.4 and 0.6, respectively. The initial weight of mutation is 0.7. The initial step length s_0 is 1. The gait cycle T is 1 s. The constant γ and φ are 10 and 1.5, respectively. In the iterative process, the condition for the iteration termination is that the error of parameter optimization is less than 0.1. In order to avoid the chance of experimental results, we conducted five simulation experiments and took the average of the experimental results for analysis.

5.2. Convergence analysis

In order to avoid the BSO algorithm from falling into the local optimum, the EOL algorithm with global search capability, the LFF algorithm with local search capability and the dynamic mutation strategy with high population diversity were introduced to improve the optimization performance. To verify the convergence of the BSO-EOLLFF algorithm, the convergence curves of each algorithm were compared with the PSO algorithm and BSO algorithm, respectively, as shown in Fig. 7.

It can be seen from Fig. 7 that the traditional PSO algorithm stops searching after 40 iterations and falls into the local optimum due to its weak global search ability [21]. When the BSO algorithm iterates 23 times, the function fitness value tends to be stable. However, it has three local optimal solutions (gray square), which will affect the convergence. Although it is lower than the global optimal value found by the PSO algorithm, the convergence speed is slower. The BSO-EOLLFF algorithm searches for the optimal solution when iterated 8 times. The BSO-EOLLFF could find the only global optimal value, which is the lowest. And the averaged convergence time is shown in Table III. The time consumption of the algorithm is 9.76 s, which is significantly faster than the other two algorithms. After the comparative analysis of the search time, the efficiency of the BSO-EOLLFF algorithm was not weakened after the addition of the EOL algorithm and LFF algorithm, which indicates that the fit and conversion between
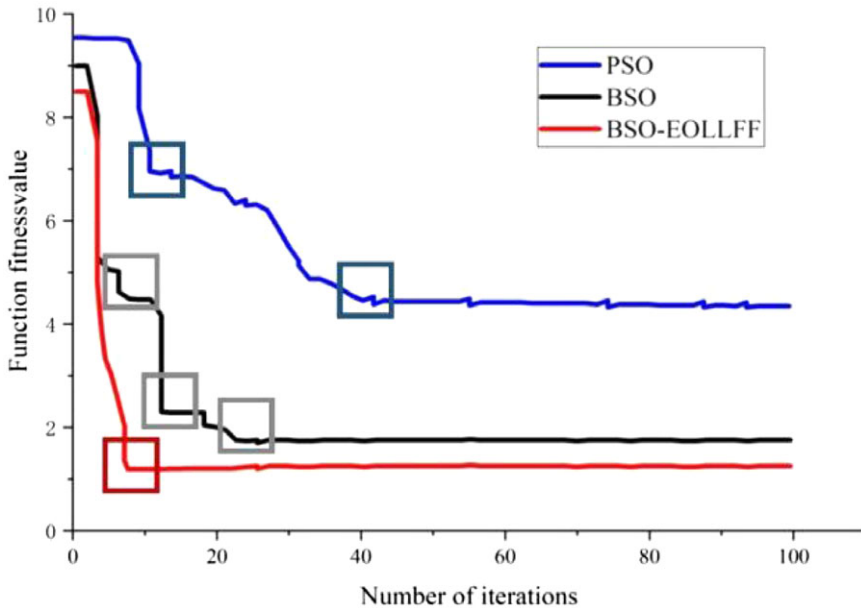
Figure 7. *Convergence curve of each algorithm.*

the algorithms are good. The strategy proposed in this paper increases the diversity of the population, which speeds up the algorithm search speed and further improves the optimization accuracy.

5.3. Stability analysis

To further verify the stability of the BSO-EOLLFF algorithm proposed in this paper, the gait ZMP stability margin and joint driving energy consumption of the lower limb exoskeleton robot optimized by the three algorithms were compared and analyzed. The exoskeleton robot's ZMP stability margin curves were optimized using the PSO algorithm, the BSO algorithm and the BSO-EOLLFF algorithm, as shown in Fig. 8.

It can be seen from Fig. 8, in one gait cycle, after the optimization of the PSO algorithm, the gait ZMP of the robot only passes through the stable center-line once. The trajectory of the center of gravity of the robot deviates from the bipedal support area, and the human body is prone to fall when walking. After optimizing by the BSO algorithm, the gait ZMP trajectory is close to the boundary. The minimum distance between the gait ZMP trajectory and the upper boundary is 0.0063 m. There are two times passing the stability center-line. The robot's center of gravity trajectory is roughly distributed in the middle of the bipedal support domain, which has an ideal stability margin. The ZMP optimized by the BSO-EOLLFF algorithm has better performance. The closest gait ZMP point to the boundary line of the support domain is 0.1272m. At this time, the minimum distance between the two is 0.0228 m. The gait ZMP trajectory is uniformly in the middle of the support domain, which can realize stable walking.
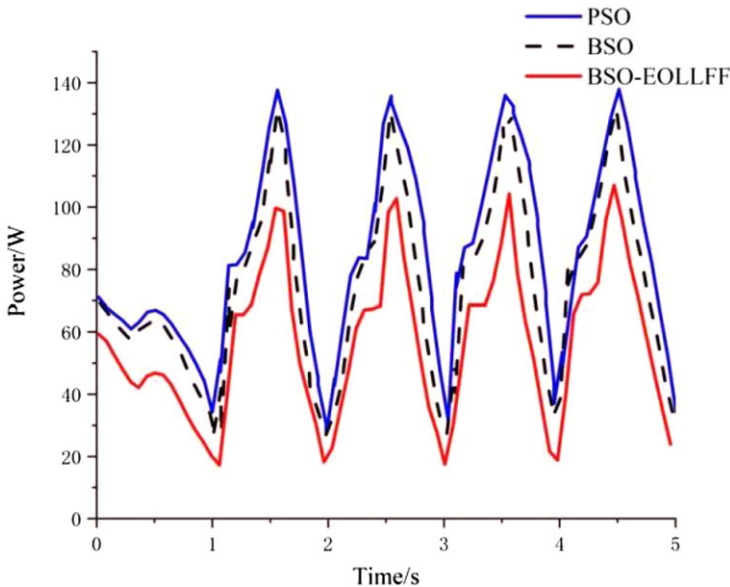
5.4. Energy consumption analysis

As can be seen from Fig. 9, the drive energy consumption of the robot increases and then decreases with each walking step, which is because the magnitude of the torque required by the joints in the oscillation process is mainly determined by the work done by gravity. To thoroughly verify the effectiveness of the proposed optimization algorithm, the minimum, maximum and average values of the drive energy consumption of the robot in 100 walking cycles optimized by the three algorithms were selected for comparison, and the algorithm energy consumption is shown in Table IV.

Table III. *The averaged convergence time.*

Algorithm	CPU time/s
PSO	67.53
BSO	35.31
BSO-EOLLFF	9.76
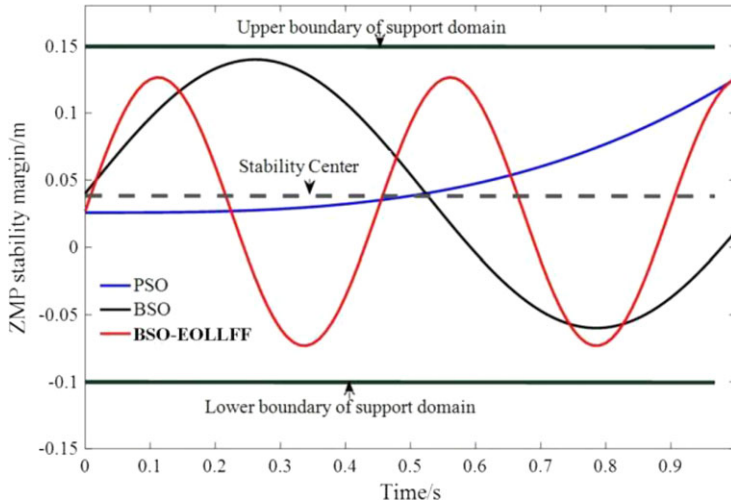
Figure 8. *ZMP stability margin curve after algorithm optimization.*


Figure 9. *Gait drive energy consumption curve.*

6. Human-machine experiments and results analysis

Human-machine experiments are conducted to demonstrate the functionality of the proposed control method. The exoskeleton robot platform is shown in Fig. 10. The range of motion of the human hip and knee joints is $-40°-145°$ and $0°-145°$, respectively. The joint motion range of the exoskeleton

Table IV. *Energy consumption analysis.*

Algorithm	Minimum/W	Maximum/W	Average/W
PSO	28.58	139.77	97.46
BSO	26.37	127.86	83.77
BSO-EOLLFF	17.82	102.33	67.48
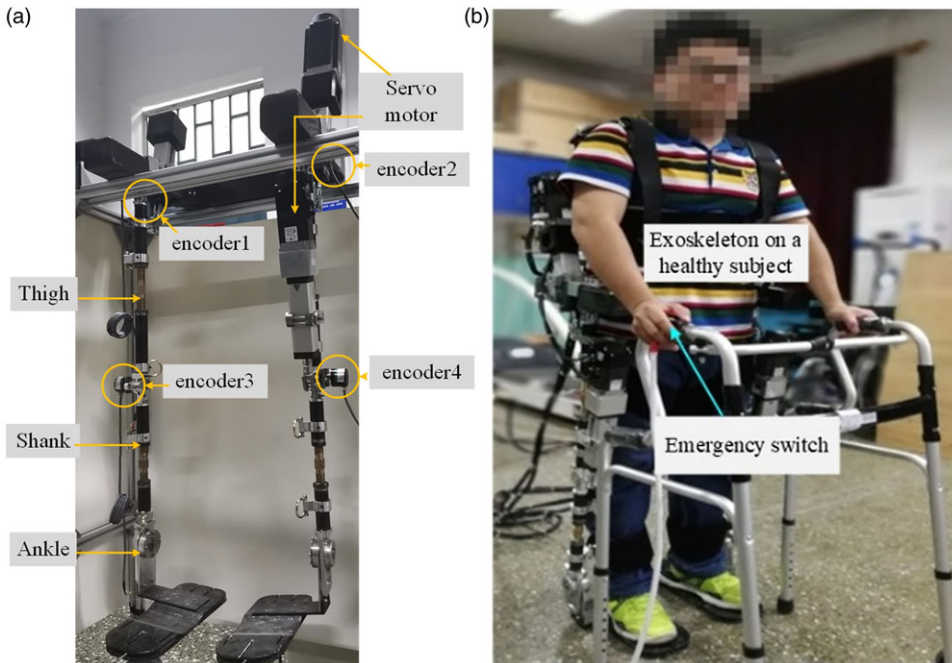
Figure 10. *(a) Prototype of the proposed exoskeleton robot and (b) Human-machine experiment.*

is 20° flexion and 40° extension for the hip joint, and 80° flexion and 0° extension for the knee joints. According to refs. [33, 34], the designed exoskeleton robot in this paper conforms to the standardization (ISO)13485. The actuator of the joint is independent and the motion speed is designed with a uniform speed of 0.3 m/s. The control system consists of the data processor Raspberry Pi and the motion controller STM32F103. Raspberry Pi is used to process data collected from angle sensors (AD36/1217AF. ORBVB, Hengstler, Germany). The communication mode between the data processor and motion controller is parallel port communication. The motion controller controls the motors through the controller area network fieldbus. Encoder 1 and encoder 2 are used to acquire the movement data of the hip joint, and encoder 3 and 4 together record the knee joint angle. For safety reasons, if the angle of the exoskeleton is greater than the allowable value, the exoskeleton will be forced to stop moving by software. In addition, the subject could cut off the power through the emergency switch button.

The main goal of this paper is to make the gait generated by the exoskeleton more efficient and stable through multi-objectives optimization. Therefore, the joint motion trajectory and ZMP trajectory(x_{ZMP}^{ref} () and y_{ZMP}^{ref} ()) of the human body need to be obtained and used as the reference trajectory. A healthy subject with the age of 25, the weight of 68 kg and the height of 170 cm is recruited for the experiment, just as shown in Fig 11. Before the gait experiment, the subject was given a detailed explanation of the test content and signed a consent form for voluntary participation in the experiment. The subject had no surgery, no history of lower extremity trauma, no balance problems or neural muscular disorders within the past six months. A written consent is obtained from the subject. The test environment was a linear walkway of 5 m in length and 1 m in width, just as shown in Fig 11. The kinematic parameters were
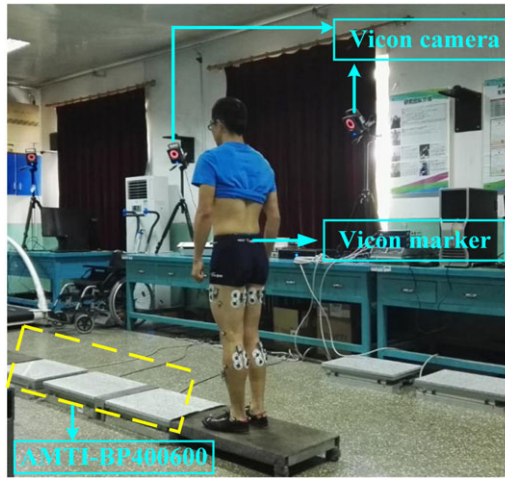
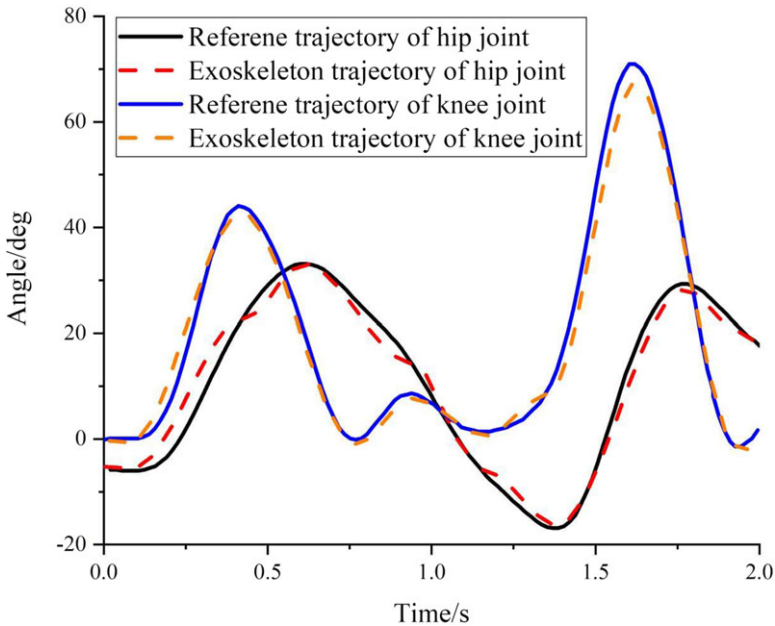**Figure 11.** *Gait experiment diagram.*

Figure 12. *Comparative analysis of joint trajectories.*

acquired by VICON acquisition system (Oxford Metrics Ltd, Oxford, UK). Ground reaction force was recorded by two AMTI force plates (AMTI Corp, Newton, USA). The data acquisition frequency is 1200 Hz. The VICON system and AMTI system are time-synchronized.

The figure for the joint position of the hip joint and the knee joint is shown in the Fig. 12. The agreement between the reference trajectory and the actual trajectory illustrates the effectiveness of the control method. In general, comparisons between a pair of kinematic or kinetic curves are performed by calculating the Pearson correlation coefficient (R), which quantifies the strength of the linear relationship between the two curves (i.e. similarity of their shapes) [35]. The results of R are shown in Table V. The experimental results show that the tracking performance of the BSO-EOLLFF controller designed in this paper is good, and the overall average error of hip joint position and knee joint position is 0.669 and 0.877 deg, respectively. And the biggest error generated during the whole exercise of hip joint position

Table V. *Analysis of reliability: Results of R.*

Joint	Hip joint	Knee joint
R	0.95	0.97
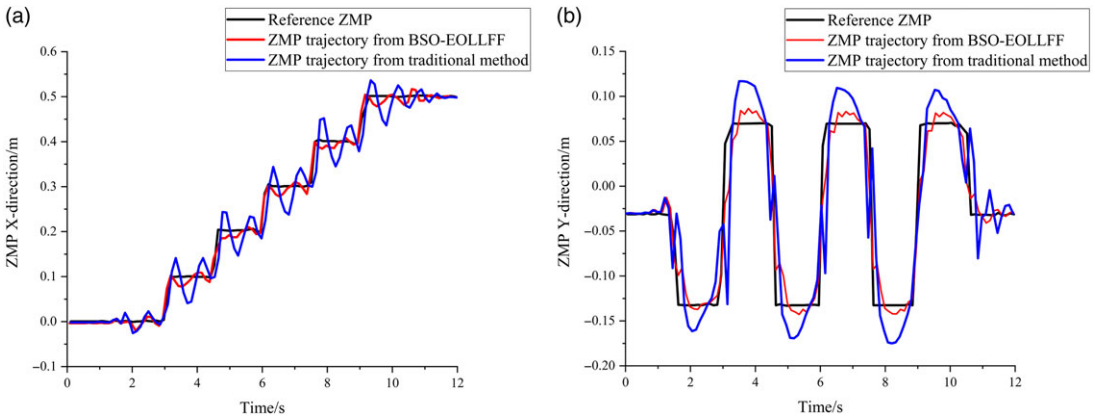
Figure 13. *Comparison analysis of ZMP trajectory. (a) X-direction; (b) Y-direction.*

and knee joint position is 1.981 and 1.319 deg, respectively, which meets the training requirements of the lower limb rehabilitation robot.

$$R = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{26}$$

$$\text{cov}(X, Y) = E\left[X - E(X)\left(Y - E(Y)\right)\right] \tag{27}$$

$$\sigma_X = \sqrt{\frac{\sum_{i=1}^{n}\left(X - \overline{X}\right)^2}{n}} \tag{28}$$

where $E(X)$ and $E(Y)$ are the expected value of variable X and Y, respectively. $\overline{X}$ and σ_X are the mean value and standard deviation of the sample X. In this paper, X and Y represent the reference trajectory and the actual trajectory, respectively.

We collected the energy consumption values for six gait cycles and take the average value to obtain the average power consumption for a gait cycle. The average power value of the BSO-EOLLFF algorithm for trajectory energy optimization is 65.8 W. Compared with the traditional PSO algorithm, the average power value was decreased by 30.5 W.

Figure 13 gives a comparison of the actual ZMP trajectories for the two sets of experiments. The magnitude of the ZMP tracking error in the X-direction and Y-direction is measured by E_x and E_y, respectively:

$$E_x = \int_{t_0}^{t_f} \left[x_{ZMP}^{ref}(t) - x_{ZMP}(t)\right]^2 dt \tag{29}$$

$$E_y = \int_{t_0}^{t_f} \left[y_{ZMP}^{ref}(t) - y_{ZMP}(t)\right]^2 dt \tag{30}$$

where $x_{ZMP}^{ref}()$ and $y_{ZMP}^{ref}()$ are the ZMP reference value in X-direction and Y-direction, respectively. $x_{ZMP}()$ and $y_{ZMP}()$ are the ZMP actual value in x-direction and y-direction, respectively. t_0 and t_f are start time and end time of gait cycle, respectively.

Compared with the traditional method, *Ex* is reduced by 45.3%, and *Ey* is reduced by 52.7%. In Fig. 13, it is evident that the exoskeleton robot system can effectively eliminate the disturbance during

the walking process. In the X-direction, the reference ZMP trajectory can be tracked better, and the oscillation phenomenon is not apparent. In the Y-direction, compared to the reference ZMP trajectory, the method proposed in this study produces a specific deviation with an overall average error of 0.035m, which is a significant progress compared to the traditional method.

7. Conclusion

A vital issue for exoskeleton robots is maintaining gait stability during walking to prevent the wearer from falling. By reducing the minimum distance between the ZMP and the center of the foot, increasing the number of intersections between the ZMP and the stability center-line and reducing the drive energy consumption, it can be judged that the stability margin of the ZMP after the parameter optimization of the BSO-EOLLFF algorithm has been improved considerably. The robot gait can achieve stable and smooth walking with less energy consumption, which fully demonstrates the effectiveness of the BSO-EOLLFF algorithm. The agreement between the reference trajectory and the actual trajectory illustrates that the applicability of the proposed algorithm in robot walking is better, and the rationality and reliability of planned walking are ensured. In the future, the exoskeleton will also consider additional movement modes, such as climbing steep hills and running. In each motion mode, dynamic stability and exoskeleton interference forces will be analyzed in detail, and adaptive control strategies adapted to each motion mode will be designed.

References

[1] Y. Li, Q. Wei, W. Gou and C. He, "Effects of mirror therapy on walking ability, balance and lower limb motor recovery after stroke: A systematic review and meta-analysis of randomized controlled trials," *Clin. Rehabil.* **32**(8), 1007–1021 (2018).

[2] E. H. Beisheim, J. R. Horne, R. T. Pohlig and J. M. Sions, "Differences in measures of strength and dynamic balance among individuals with lower-Limb loss classified as functional level K3 versus K4," *Am. J. Phys. Med. Rehab.* **98**(9), 1–10 (2019).

[3] Y. Zou, H. Ma and Z. Han, "Force control of wire driving lower limb rehabilitation robot," *Technol. Health Care* **26**(11), 1–10 (2018).

[4] A. Esquenazi, M. Talaty, A. Packel and M. Saulino, "The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury," *Am. J. Phys. Med. Rehab.* **91**(11), 911–921 (2012).

[5] C. Mummolo, W. Z. Peng, S. Agarwal, R. Griffin, P. D. Neuhaus and J. H. Kim, "Stability of mina v2 for robot-assisted balance and locomotion," *Front. Neurorobot.* **12**, 62 (2018).

[6] S. Wang, L. Wang, C. Meijneke, E. Van Asseldonk, T. Hoellinger, G. Cheron and H. Van Der Kooij, "Design and control of the MINDWALKER exoskeleton," *IEEE Trans. Neur. Syst. Reh.* **23**, 277–286 (2014).

[7] B. W. Gasser, D. A. Bennett, C. M. Durrough and M. Goldfarb, "Design and Preliminary Assessment of Vanderbilt Hand Exoskeleton," **In:** *2017 International Conference on Rehabilitation Robotics (ICORR)* (IEEE, 2017) pp. 1537–1542.

[8] M. Yang, X. Wang, Z. Zhu, R. Xi and Q. Wu, "Development and control of a robotic lower limb exoskeleton for paraplegic patients," *Proc. Inst. Mech. Eng.* **233**, 1989–1996 (2019).

[9] X. Zhang, K. Qin and Y. Shi, "Review of human exoskeleton suit technology," *Comput. Sci* **42**, 1–7 (2015).

[10] M. Vukobratović and B. Borovac, "Zero-Moment Point," **In:** *Thirty Five Years of Its Life* vol. 1 (2004) pp. 157–173.

[11] X. Liu, K. H. Low and H. Yu, "Development of A Lower Extremity Exoskeleton for Human Performance Enhancement," **In:** *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2004) pp. 3889–3894.

[12] K. H. Low, X. Liu and H. Yu, "Development of NTU wearable exoskeleton system for assistive technologies," **In:** *IEEE International Conference on Mechatronics and Automation* vol. 2 (IEEE, 2005) pp. 1099–1106.

[13] X. Liu and K.H. Low, "Development and Preliminary Study of the NTU Lower Extremity Exoskeleton," **In:** *2004 IEEE Conference on Cybernetics and Intelligent Systems* (IEEE, 2004) pp. 1243–1247.

[14] K. H. Low, X. Liu, C. H. Goh and H. Yu, "Locomotive control of a wearable lower exoskeleton for walking enhancement," *J. Vib. Control* **12**(12), 1311–1336 (2006).

[15] C. W. WuC. W. Wu, Dynamic Stability Analysis and Gait Tracking Control for the Lower Extremity Exoskeleton", M.S. Thesis, (Central South University, Hunan, China, 2014).

[16] S. Feng, E. Whitman, X. Xinjilefu and C. G. Atkeson, "Optimization-based full body control for the DARPA robotics challenge," *J. Field Robot* **32**(2), 293–312 (2015).

[17] G. M. Gasparrigm, S. Manaras and D. Caporaled, "Efficient walking gait generation via principal component representation of optimal trajectories: Application to a planar biped robot with elastic joints," *IEEE Robot. Automat. Lett*. **3**, 2299–2306(2018).

[18] Z. Ang, M. Xiaojing, Z. Jiayong and H. Kun, "Integrated design and finite element analysis of the upper limb exoskeleton assist system," *Modern Manufact. Eng*. **5**, 59–63 (2020).

[19] L. Juan, C. Xiaowei, L. Weida, Z. Xiao and Z. Qihuan, "Static and stable gait planning method of lower limb exoskeleton walking-assisting robot based on center of gravity stability constraint," *Mech. Electr. Eng*. **34**, 1075–1080 (2017).

[20] K. Sreenath, H.-W. Park and I. Poulakakis, "Embed- ding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *Int. J. Robot. Res*. **32**(3), 324–345 (2013).

[21] P. Zhang and J. Zhang, "Lower limb exoskeleton robots' dynamics parameters identification based on improved beetle swarm optimization algorithm," *Robotica* **40**(8), 1–16 (2022). doi: 10.1017/S0263574721001922.

[22] X. Jiang and S. Li, "Beetle antennae search without parameter tuning (BAS-WPT) for multi-objectives optimization," *Neural Evol. Comput*. **27** , 78–86(2019), Available: ar Xiv: 1711. 02395.

[23] Y. Liu, Z. Qian and D. Jia, "Universal localization algorithm based on beetle antennae search in indoor environment," *J. Electron. Inform. Technol*. **41**, 1565–1571 (2019).

[24] Q. Wu, Z. Ma, G. Xu, S. Li and D. Chen, "A novel neural network classifier using beetle antennae search algorithm for pattern classification," *IEEE Access* **7**, 64686–64696 (2019).

[25] H. R. Tizhoosh, "Opposition-based Learning: A New Scheme for Machine Intelligence," **In:** *Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation* (USA: IEEE, 2005) pp. 695–701.

[26] X.-Y. Zhou, Z.-J. Wu, H. Wang, K. S. Li and H. Y. Zhang, "Elite opposition-based particle swarm optimization," *Acta Electron. Sin*. **41**, 1647–1652 (2013).

[27] R. Jiadong, L. Xinqian, W. Qian H. Haitao and Z.. Xiaolin, "An multi-level intrusion detecti on method based on KNN outlier detection and random forests," *J. Comput. Res. Develop*. **56**, 566–575 (2019).

[28] G. Ni, G. Ling, Y.-Y. He and W. Hai, "A light weight intrusi on detection model based on auto encoder network with feature reduction," *Acta Electron. Sin*. **45**, 730–739 (2017).

[29] G. Rudolph, "Local convergence rates of simple evolutionary algorithms with Cauchy mutations," *IEEE Trans. Evolut. Comput*. **4**(4), 249–258 (1997).

[30] J. N. Maharaj, M. J. Rainbow, A. G. Cresswell, S. Kessler, N. Konow, D. Gehring and G. A. Lichtwark, "Modelling the complexity of the foot and ankle during human locomotion: The development and validation of a multi-segment foot model using biplanar videoradiography," *Comput. Method Biomech. Biomed. Eng.* **25**(5), pp. 554–565 (2022).

[31] B. Killinger, J. D. Lauver, L. Donovan and J. Goetschius, "The effects of blood flow restriction on muscle activation and hypoxia in individuals with chronic ankle instability," *J. Sport Rehabil*. **29**(5), pp. 633–639 (2019).

[32] N. Sato, H. Nunome, M. Hattori and Y. Ikegami, "Effect of individual ankle taping components on the restriction of ankle external inversion moment," *Sport Biomech*. **37**, 1–11 (2020).

[33] S. Mihcin, "Simultaneous validation of wearable motion capture system for lower body applications: Over single plane range of motion (ROM) and gait activities," *Biomed. Eng./Biomedizinische Technik* **67**(3), 185–199 (2022). doi: 10.1515/bmt-2021-0429.

[34] S. Mihcin, S. Ciklacandir, M. Kocak and A. Tosun, "Wearable motion capture system evaluation for biomechanical studies for hip joints," *ASME. J. Biomech. Eng.* **143**(4), 044504 (2021). doi: 10.1115/1.4049199.

[35] M. Iosa, A. Cereatti, A. Merlo, I. Campanini, S. Paolucci and A. Cappozzo, "Assessment of waveform similarity in clinical gait data: the linear fit method," *BioMed Res. Int.,* **2014**, Article ID 214156 , 7 pages (2014). doi: 10.1155/2014/214156.