

Research Article

Exoskeleton Follow-Up Control Based on Parameter Optimization of Predictive Algorithm

Shijia Zha¹, Tianyi Li¹, Lidan Cheng¹, Jihua Gu¹, Wei Wei¹, Xichuan Lin², and Shaofei Gu³

¹College of Optoelectronics Science and Engineering, Soochow University, Suzhou Jiangsu Province, 215000, China

²Micro-Nano Automation Institute, Jiangsu Industrial Technology Research Institute, Suzhou, Jiangsu Province 215131, China

³Shanghai Huangpu District Fire Rescue Detachment, Shanghai 200001, China

Correspondence should be addressed to Wei Wei; weiwei0728@suda.edu.cn

Received 28 May 2020; Revised 15 October 2020; Accepted 8 January 2021; Published 21 January 2021

Academic Editor: Jose Merodio

Copyright © 2021 Shijia Zha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The prediction of sensor data can help the exoskeleton control system to get the human motion intention and target position in advance, so as to reduce the human-machine interaction force. In this paper, an improved method for the prediction algorithm of exoskeleton sensor data is proposed. Through an algorithm simulation test and two-link simulation experiment, the algorithm improves the prediction accuracy by $14.23 \pm 0.5\%$, and the sensor data is smooth. Input the predicted signal into the two-link model, and use the calculated torque method to verify the prediction accuracy data and smoothness. The simulation results showed that the algorithm can predict the joint angle of the human body and can be used for the follow-up control of the swinging legs of the exoskeleton.

1. Introduction

The exoskeleton is a wearable device that combines human intelligence and mechanical power, widely appearing in the fields of assistance, rehabilitation training, and disability assistance. In recent years, with the continuous development of drive technology and sensor technology, more and more exoskeletons have appeared in the market [1]. Exoskeletons are often designed to perform specific functions such as walking, weight-bearing, lumbar support, and spinal support. In order to assist the specific movements of the human body, a very critical point for the exoskeleton is to realize the recognition of the human motion intention.

1.1. Introduction to Exoskeleton Classification. There are many ways to classify active exoskeletons. One classification way is based on the methods of the exoskeleton acquiring the intention of human motion, which can be roughly divided into four categories: The first is preprogrammed. The gait of the exoskeleton system is designed, so the wearer can only intervene in limited ways with devices such as but-

tons or HMI (Human Machine Interface). However, only relying on some external devices to obtain the motion intention of the human body limits the use scope of the exoskeleton, which makes this kind of exoskeleton often appear in the rehabilitation and correction equipment with fixed gait [2]. The second is the use of EEG (electroencephalogram) signals to identify the intention of human movement. Such systems are susceptible to interference from the external environment and are not suitable for the wearer to perform multibrain tasks [3]. The third method uses the surface EMG (electromyography) signals of the human body to capture the movement intention of the human body. Obtaining surface EMG signals is usually done by attaching electrodes on the skin of the human body, which is also vulnerable to the environment. Sweat on the surface of the skin often affects the accuracy of the signals [4], and long-term wear is easy to cause the surface electrode to fall off; some teams have developed exoskeletons (e.g., HAL) that recognize the body's intentions based on electromyographic signals. The fourth way is to collect human body motion mechanics information to identify human body motion intentions. Such systems have installed

a large number of kinematics and dynamics sensing devices on the exoskeleton and human body to obtain the interaction information of the human body and the motion intention of the human body. Many lower limb exoskeletons, such as WEAR [5], MEBOTX-EXO [6], and HUALEX [7], use kinematic information to capture the human body's intention and kinetic information to control the motion of the exoskeleton. This type of exoskeleton has the characteristics of easy to put on and take off, and the system sensor signal is stable, which makes it become the main research direction of exoskeleton. However, the kinematic sensor requires the human body to drive the exoskeleton movement, inevitably causing the delay of motion intention perception and control.

Since the exoskeleton involved in this paper is the fourth method to obtain the human movement intention and the sensor data is collected by the sensor network, it takes some time until the control signal is generated. The actuators (typically motors or hydraulic) cause time delay, too [8]. Meanwhile, the motion signal of the human body is behind the motion intention of the human body. Therefore, it is necessary to solve the time delay in the fourth exoskeleton system.

1.2. Introduction to Prediction Algorithm. In the exoskeleton control strategies, only relying on the sensor data detected by the sensors of the system and giving the exoskeleton drive structure motion instruction will cause the exoskeleton and the human body a position deviation. The position deviation is the main source of human-machine interaction. An appropriate prediction algorithm needs to be added to help the control strategies judge the wearer's motion intention and the possible motion position at the next moment according to the current and historical motion sensing data, so as to give the system appropriate control commands. Due to the significant difference in the delay between the exoskeleton and the wearer due to different wearing methods and the tightness of the binding, some prediction methods can only predict the one step size, such as Kalman prediction [9], need to continuously iterate the Kalman gain coefficient according to the measured value, and cannot deal with the longer step size prediction. Similar problems also exist in the prediction methods based on the ARMA model, such as LMS linear prediction [10] and RLS linear prediction [11, 12]. When the prediction step size becomes longer, the accuracy of the prediction will decrease significantly [13]. Moreover, LMS linear prediction and RLS linear prediction require more historical data and a large amount of computation, and the prediction curve is not smooth, which makes it difficult to implement such prediction algorithm in an embedded system.

Yang [13] et al. compared the performance of LMS and Takens prediction algorithms in the prediction of human gait data. The results showed that Takens is more stable, accurate, and suitable than LMS, but it did not solve the zero dead time problem of the prediction algorithm. So, Duan et al. [14] use the combination of the Takens prediction algorithm and Newton-based-method to solve the problem of the zero dead time of the prediction algorithm but did not make progress in solving the parameter tuning process.

In this work, we have made improvements to the prediction algorithm for exoskeleton, so that the parameters of the

prediction algorithm can be optimally determined. Applying it to exoskeleton gait prediction can accurately predict the joint angle of the exoskeleton, making joints motor reach the aim position in advance.

The reminder of this paper is organized as follows:

In Section 2.1, we introduce the IMU-based pose capture system. In Section 2.2, we briefly introduce the Takens prediction algorithm. In Section 2.3, we introduce the process of the PSO-Takens algorithm. Section 2.4 is mainly about the simulation of the control of the prediction algorithm in the two-link model.

In Section 3, the experimental results are analyzed. We mainly discussed the performance of the improved prediction algorithm and the original prediction algorithm in exoskeleton gait data prediction.

In Section 4, we conclude the paper.

2. Materials and Methods

2.1. Motion Capture System. The process of human lower limb movement has the characteristics of high autonomy, complex information, diverse movements, and multiple degrees of freedom. Biomechanical simulation and experimental studies have shown that the motion consumption power in the sagittal plane of the human body is the largest relative to the frontal plane and horizontal plane [15], so the three-dimensional walking motion of human hip joint often can be simplified to a single plane motion in the sagittal plane. The corresponding position sensor used in the exoskeleton is usually placed in a position parallel to the sagittal plane of the human body, which describes the motion angle of the sagittal plane of the lower limb joints of the human body. With the development of inertial measurement units, more and more IMUs (inertial measurement units) were used as examples of kinematic sensors, beginning to appear in the motion capture system. Compared with the method of collecting joint angles using an absolute encoder, IMU has an advantage of flexible installation and cheap price, and its measurement accuracy can reach 0.1° , which can meet the needs of the human motion capture system.

In order to collect human gait information and verify the accuracy of the prediction algorithm, we designed a flexible motion capture system for human sagittal plane motion based on IMUs. The system will automatically capture the sagittal plane motion data of the human body at 50 Hz sampling frequency and transmit the motion data through CAN (Controller Area Network) (Figure 1(a)) to the data collection and processing module. As shown in Figure 1(b), the system mainly collected the pitch angle data of the back, thigh, shank, and the pressure sensor data of the heel. The system also includes power module, data collection and processing module, and data transmission module. Figure 1(c) shows the actual mechanical structure of the exoskeleton system, a hinged structure is adopted between the thigh and shank, and this structure can support the exoskeleton system and reduce the load on the body while standing upright, but it also leads to the inability to directly measure the motion angle of the human knee joint through the angle sensor.

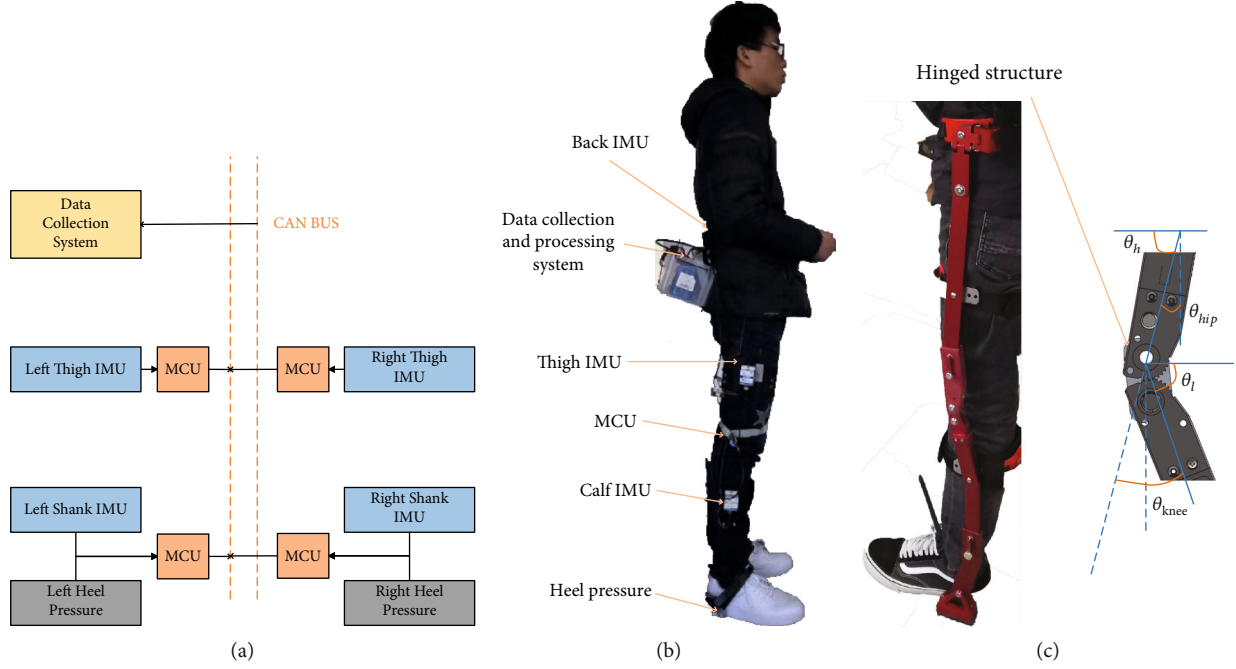


FIGURE 1: Motion capture system and exoskeleton structure. (a) Nets of the motion capture system. (b) Motion capture system. (c) Mechanical structure of the exoskeleton.

Therefore, IMUs were placed on the thighs, shanks, and back to calculate the angle of the human joints.

The formula for calculating the angle of the hip and knee joints is as follows:

$$\begin{aligned}\theta_{\text{hip}} &= \frac{\pi}{2} - \theta_h, \\ \theta_{\text{knee}} &= \pi - \theta_h - \theta_l.\end{aligned}\quad (1)$$

θ_h, θ_l represents the pitch angle of thigh and shank, respectively; due to the mirror relationship between the left and right legs, the calculated angle values are opposite, and there is a phase difference between the hip joints and knee joints.

2.2. Takens Prediction Algorithm. The continuous walking of the human body is periodic and nonlinear. The left and right leg sensor data are only different in phases and directions. Therefore, the algorithms used in the prediction of the joint angle are the same; the prediction algorithms discussed in this section are nonlinear time series Takens prediction algorithm of the lower limb hip and knee joint data.

The Takens algorithm for nonlinear time series forecasting is based on the Takens embedding theorem, which is also called the weighted zero-order local prediction method. This method is closely related to Takens' reconstruction theorem. It is essentially a nonlinear time series analysis method, which requires historical data of the system to obtain the best performance. The main idea is to find several historical data vectors that are the most similar to the current reconstruction delay vector by traversing the sensor data for a period of time, and the historical data vector is normalized and weighted according to the similarity with the current sampling data. The prediction data at a specific time is deter-

mined through function fitting or search. The predicted data of each vector are multiplied by their respective weights to obtain the final predicted value at the determined moment. The algorithm implementation steps are as follows:

- (1) According to the Takens embedding theorem, for a given time series $y(t)$, select the appropriate sampling interval Δt and embedding dimension n , and collect and store n data; the adjacent data differ in time Δt ; constitute the reconstruction delay vector $D(t)$. The time series $y(t)$ can be any type of sensor data in the motion capture system, such as angle, angular velocity, or heel pressure data. The contents of $D(t)$ is

$$D(t) = [y(t), y(t - \Delta t), y(t - 2\Delta t), y(t - 3\Delta t) \cdots y(t - (n - 1)\Delta t)] \quad (2)$$

- (2) Calculate the similarity between the current reconstruction delay vector $D(t)$ obtained by sampling at the current time and the historical reconstruction delay vector $D_i(t)$ obtained from all previous observations; the methods for calculating similarity in reconstruction delay vectors are usually Euclidean distance, Pearson's correlation coefficient, Manhattan distance, and hash distance. The Euclidean distance has the following expression in this algorithm:

$$\delta(i) = \sqrt{\sum_{k=1}^n (y(t - (k - 1)\Delta t) - y_i(t - (k - 1)\Delta t))^2} \quad (1 \leq i \leq N). \quad (3)$$

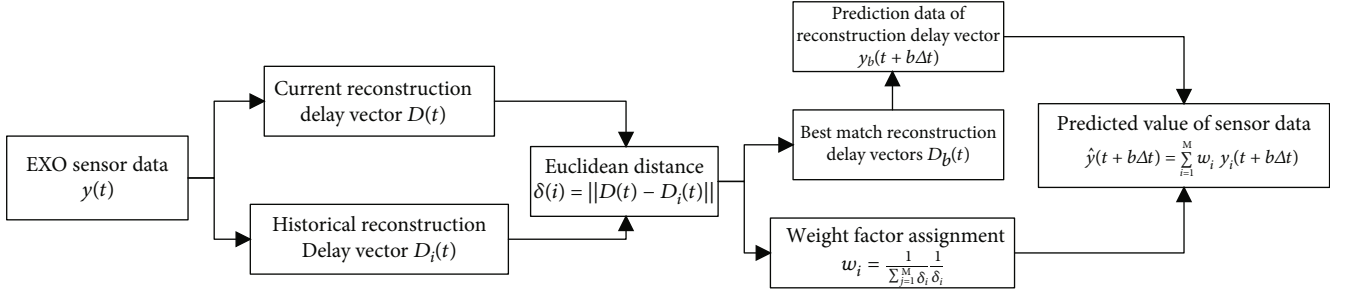


FIGURE 2: Flow chart of the Takens algorithm.

In formula (3), the value of i represents the position of the currently traversed vector, N is the reconstruction delay vector length. The smaller the Euclidean distance is, the more similar the two reconstruction delay vectors are, and the more accurate when the historical reconstruction delay vector is used to predict the data of the subsequent period of time. While Pearson's correlation coefficient or Manhattan distance is used as a similarity calculation, the expression becomes

$$\delta(i) = \frac{\sum_{k=1}^n (y(t - (k-1)\Delta t) - \bar{y})(y_i(t - (k-1)\Delta t) - \bar{y}_i))}{\sqrt{\sum_{k=1}^n (y(t - (k-1)\Delta t) - \bar{y})^2} \sqrt{\sum_{k=1}^n (y_i(t - (k-1)\Delta t) - \bar{y}_i)^2}} \quad (1 \leq i \leq N),$$

$$\delta(i) = \sum_{k=1}^n |y(t - (k-1)\Delta t) - y_i(t - (k-1)\Delta t)| \quad (1 \leq i \leq N). \quad (4)$$

In formula (4), \bar{y} is the average value of the reconstructed delay vector, Pearson's correlation coefficient and Manhattan distance can also be used as a similarity calculation function, and the difference is only the amount of calculation. Among the three calculation methods, Manhattan has the smallest amount of calculation and Pearson's correlation coefficient is the largest

- (3) From the observed historical reconstruction delay vector $D_i(t)$, the best matching M sets of reconstruction delay vectors $\{D_1(t), \dots, D_M(t)\}$ are selected according to the similarity with the reconstruction delay vector $D(t)$ at the current moment. The corresponding Euclidean distances are $\{\delta(1), \dots, \delta(M)\}$. The predicted value $\{y_1(t + b\Delta t), \dots, y_M(t + b\Delta t)\}$ of each group of reconstruction delay vectors is determined by searching, $T = b\Delta t$ is the predicted duration. The weighting factor formula of each best matching amount in the final prediction amount calculation formula is as follows:

$$w_i = \frac{1}{\sum_{j=1}^M \delta_j} \frac{1}{\delta_i} \quad (1 \leq j \leq M) \quad (5)$$

- (4) According to the predicted values and weight factors of each historical reconstruction delay vector group, the estimated predicted values at time $b\Delta t$ after the current moment $y(t)$ are calculated as follows:

$$\hat{y}(t + b\Delta t) = \sum_{i=1}^M w_i y_i(t + b\Delta t) \quad (6)$$

The flow chart of the algorithm is shown in Figure 2.

The Takens algorithm is to make predictions on the sensor data for a period of time by searching historical sensor data, which means that the prediction algorithm is to find the most similar data out of the current motion situation from the historical vectors. The sensor data of the human walking motion detected on the sagittal plane has obvious periodicity, so the Takens algorithm has a well performance in predicting such signals. Among multiple optimal reconstruction delay vectors, the algorithm can dynamically adjust the weight factor through the Euclidean distance, which can adjust the proportion of the optimal reconstruction delay vector in the final prediction data.

2.3. PSO-Takens Prediction Algorithm Design. Although the Takens algorithm cannot predict that all motion sensor data are completely correct, some papers also give an optimization method based on this algorithm [14], which makes the Takens algorithm used in embedded systems become possible. Whitney's topological embedding theorem shows that the phase space of the original system can be reconstructed by any value of the delay time of the single variable time series in the case of noiseless and infinitely long time series. In fact, the measured time series is of limited length and is inevitably polluted by noise. Therefore, arbitrary delay time cannot reconstruct the phase space of the system; the key to reconstruct the phase space using the measured time series is how to select these reconstruction parameters. However, the Takens embedding theorem does not indicate how to select these parameters. This problem has become an important issue in nonlinear time series analysis.

Duan et al. have found that inappropriate historical vector dimension and embedding dimension will cause irrelevant data to enter the reconstruction delay vector, resulting in lower prediction accuracy [14]. While inappropriate prediction duration and optimal matching number will cause data jitter and affect the smoothness of the predicted data. Fortunately, the predicted duration, which is also called delay time, is a constant we want to compensate. It is mainly caused by inertial sensors and inherent defects of the system, and the duration can be measured by wearing exoskeletons. The delay time is expressed as the angle difference between the human body and the exoskeleton.

In order to evaluate the algorithm performance and adjust the parameters of the algorithm, the following two formulas were introduced which are aimed at judging the pros and cons of the prediction effect:

2.3.1. Prediction Accuracy Rate (PR).

$$e_k(t) = y(t) - \hat{y}(t | t - b\Delta t),$$

$$\text{RMS}(e_k) = \sqrt{\frac{1}{n} \sum_{t=2}^n e_k(t)^2}. \quad (7)$$

$e_k(t)$ records the error between the predicted value and the actual value at each sampling point, and $\text{RMS}(e_k)$ (Root Mean Square error) characterizes the degree of deviation of the overall predicted value from the actual value. The greater the $\text{RMS}(e_k)$ is, the greater the deviation of the predicted data from sensor data and the worse the predicted performance. By normalizing the $\text{RMS}(e_k)$, the prediction accuracy rate (PR) is defined as

$$\text{PR}(y(t), e_k(t)) = 1 - \frac{\text{RMS}(e_k(t))}{\text{RMS}(y(t))} \times 100\%. \quad (8)$$

2.3.2. Smooth Factor (SF).

$$\text{SF}(\hat{y}(t)) = \frac{1}{t_{\text{end}}(\max(\hat{y}(t)) - \min(\hat{y}(t)))} \int_0^{t_{\text{end}}} |f(t) - \hat{y}(t)|. \quad (9)$$

In formula (9), t_{end} is the duration of the algorithm, and a 5 Hz low-pass filter is used to filter $y(t)$ to obtain $f(t)$. The parameter smooth factor (SF) indicates whether the data is smooth or not. In the course of human gait walking, the predicted sensor data is smooth and stable to match the actual walking data. Therefore, the smaller the SF is, the closer the predicted sensor data is to the sensor data measured by the actual gait walking. Since the data actually collected in the algorithm is discrete, when calculating the SF, formula (12) is usually discretized to the following formula:

$$\text{SF}(\hat{y}(t)) = \frac{1}{N(\max(\hat{y}(t)) - \min(\hat{y}(t)))} \sum_{i=0}^N |f(i) - \hat{y}(i)|. \quad (10)$$

The parameters affecting the prediction effect of the Takens algorithm are as follows: (i) the predicted duration (T), (ii) historical reconstruction delay vector length (P), (iii) optimal matching reconstruction vector number (M), and (iv) embedding dimension (N).

We introduced PSO (Particle Swarm Optimization) on the basis of Takens prediction algorithms, which is a swarm intelligence optimization algorithm in addition to an ant colony algorithm and a genetic algorithm in the field of computer intelligence. It originated from the study of bird predation behavior, and the basic idea is to solve and cooperate with individuals in the group to achieve the search for the optimal solution in a complex space [16]. Compared with other intelligent adjustment algorithms [17–18], such as derivative-based or derivative-based (or gradient-based) (e.g., backpropagation (BP), Levenberg-Marquardt (LM), Kalman filter, least square methods, and sliding-mode learn-

ing algorithm), hybrid learning methods, it belongs to the derivative-free method and does not need to update complex parameter equations, making it more suitable for nonlinear output situations. Meanwhile, it has the advantages of strong global search ability and easy implementation and has strong convergence and robustness in the process of solving. In the PSO algorithm, each particle represents a potential solution, and the velocity of the particle represents the direction and distance of each potential solution, which can be used to seek the optimal value in the multidimensional space.

The PSO-Takens algorithm flow chart is shown in Figure 3.

Step 1. Initialize a swarm of particles (population size m), including the random position and velocity, and limit the upper and lower limits of the particle's velocity and position. The optimized parameters in the Takens algorithm are as follows: historical vector dimension P , optimal matching number M , and embedding dimension N ; therefore, the population of particles is $\{x_1, x_2, x_3 \dots X_m\}$, and the position of the i -th particle is $\{XP_i, XM_i, XN_i\}$, The velocity of the i -th particle is $\{VP_i, VN_i, VM_i\}$.

Step 2. Use the evaluation function to calculate the fitness of each particle.

Step 3. For each particle, comparing its fitness with the optimal value $pbest$, it has passed through. If the current value is better than $pbest$, the $pbest$ will be set to the current value, and the position of $pbest$ is set as the current position in the n -dimensional space.

Step 4. For each particle, comparing its fitness value with the best position $gbest$, all particles passed through. If the current value is better, the current position will be regarded as the best position $gbest$.

Step 5. After updating the local and global optimal values of the current iteration, the particle adjusts its velocity and position through the following formulas.

$$V_{i+1} = V_i + c_1 \times \text{rand}() \times (pbest_i - x_i) + c_2 \times \text{rand}() \times (gbest_i - x_i),$$

$$x_{i+1} = x_i + V_{i+1}. \quad (11)$$

When the inertia factor is added, the velocity expression becomes

$$V_{i+1} = \omega V_i + c_1 \times \text{rand}() \times (pbest_i - x_i) + c_2 \times \text{rand}() \times (gbest_i - x_i). \quad (12)$$

Step 6. If the condition to terminate the iteration is not reached, go to Step2, until the iteration termination condition is met.

According to the specific issue, the iteration termination condition is generally to reach the maximum iteration number G_k or the optimal value of the evaluation function to

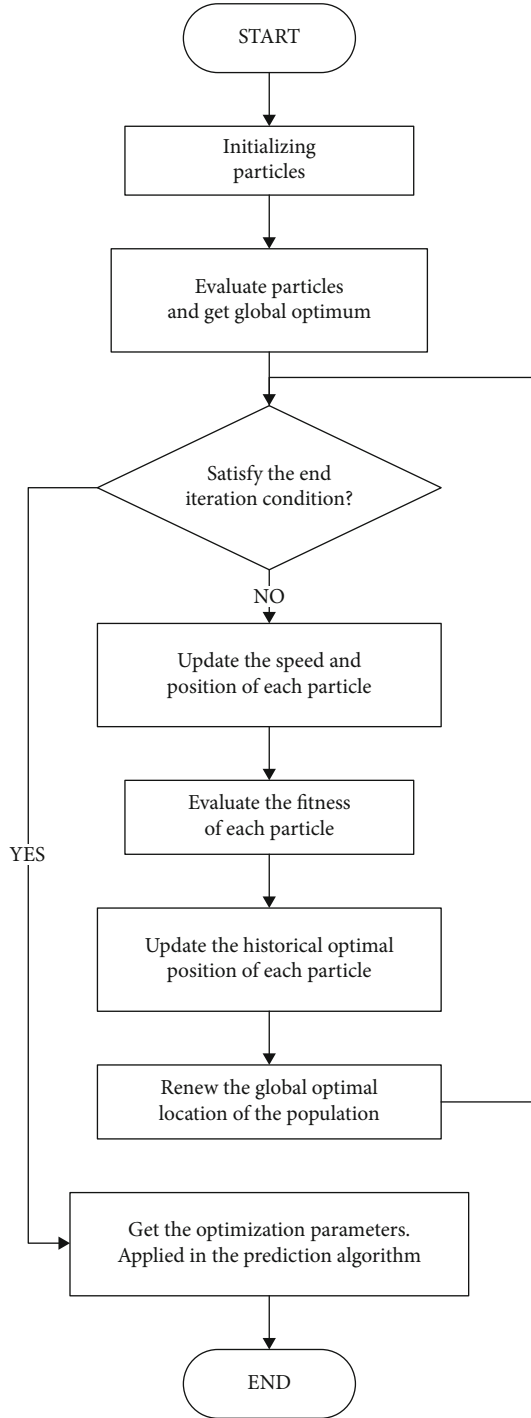


FIGURE 3: Flow chart of the PSO-Takens algorithm.

meet the established threshold. The condition for the algorithm to stop iteration is as follows: the number of iterations over 50, or the fitness calculated by the evaluation function over 100. The fitness (Y_{ef}) evaluation function expression is follows:

$$Y_{ef} = w_1 \times \frac{1}{SF} + w_2 \times PR. \quad (13)$$

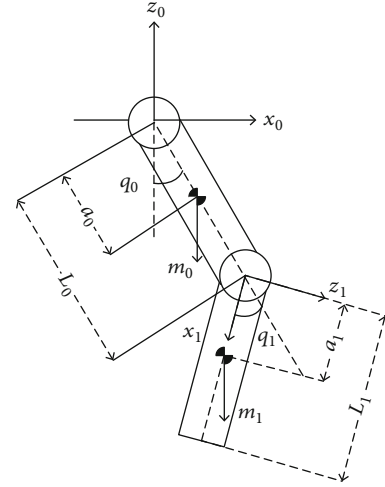


FIGURE 4: Lower extremity exoskeleton leg's simplified second link kinetic model.

In formula (15), w_1 and w_2 are constants (0.5), which means that we believe that the accuracy of data is as important as the smooth factor in the human lower limb prediction algorithm. The goal of the PSO-Takens prediction algorithm is to adjust the complex parameter tuning problem of the Takens algorithm in gait prediction, so as to achieve the optimal prediction effect.

2.4. Follow-Up Control Design

2.4.1. Dynamic Model of Swing Leg. The movement of the lower limbs of the human body can be regarded as a combination of swing phase and support phase. Usually, the swing phase accounts for 60% of the gait cycle and the support phase accounts for 40% [19]. For the swinging phase, the swinging legs of the human body can be regarded as an inverted two-link with uniform mass distribution. During the support phase, the leg and the upper body of the human body can be regarded as three links fixed at the bottom. This division method can simplify the complexity of torque calculation in inverse dynamics, making it easy to implement on embedded systems.

The lower limb-assisted exoskeleton single leg model is shown in Figure 4. L_0 and L_1 are the length of thigh and shank, and the ankle joint is omitted.

Assuming that the mass distribution of the exoskeleton lower limb members is uniform, the centroid coordinates $G_i(x_i, z_i)$ of the exoskeleton thigh and shank are as follows:

$$\begin{cases} x_0 = a_0 \sin q_0, \\ z_0 = -a_0 \cos q_0, \end{cases} \quad (14)$$

$$\begin{cases} x_1 = L_0 \sin q_0 + a_1 \sin q_1, \\ z_1 = -(L_0 \cos q_0 + a_1 \cos q_1). \end{cases} \quad (15)$$

The kinetic energy of the two connecting rods of lower limb swing is the sum of the rotational kinetic energy and

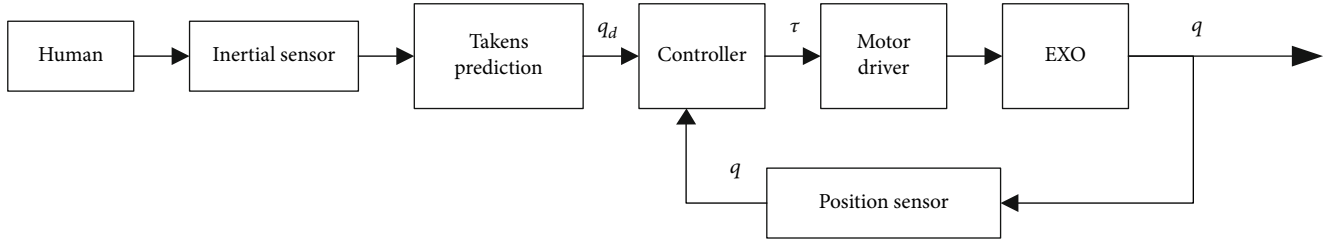


FIGURE 5: Follow-up control strategy block diagram of the exoskeleton.

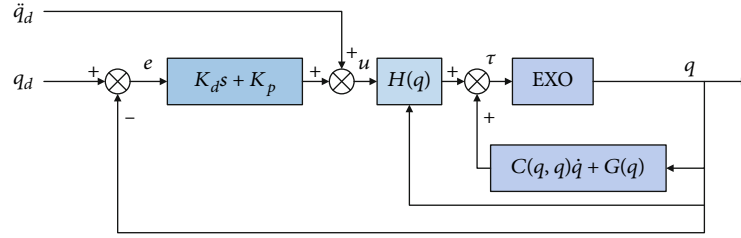


FIGURE 6: Control block diagram of calculate torque method.

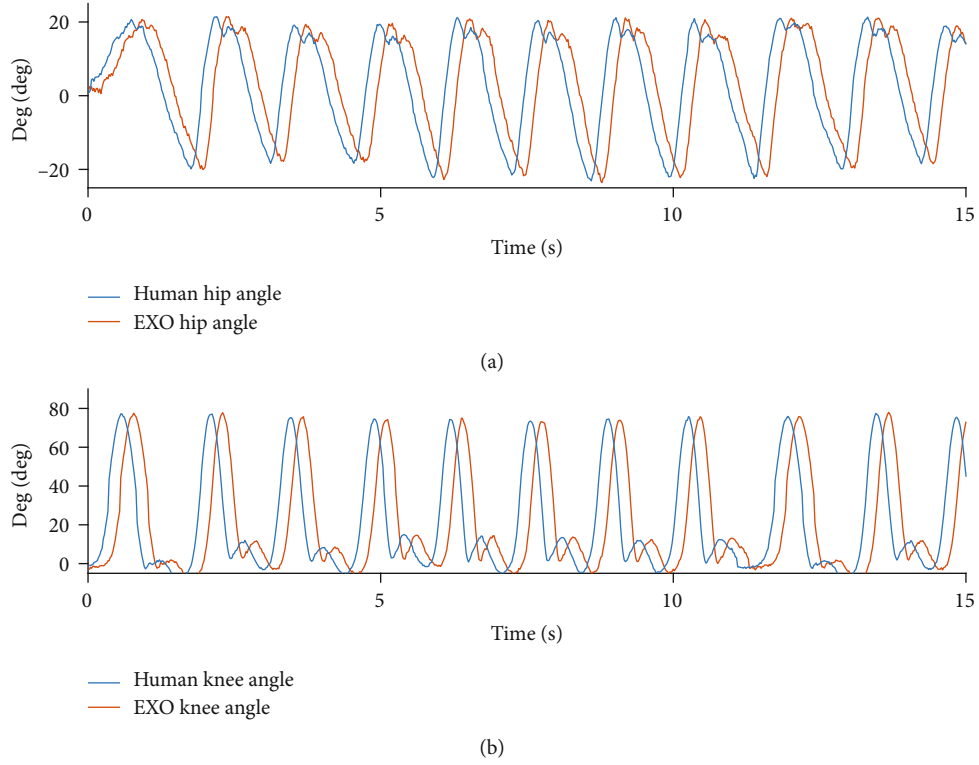


FIGURE 7: Joint angle deviation between human and exoskeleton.

kinetic energy of the connecting rod, which can be expressed as follows:

$$E = \frac{1}{2} \left(\sum_{i=0}^1 I_i \ddot{q}_i + m_i (\dot{z}_i^2 + \dot{x}_i^2) \right). \quad (16)$$

In formula (16), I_i is the moment of inertia when the exoskeleton is the i -th lower extremity connecting rod rotating

around the sagittal plane of the joint. The expression is $I_i = (1/3)m_i L_i^2$, and \ddot{q}_i is the angular acceleration of the joint. \dot{z}_i and \dot{x}_i are the velocities of the linkage in the x and z directions, and the total potential energy of the lower limb linkage is as follows:

$$P = \sum_{i=0}^1 m_i g z_i. \quad (17)$$

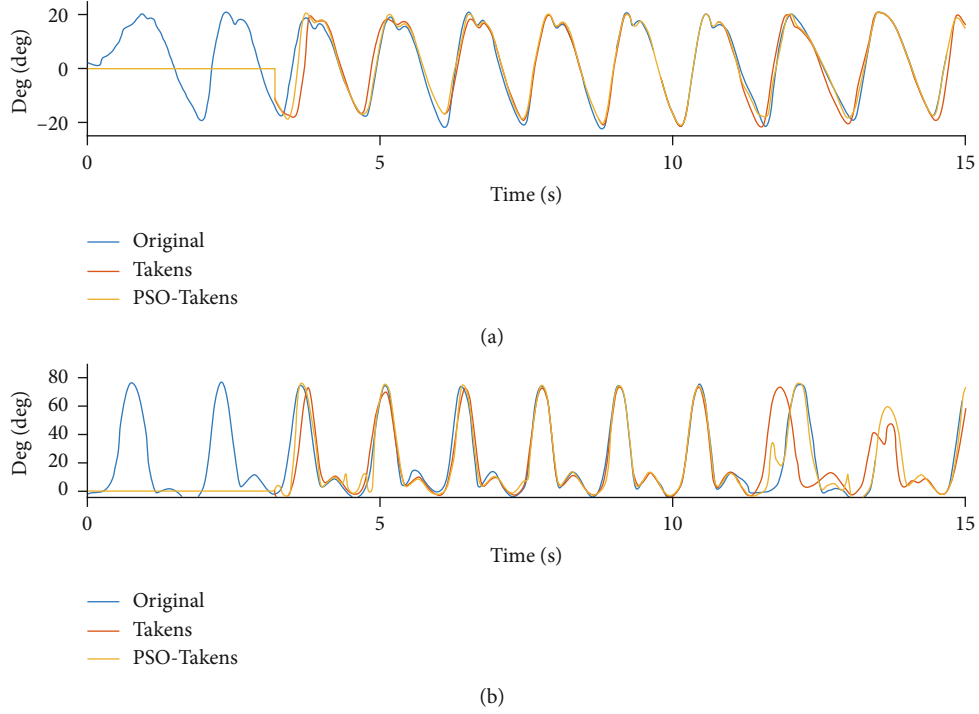


FIGURE 8: Comparison of Takens and PSO-Takens in prediction of nonstandard gait data. (a) Comparison of hip joint prediction angle. (b) Comparison of knee joint prediction angle.

By defining the Lagrangian function $L = E - P$ and the Lagrangian formula, the expression of the joint torque in the two-link can be expressed as

$$H(q)\ddot{q}_i + C(q, \dot{q})\dot{q} + G(q) = \tau_i. \quad (18)$$

In formula (18), τ_i is the joint torque between the links, $H(q)$ is the inertia matrix, $C(q, \dot{q})$ is the friction matrix, and $G(q)$ is the gravity vector. The expressions of the matrix $H(q)$, $C(q, \dot{q})$, and the gravity vector $G(q)$ are as follows:

$$\begin{aligned} H(q) &= \begin{bmatrix} \frac{7}{12}m_0l_0^2 + m_1l_1^2 & m_1l_0l_1 \cos(q_0 + q_1) \\ 0 & \frac{7}{12}m_1l_1^2 \end{bmatrix}, \\ C(q, \dot{q}) &= \begin{bmatrix} 0 & -m_1l_0l_1 \sin(q_0 + q_1) \\ -m_1l_0l_1 \sin(q_0 + q_1) & 0 \end{bmatrix}, \\ G(q) &= \begin{bmatrix} \frac{1}{2}m_0gl_0 \sin q_0 + m_1gl_0 \sin q_0 \\ \frac{1}{2}m_1gl_1 \sin q_1 \end{bmatrix}. \end{aligned} \quad (19)$$

2.4.2. Follow-Up Control Strategy. After obtaining the inverse dynamics model of the joint, we use the prediction angle obtained by the prediction algorithm to do Simulink simulation of the two-link model in MATLAB. The follow-up control strategy block diagram is shown in Figure 5.

TABLE 1: Parameter changes of hip joint prediction algorithm under nonstandard gait data.

#	T	P	M	L	PR	SF	Fitness
Takens	10	80	10	300	73.23%	0.2226	38.86
PSO-Takens	10	26	7	644	87.74%	0.2196	46.14

TABLE 2: Parameter changes of knee joint prediction algorithm under nonstandard gait data.

#	T	P	M	L	PR	SF	Fitness
Takens	10	80	10	300	71.80%	0.2317	38.05
PSO-Takens	10	48	12	536	85.75%	0.2388	44.97

The input of the control system is the angle of the human joint q_d and the angle sensor placed on the exoskeleton q , and the joint torque is as the output of the controller and outputted by the motor driver. It is not necessary and convenient to install an angle sensor on the lower limbs of the human body. Since the angle deviation between the exoskeleton and the human body is usually a constant value, this means we can predict the angle of the human body through the position sensor of the exoskeleton, instead of placing the inertial sensor outside the human body, which is the role of the predict algorithm played in follow-up control strategy.

For the angle data predicted by an algorithm, we used the calculated torque method for systematic control, aiming at making the lower extremity two-link of the exoskeleton follow the changing joint curve predicted by the algorithm. Compared with PD control and PD control with gravity compensation terms, the calculated torque method changes

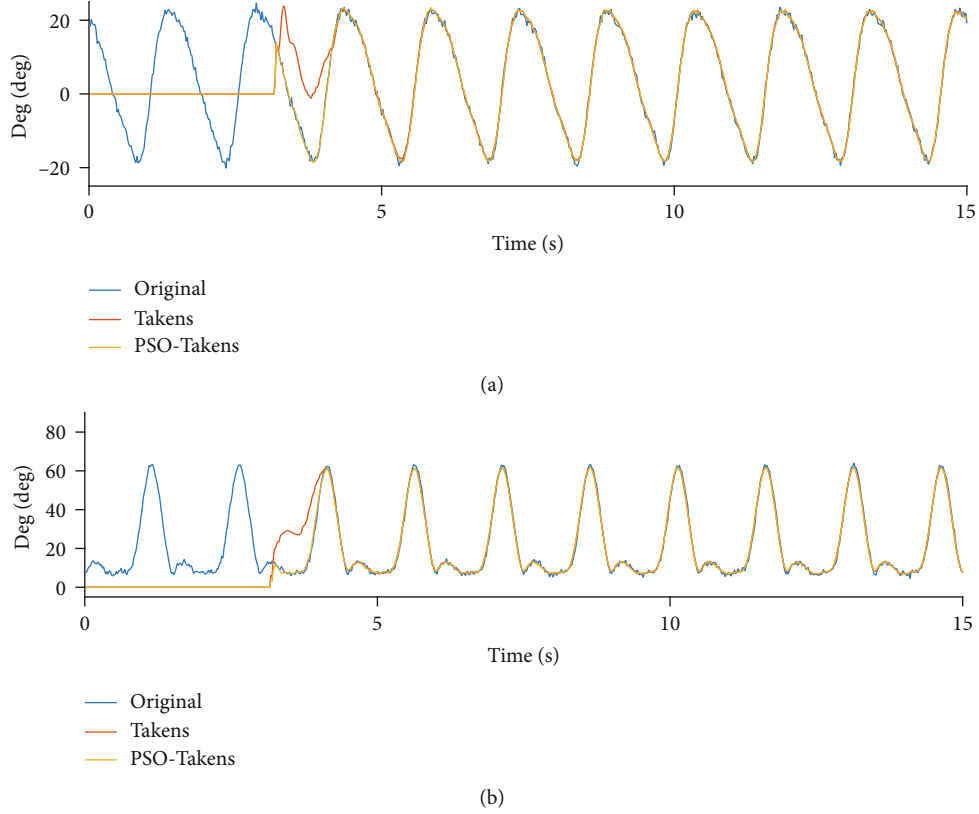


FIGURE 9: Comparison of Takens and PSO-Takens in prediction of standard gait data. (a) Comparison of hip joint prediction angle. (b) Comparison of knee joint prediction angle.

the lower extremity joints of the exoskeleton into a linear time-invariant system that is easier to control due to the introduction of nonlinear compensation. Huo [20] proved the stability of the control method. Based on the inverse dynamics equation, the system can follow the target position well. The control block diagram of this method is shown in Figure 6.

The torque expression calculated by the system control output to the hip and knee joints is as follows:

$$\tau = H(q)(\ddot{q} + K_d\dot{e} + K_p e) + C(q, \dot{q})\dot{q} + G(q). \quad (20)$$

In formula (20), the parameters $H(q)$, $C(q, \dot{q})$, and $G(q)$ are the matrixes of inverse dynamics in the two-link model; K_d and K_p are diagonal matrices, which are proportional and differential terms for the tracking error e .

3. Results and Discussion

3.1. Experimental Results

3.1.1. Prediction Algorithm Experiment Result. Figure 7 represents the sensor data of the motion angle data of the joints in the continuous walking movement. Compared with the sensor data of the wearing exoskeleton, we found that the angle detected from the exoskeleton and the actual motion angle detected from the human body are generally delayed by 200 ms, about 10 sampling cycles. Therefore, in a prediction algorithm, the sensor data at 200 ms after the current sampling point will be predicted based on EXO historical

TABLE 3: Parameter changes of hip joint prediction algorithm under standard gait data.

#	T	P	M	L	PR	SF	Fitness
Takens	10	120	10	900	77.52%	0.2133	41.10
PSO-Takens	10	69	3	767	88.19%	0.2091	46.48

TABLE 4: Parameter changes of knee joint prediction algorithm under standard gait data.

#	T	P	M	L	PR	SF	Fitness
Takens	10	120	15	900	87.14%	0.2063	45.99
PSO-Takens	10	77	15	1296	95.50%	0.2051	50.18

data. The angle sensor data obtained by three participants (65 ± 5 kg, 170 ± 5 cm) continuously walking at a speed of 3.6 km/h on a treadmill was recorded and predicted. None of the participants reported healthy problems in the previous three months.

From the analysis of the accuracy of the prediction angle data shown in Figure 8 and Tables 1 and 2, for both joints, the PSO-Takens has increased the accuracy of the prediction nearly $14.23 \pm 0.5\%$ ($n = 3$, two-sided paired t -test, $p = 0 < 0.05$), while the smoothing factor is almost unchanged. Due to the fact that the algorithm needs to store historical data, the algorithm will not immediately predict when the algorithm starts to execute. The area where the predicted value is 0 is the dead zone.

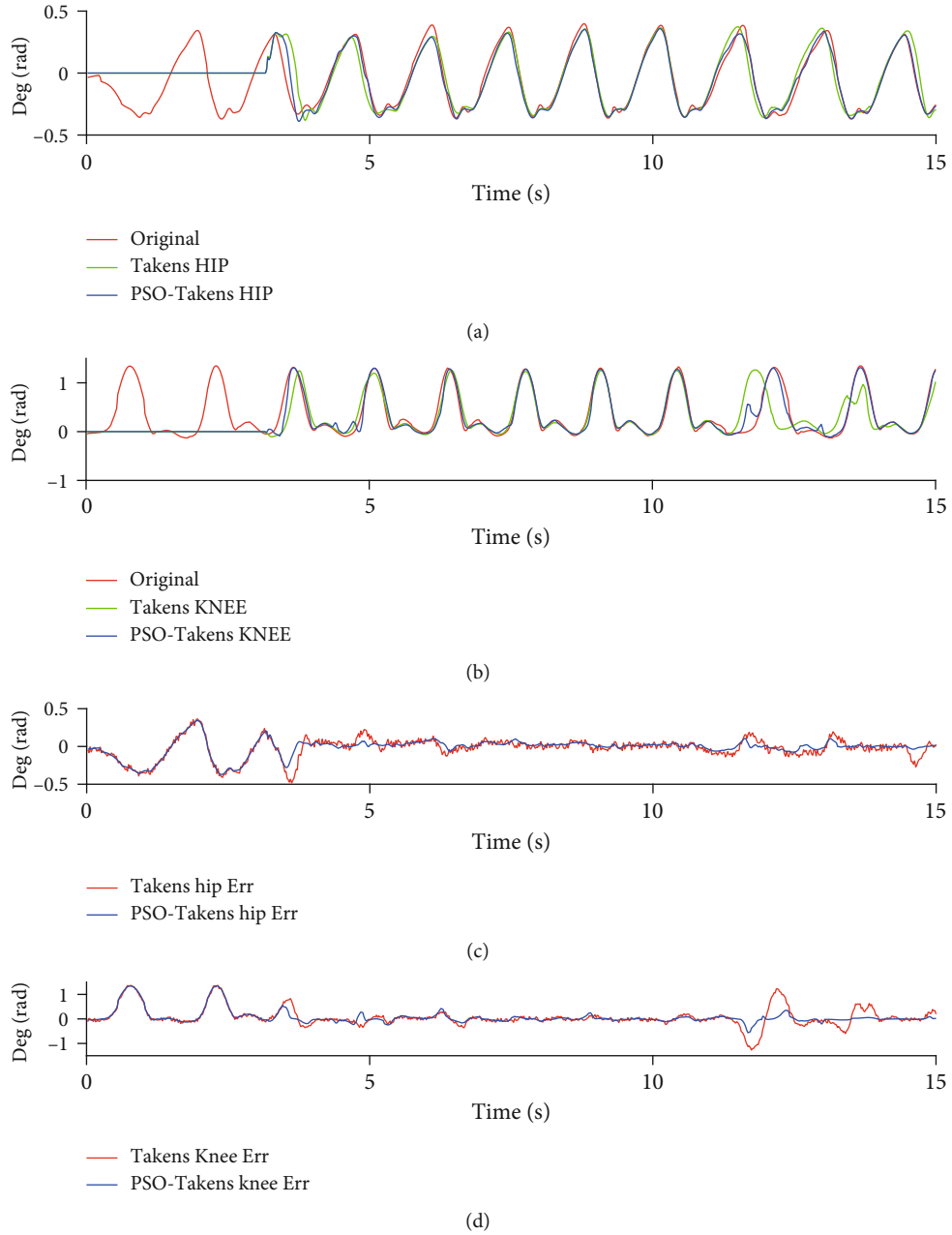


FIGURE 10: System joint angular output (a). System hip joint angle output (b). System knee joint angle output (c). Hip joint follow control angle error (d). Knee joint follows control angle error.

Figure 9 and Tables 3 and 4 show the performance of the two prediction algorithms under the standard and fixed frequency of the joint angle (data from [21]). The prediction accuracy in Tables 3 and 4 shows that the accuracy of the prediction is improved (hip joint from 77.85% to 88.19%, knee joint from 87.14% to 95.50%), while the smooth factor remains almost unchanged. Therefore, after the PSO-Takens algorithm is executed, for the movement data of the lower limbs, the prediction algorithm with optimized parameters becomes accurate and stable.

3.2. Prediction Algorithm Experiment Result. Since the current pose and the motion-sensing data of the previous period

can only reflect the current motion state, to achieve better performance, the exoskeleton system needs to predict the posture data of the following period of time. After processing and analyzing the predicted pose data, the motion intention was judged in advance. By predicting the motion angle of the human body and introducing it into the control of the exoskeleton, the human-machine coordinated motion process of the exoskeleton can be more flexible and smoother.

In order to compare the performance of prediction algorithms and algorithm optimization in simulation, the joint data during motion and the predicted joint data were imported into the two-link model simulation system for a comparative experiment, and the evaluation method of the

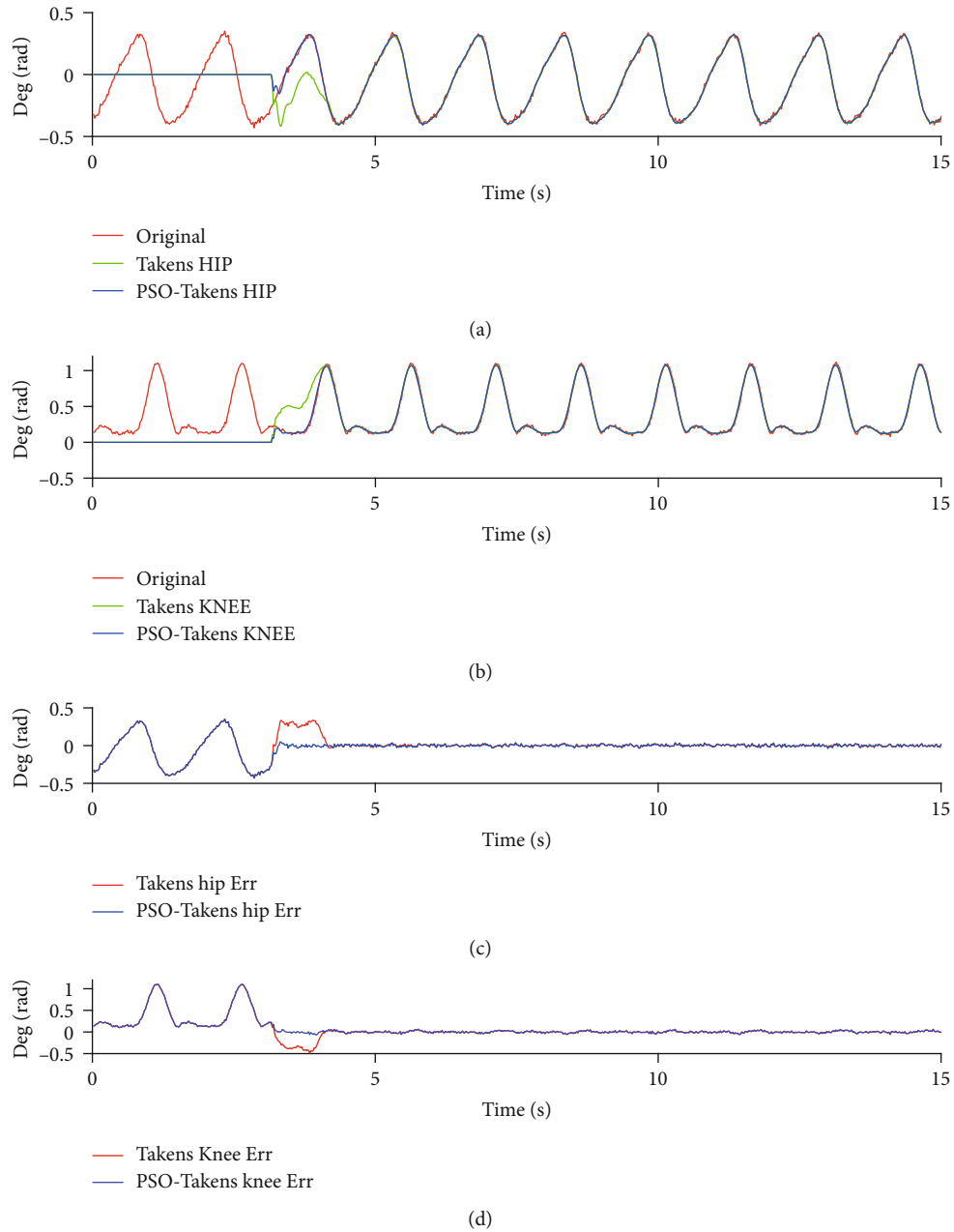


FIGURE 11: System joint angle output for standard gait data input (a). Hip joint angle output (b). Knee joint angle output (c). Hip joint follow control angle error (d). Knee joint follows control angle error.

algorithm mentioned in the second section was used to evaluate the performance of the algorithm. The experimental results are shown in Figure 10.

It can be found that the predicted data lags behind the sensor data of human motion when the system's lower limb joints are controlled to follow the human body at the beginning. Over time, the torque control algorithm with gravity compensation can control the two links to the angle predicted by the two prediction algorithms. When the state or frequency of the motion changes, the prediction error occurs, but the PSO-Takens algorithm shows a better antidisturbance ability, and the maximum error during movement is also better than the original algorithm.

It can be found from Figure 10, in the case of control the movement of the exoskeleton following the lower limbs of the human body without using prediction algorithm, due to the large deviation of the angle, a large interaction force of human-computer interaction force may occur. If the method of prediction is adopted in the control loop of the exoskeleton, the movement of the exoskeleton will be at an angle different from the actual movement of the human body at the initial stage. With the increase of time, the two links of the lower limbs of the exoskeleton gradually follow the movement of the human body and maintain a high consistency with the movement of the human body. It is undeniable that there is still a deviation between the exoskeleton and the

human body. According to formula (8), the accuracy of the prediction using the Takens method for hip joint and knee joint is 65.10% and 66.71%, while the PSO-Takens has increased to 82.13% and 83.03%. When the gait becomes stable, the error of Takens algorithms ranges from -0.13 rad to 0.18 rad (hip) and from -0.17 rad to 0.11 rad (knee). While the error of PSO-Takens ranges from -0.04 rad to 0.07 rad (hip) and from -0.05 rad to 0.09 rad (knee).

For standard gait data with a fixed frequency, inappropriate parameters of the Takens algorithm will have obvious errors at the beginning of the prediction, while the PSO-Takens algorithm will not have this problem. Figure 11 shows that the error of Takens algorithms ranges from -0.14 rad to 0.10 rad (hip) and from -0.12 rad to 0.09 rad (knee), the error of PSO-Takens algorithms ranges from -0.05 rad to 0.06 rad (hip) and from -0.03 rad to 0.04 rad (knee). According to formula (8), the Takens algorithm prediction accuracy rates of the hip and knee joints are calculated as 72.10% and 83.73%, and the PSO-Takens algorithm prediction accuracy rates are 86.07% and 94.01%, which also shows the PSO-Takens algorithm performs better in predicting such data.

4. Discussion

In the case of ignoring the elasticity coefficient and inertia coefficient, the human-computer interaction model can be simplified; in this situation, Racine [22] proposed the interaction force can be calculated as follows:

$$F_e = K(q_h - q_e). \quad (21)$$

In formula (24), F_e is the human-machine interaction force and q_h, q_e represent the human joint angle and EXO joint angle detected by a sensor, while K equals K_0/L_0 (Nm/rad-m), L_0 indicates the distance between the measurement point and the joint rotation center, and K_0 is a constant that varies from the system. Formula (24) means that if there is no angle deviation between the human body and the exoskeleton, it can be considered that the exoskeleton will not hinder the movement of the human body in the walking state of the human body, and the human-machine interaction force will close to 0.

The difficulty of the exoskeleton system controller is how to control the system to move to the target posture of the human body at the next moment. Relying on the installation of an inertial sensor on the exoskeleton has hysteresis and cannot judge the posture of the human body at the subsequent moment. This paper is devoted to the realization of human body motion prediction angle algorithm and introduces it to the control system. The simulation experiment results show that, for the angle prediction of exoskeleton, the PSO-Takens algorithm has the function of improving algorithm parameters. Whether it is linear prediction based on ARMA model LMS, RLS, or DMP [23], the algorithm parameters need to be optimized according to different systems. The PSO-Takens algorithm can deal with periodic during gait walking data, and when the body's movement law changes, such as from walking to jumping or stopping, the

accuracy of the prediction algorithm will be reduced, which is the point that the prediction algorithm needs to be improved in the future study.

5. Conclusions

In this paper, a human motion capture system is designed to acquire human walking joint data, and a method for optimizing parameters of Takens nonlinear prediction algorithm is proposed. Compared with the original Takens prediction algorithm, the prediction angle obtained by the improved prediction PSO-Takens algorithm is more closely related to the actual motion angle data of the human body, with a smaller error rate and smooth features. When the predicted sensor data from the PSO-Takens algorithm were applied to the joint angle prediction algorithm of the lower extremity exoskeleton, it can improve the accuracy of prediction and enhance the adaptability of the exoskeleton and human body.

Data Availability

The EXO and predicted sensor data based on different algorithms used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is partly supported by the Natural Science Research General Program of Higher Education of Jiangsu Province (16KJB510040). We would like to thank Zhicheng Qu and Wei Wang for the help with control design.

References

- [1] J. Wesslén, "Exoskeleton exploration: research, development, and applicability of industrial exoskeletons in the automotive industry, [M.S. thesis]," *Master's Thesis, School of Engineering in Jönköping, Jönköping, Sweden*, 2018.
- [2] 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," *American Journal of Physical Medicine & Rehabilitation*, vol. 91, no. 11, pp. 911–921, 2012.
- [3] D. P. Yang, L. Jiang, J. D. Zhao, and H. Liu, "Prosthetic hand control based on intelligent control of EEG," *Journal of Jilin University*, vol. 38, no. 5, pp. 1225–1230, 2008.
- [4] Y. H. Yin, Y. J. Fan, and L. D. Xu, "EMG and EPP-integrated human machine interface between the paralyzed and rehabilitation exoskeleton," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 542–549, 2012.
- [5] A. M. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: challenges and state-of-the-art," *Transaction-son Robotics*, vol. 24, no. 1, pp. 144–158, 2008.
- [6] W. Wei, S. Zha, Y. Xia, J. Gu, and X. Lin, "A hip active assisted exoskeleton that assists the semi-squat lifting," *Applied Sciences*, vol. 10, no. 7, p. 2424, 2020.

- [7] R. Huang, H. Cheng, H. Guo, Q. Chen, and X. Lin, "Hierarchical interactive learning for a human-powered augmentation lower exoskeleton," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 257–263, Stockholm, 2016.
- [8] P. Zhou, Y. Yang, L. Ma, and Y. Shu, "Human gait awareness system design of exoskeleton robot," *Sensor and Microsystem*, 2016.
- [9] Z. Cao, X. Ye, Y. C. Wang, L. J. Cheng, and H. S. Zhu, "Study of swing leg follow-up control of exoskeleton based on Kalman prediction," in *Science Technology and Engineering*, 2016.
- [10] F. Ernst, A. Schlaefer, S. Dieterich, and A. Schweikard, "A fast lane approach to LMS prediction of respiratory motion signals," *Biomedical Signal Processing and Control*, vol. 3, no. 4, pp. 291–299, 2008.
- [11] P. R. Diniz, *Adaptive filtering: algorithms and practical implementation second edition*, Kluwer Academic Publisher Norwell, 2004.
- [12] G. V. Moustakides, "Study of the transient phase of the forgetting factor RLS," *IEEE Transactions on Signal Processing*, vol. 45, no. 10, pp. 2468–2476, 1997.
- [13] Y. Yang, P. Zhou, L. Ma et al., "Gait recognition and trajectory prediction of lower limb load exoskeleton," in *2016 35th Chinese Control Conference (CCC)*, Chengdu, China, July 2016.
- [14] P. Duan, S. Li, Z. Duan, and Y. Chen, "Bio-inspired real-time prediction of human locomotion for exoskeletal robot control," *Applied Sciences*, vol. 7, no. 11, p. 1130, 2017.
- [15] T. Hayashi, H. Kawamoto, and Y. Sankai, "Control method of robot suit HAL working as operator's muscle using biological and dynamical information," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alta., Canada, August 2005.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, 1995.
- [17] K. Gao, Z. Cao, Z. C. Le Zhang, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 875–887, 2019.
- [18] J. Wang and T. Kumbasar, "Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods," *Acta Automatica Sinica*, vol. 6, no. 1, pp. 247–257, 2019.
- [19] A. D. Grant, "Gait Analysis: Normal and Pathological Function," *Jama*, vol. 304, no. 8, p. 907, 2010.
- [20] W. Huo, *Robot Dynamics and Control*, Higher Education Press, Beijing, 2005.
- [21] Y. L. Han and X. S. Wang, "Research on the biomechanics of human walking lower limbs," *Science China: Technical Sciences*, vol. 41, no. 5, pp. 592–601, 2011.
- [22] J. L. C. Racine, *Control of a lower extremity exoskeleton for human performance amplification*, University of California, Berkeley, 2003.
- [23] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.