Sensors and Materials, Vol. 37, No. 3 (2025) 1053–1072
MYU Tokyo

S & M 3974

Gait Phase Recognition Based on Lower Limb Surface Electromyography Signals Using a Novel Algorithm to Improve Motion Intention Recognition Accuracy

Bing Xie,[1,2] Yuming Qi,[3*] and Wenhua Gao[1**]

[1]Tianjin University of Science and Technology, Tianjin 300222, China
[2]Tianjin Key Laboratory of Integrated Design and Online Monitoring of Light Industry & Food Engineering Machinery and Equipment, Tianjin 300222, China
[3]Tianjin University of Technology and Education Robotics and Intelligent Equipment Research Institute, Tianjin 300222, China

(Received May 2, 2024; accepted December 2, 2024)

Keywords: gait recognition, surface electromyography, classification recognition, support vector machine, cuckoo-search-based support vector machine

Gait phase recognition plays a key role in the motion control of exoskeleton robots. Surface electromyography (sEMG) is predictive and contributes to accurate gait phase recognition. To address the challenge of low accuracy of intention recognition in exoskeleton robots, a hybrid algorithm using an improved cuckoo-search-based support vector machine (ICS-SVM) was proposed to achieve accurate gait phase recognition. First, the raw sEMG signals from ten subjects were collected through gait experiments. Second, time-domain features including mean absolute value, waveform length, variance, and root mean square were extracted from the sEMG signals. Third, the support vector machine (SVM) used in this study is the most common model for intention recognition. However, the SVM has the problem of being sensitive to parameter tuning. A cuckoo search (CS) algorithm was applied to optimize the penalty factor and kernel function parameter of the SVM to accelerate convergence. An information-sharing mechanism, a local enhancement operator, and a new way to build a bird's nest are introduced to overcome the low search efficiency of the original algorithm and its tendency to fall into local optimal solutions. Experiments showed that the algorithm model combines the advantages of the ICS algorithm and the SVM model, and can accurately distinguish seven gait phases, with an average recognition accuracy of 95.125%, which is higher than those of the SVM (92.177%) and CS-based SVM (CS-SVM) (94.170%) models. This study will provide technical support for the development of intelligent medical and exoskeleton robotics fields.

1. Introduction

Walking in humans represents a complex, periodic motion. This process is predominantly governed by the interaction of muscles, joints, and bones, orchestrated by the nervous system,[1,2] which mirrors the motion dynamics inherent to the human anatomy. The assessments of the

*Corresponding author: e-mail: chigym@163.com
**Corresponding author: e-mail: gaowenhua@tust.edu.cn
https://doi.org/10.18494/SAM5123

ISSN 0914-4935
https://myukk.org/

locomotor performance and control of exoskeletons rely significantly on the analysis of gait phases,[3] especially crucial in the motion control of exoskeleton robots, which is the basis for developing control strategies and realizing human–robot integration.[4] Currently, researchers collect and detect gait information through a variety of sensors, such as photoelectric sensors,[5–7] inertial measurement units,[8] vision sensors,[9] joint angle sensors,[10–12] force platforms,[13] and foot pressure insoles.[14]

Surface electromyography (sEMG) signals consist of the temporally and spatially integrated superposition of muscle fiber motor unit action potentials,[15] which contain movement-related information such as muscle activity intensity[16,17] and activity time.[18] sEMG signals are generated before limb movement, which compensates for the delay problem in data processing and pattern recognition.[19] In addition, changes in sEMG signals are directly related to the levels of muscle contraction and activation, accurately reflecting the intent of the movement.[20,21] The use of human gait recognition technology using sEMG signals in exoskeleton research has been a hot research topic in recent years.[22] However, the non-smooth, weak, and low-frequency characteristics of sEMG signals, as well as the effects of various factors such as sensor noise and cardiac interference on signal quality, increase the difficulty of gait phase recognition.[23–25]

In this context, a variety of algorithmic models have been developed for processing and recognizing sEMG-based gait information. These models include neural networks,[26] hidden Markov models,[27] and linear discriminant analysis (LDA).[28] In particular, gait recognition approaches using the support vector machine (SVM) have shown remarkable effectiveness. SVM has been shown to outperform neural networks and other classifiers such as nonlinear logistic regression, multilayer perceptron (MLP), and LDA in classifying a wide range of gait patterns, demonstrating its superior classification ability.[29,30]

SVM algorithms have attracted attention for their excellent performance in classifying small samples and motion intentions, as well as for their strong generalizing ability. However, how to optimize the parameters of SVM is still a challenge. Several researchers have attempted to optimize the parameters and improve the classification accuracy using genetic algorithms, quadratic discriminant analysis, and other methods. For example, Zheng and Wang recognized the lower limb motion states by the SVM method, and in order to reduce the problem of recognition error during the motion state transition process, the quadratic discriminant analysis method was introduced, which was used to recognize the motion patterns in the state transition stage.[31] Zhang *et al.*[32] suggested the use of a genetic algorithm to refine the parameters of the SVM method, which improved the local search capabilities and convergence of the algorithm, and address the issue of low SVM classification accuracy. However, the fusion algorithm has difficulty in parameter tuning, which increases the complexity of the optimization process. In contrast to other methods, the cuckoo search (CS) algorithm has several advantages, including fast convergence, a minimalistic parameter set, and easy implementation. Research scholars optimize the parameters of SVM by introducing the CS optimization algorithm, and the gesture recognition experiments show that the average correctness rate reaches 90.28%, which solves the problems of the SVM algorithm, such as low convergence speed and low optimization accuracy. In addition, current research in the area of motor intent recognition mostly focuses on the analysis of upper limb movements. Compared with the hand and arm, the dynamics of lower

limb movements usually exhibit higher redundancy, which poses additional challenges for recognition and classification. The performance of existing lower limb motion intent recognition algorithms is not always satisfactory, owing to the more complex biomechanical properties of lower limb movements, as well as the signal noise and variability problems encountered in acquiring and processing lower limb electromyographic signals.

In this study, a new hybrid algorithm [improved CS-based SVM (ICS-SVM)] combining the improved CS and SVM is proposed to enhance the recognition accuracy of seven phases on the basis of lower limb sEMG signals. By applying the CS algorithm to optimize both the penalty and kernel function parameters (C and σ) of the SVM, the generalization capabilities and classification performance of the model were significantly improved. This study provides valuable technical insights and new perspectives for the application of exoskeletons and rehabilitation robots.

2. Materials and Methods

2.1 Gait phase segmentation

Human gait typically consists of two primary phases:[33] the stance phase and the swing phase. On the basis of the gait cycle outlined by Condie,[34] the gait phase can be subdivided into load response (LR), midstance (MST), terminal stance (TST), pre-swing (PSW), initial swing (ISW), mid-swing (MSW), and terminal swing (TSW).[35] This partitioning of gait temporal phases preserves rich gait information, which helps to provide a deeper understanding of gait variations during the gait cycle and, in turn, provides more fine-grained guidance for formulating robot control strategies. Therefore, in this study, we divide the gait cycle into seven phases and base the gait cycle on the displacement trajectories of the toe and heel. Figure 1 demonstrates the gait phase recognition based on marker displacement. The transition conditions for each gait phase are as follows: the LR to MST phase transition occurs when the right toe (R-toe) reaches a local minimum. When the right heel (R-heel) leaves the ground, it marks the transition from MST to TST. The TST to PSW phase transition occurs when the L-heel reaches a local minimum. The PSW phase transition occurs when the R-heel reaches a local minimum, and the ISW phase transition occurs when the R-heel reaches a local maximum. The MSW phase transition occurs when the R-toe is at the same level as the R-heel in the vertical direction. Finally, when the R-heel reaches a local minimum again, the TSW transitions to the LR phase of the next gait cycle.

2.2 Processing of sEMG signals

The sEMG is a nonstationary, time-varying signal whose frequency characteristics are mainly concentrated in the low frequency range. To obtain high-quality sEMG information, several preprocessing steps must be performed on the raw signal, including filtering, denoising, and eliminating sources of noise that may be introduced during the acquisition process. This process typically involves a number of key steps: first, a bandpass filter (cut-off frequency 20–

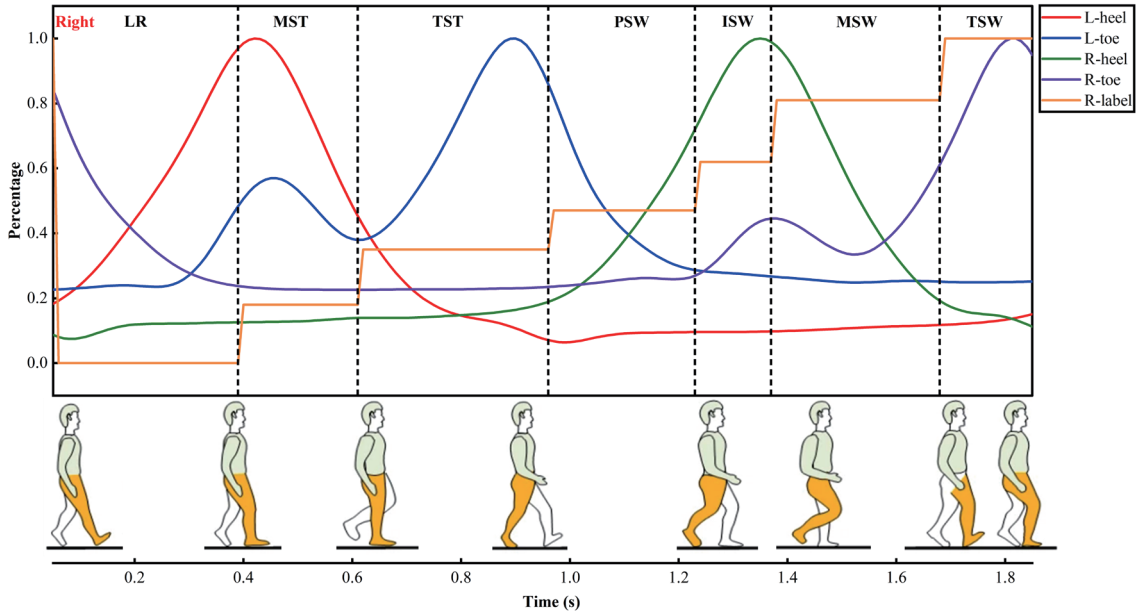

Fig. 1. (Color online) Recognition of gait phases based on marker displacements.

400 Hz, 4th-order Butterworth filter) is applied to remove noise outside the frequency range; second, the mean value of the signal is reduced by removing the mean value of the signal and performing full-wave rectification; third, a low-pass filter (cut-off frequency 6 Hz, 4th-order IIR filter) is used to further smooth the signal. The signal envelope was then normalized to allow objective feature comparisons between different individuals or muscles. A variety of normalization methods for sEMG signals have been proposed in the literature. In this study, the method described in the literature[36] was used, i.e., the normalization was based on the average amplitude of the rectified signal for each individual at a speed of 1.35 m/s.

In classification and recognition based on sEMG signals, the length of the signal data has a direct impact on the accuracy of the classification model.[37] To effectively segment sEMG signals, the neighborhood and overlapping windowing methods are commonly used.[38] In this study, we used a sliding analysis window with a length of 60 ms and an increment of 15 ms to continuously extract the feature set.[39] As shown in Fig. 2, the preprocessing of tibialis anterior (TA) muscle sEMG signals is demonstrated, which can effectively extract key features for subsequent analysis.

In the field of motor intention recognition, the feature selection and extraction of EMG signals are the core aspects of the whole recognition process.[40] The features of EMG signals are usually categorized into three main types: time-domain features, frequency-domain features, and time-frequency-domain features. Considering the computational efficiency, the time-domain features with low computational cost are selected as the input feature set of the classification model in this study. The selected time-domain features include the following:
Mean Absolute Value (*MAV*): reflects the contraction level of the muscle.[41]

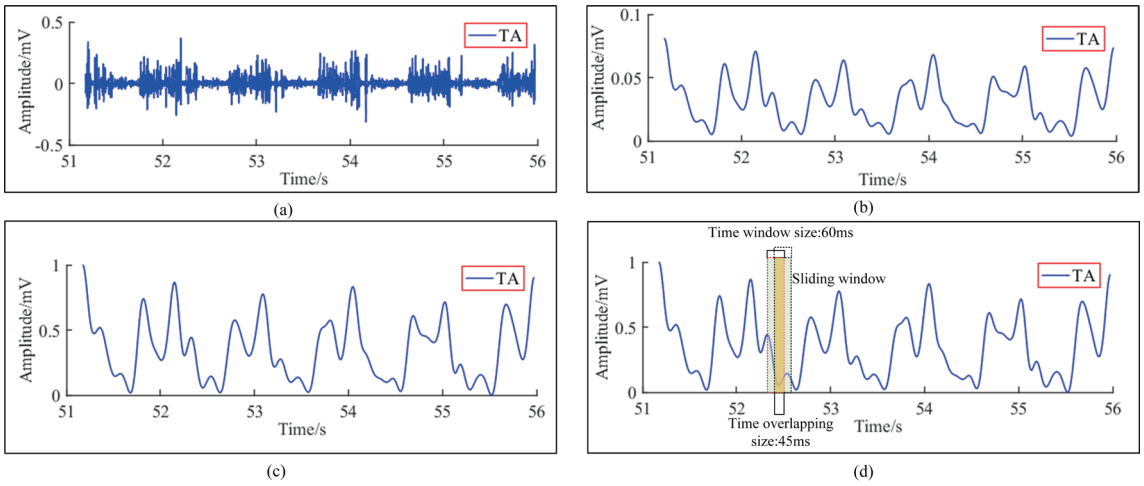

Fig. 2. (Color online) Preprocessing flowchart of TA muscle sEMG signals.

$$\overline{x}_i = \frac{1}{L}\sum_{k=1}^{L}\left|x_k\right| \quad i = 1,2,\cdots,M \tag{1}$$

Waveform Length (*WL*): represents the amplitude, *i* duration, and frequency of the sEMG signal. It is a measure of signal complexity.

$$l_i = \sum_{k=2}^{L}\left|x_k - x_{k-1}\right| \quad i = 1,2,\cdots,N \tag{2}$$

Variance (*VAR*): indicates the power of the measured sEMG signal.

$$\hat{x}_i = \frac{1}{L-1}\sum_{k=1}^{L}\left(x_k - \overline{x}\right)^2 \tag{3}$$

Root Mean Square (*RMS*): reveals the strength produced by muscles.

$$\tilde{x}_i = \sqrt{\sum_{k=1}^{L} x_k^2 \Big/ L} \quad i = 1,2,\cdots,M \tag{4}$$

Here, L represents the length of the sliding analysis window, x_k represents the k-th sample in the i-th window, M is the total number of collected sEMG signal samples, $\overline{x}$ is the average value, and L represents the sample length.

2.3 ICS-SVM gait classification algorithm

2.3.1 CS algorithm

A new metaheuristic search algorithm, the CS algorithm, was proposed by Yang and Deb in 2009, which was inspired by the brooding behavior of cuckoos and the Levy flight behavior of fruit flies.[42,43] Studies have shown that CS algorithms have advantages such as high convergence speed and stability in solving optimization problems compared with traditional genetic and particle swarm algorithms.[44] However, CS algorithms may encounter low convergence speed and insufficient local search accuracy in some cases. To overcome these limitations, researchers have proposed some adaptive parameter tuning strategies.[45] These strategies can dynamically adjust the algorithm parameters according to the current search state to improve the global search capability and local search accuracy of the algorithm. For example, Meng *et al.*[46] used a dynamic step factor function whose value is associated with the distance to the current optimal solution, which can effectively guide the search process to evolve toward a more optimal solution, thus rapidly improving the quality of the non-optimal solution and enhancing the overall search efficiency. The ICS algorithm proposed in this paper is presented below.

In the process of nest searching, the random walk search method in Levy's flight mode is introduced into the CS algorithm and updates the nest position on the basis of these three rules.

$$z_i^{(t+1)} = z_i^t + \beta \otimes L(\lambda) = z_i^t + \beta_0 \cdot (z_i^t - best) \cdot L(\lambda) \tag{5}$$

Here, $z_i^{(t+1)}$ and z_i^t are the positions of the i-th cuckoo nest in the t-th and $t + 1$ generations, respectively. β represents step control. $\otimes$ represents the entry-wise multiplications. $L(\lambda)$ is the Lévy distribution. best is the current optimal position. β_0 is constant, and in this paper, it is set to 0.01 for the parameter optimization in the CS algorithm.

The random number of the Lévy distribution is expressed as [35].

$$L(\lambda) = \frac{\sigma_u \cdot u}{|v|^{1/\lambda}} = \frac{u}{|v|^{1/\lambda}} \left(\frac{\Gamma(1+\lambda)\sin(\pi \cdot \lambda / 2)}{\Gamma((1+\lambda/2) \cdot \lambda \cdot 2^{(\lambda-1)/2})} \right) \quad (1 < \lambda \leq 3) \tag{6}$$

Here, Γ is the gamma function. u and v are independent random variables that follow a normal distribution with a mean of zero and a variance of one.

After the nest position is updated, the optimal degree of the nest is calculated by comparing the discovery probability with a random number. If the discovery probability is less than the random value, the worse nest is discarded and the nest position is reconstructed. The same number of new solutions is generated using a form of preference wandering.

$$z_i^{(t+1)} = z_i^t + r \cdot (z_j^t - z_k^t) \tag{7}$$

Here, $r \in (0,1)$, and z_j^t and z_k^t are the positions of two random nests at the t-th iteration.

2.3.2 ICS algorithm

In the CS algorithm, although Lévy flight is effective in exploring the global solution space, the traditional nest position update strategy relies mainly on the comparison between the current solution and the optimal solution, which limits the exploration of the diversity of the population in the early iteration phase of the algorithm, leading to a decrease in diversity. As the algorithm progresses into the middle and late iterations, the likelihood of convergence to a local optimum increases owing to the greater similarity between individuals. Similar search strategies can lead to a decrease in sample diversity, as described in the literature.[47] To overcome this problem, we introduced an information sharing mechanism while preserving the Levy flight properties. This mechanism enhances the diversity of the population by incorporating diverse interaction information into the position updating strategy and exploiting the information exchange between multiple solution sets of different evolutionary operators. Moreover, the periodicity and oscillation of the sine-cosine function are used to prevent convergence to local optima. The specific strategy is to randomly select three nest positions from the entire population and then reorganize each individual after each iteration, with the positive cosine function introduced during the reorganization process. The updating of the nest positions follows Eq. (9).

$$b = 2 - 2 \times t / \mathrm{iter}_{max} \tag{8}$$

$$z_i^{(t+1)} = \begin{cases} z_{p_3}^t + b \cdot \sin(2\pi r) \cdot \beta_0 \cdot L(\lambda) \cdot (best - z_{p_1}^t) & \text{if } r < 0.5 \\ z_{p_1}^t + b \cdot \cos(2\pi r) \cdot (best - \beta_0 \cdot L(\lambda) \cdot z_{p_2}^t) & \text{if } r \geq 0.5 \end{cases} \tag{9}$$

Here, $z_{p_1}^t$, $z_{p_2}^t$, and $z_{p_3}^t$ represent the three positions randomly selected from all nested positions in the t-th iteration. iter_{max} represents the maximum number of iterations.

Although the CS algorithm can escape local optima in late iterations using the Lévy flight strategy, its search precision depends primarily on the accumulation of small step sizes. This reliance limits the algorithm's accuracy and convergence rate, especially when tackling complex problems. A localized enhancement operator has been incorporated into the algorithm to allow for a more refined search around the local optimum. This is achieved by applying a perturbation strategy to the population, which helps to explore a wider solution space and potentially avoid premature convergence to a suboptimal result. The design of the perturbation step size is based on the difference between the global optimal solution and the current nest position. Specifically, the perturbation step size is set as follows.

$$s_i^t = \xi_1 \times (\xi_2 \cdot best - z_i^t) \tag{10}$$

$$\xi_1 = 2b / r_1 - b \tag{11}$$

$$\xi_2 = 2r_1 \tag{12}$$

At this time, the position of the nest can be updated using Eq. (13).

$$z_i^{(t+1)} = best - s_i^t \tag{13}$$

In addition, the method of generating new solutions in the algorithm was also improved in this study. The traditional CS algorithm uses a random wandering strategy to increase the diversity of new nest positions, but this method fails to preserve the optimal positions in the current population, leading to a decrease in search efficiency. To address this problem, we redefine Eq. (7) to ensure that the tracking and use of the current optimal solution are maintained while exploring the new solution space.

$$z_i^{(t+1)} = z_i^t + r_1 \cdot (z_j^t - z_k^t) + r_2 \cdot (best - z_i^t) \tag{14}$$

In summary, the proposed ICS algorithm, by introducing the information sharing mechanism, the local enhancement operator, and the new nest generation strategy, aims to increase the population diversity, strengthen the local search ability, and accelerate the convergence of the algorithm. The core procedures of the ICS algorithm are encapsulated in the pseudo code shown in Table 1.

Table 1
ICS algorithm.

Algorithm 1 Improve CS Algorithm	
1:	Set the initial value of n, iter$_{max}$, Pa, and current iteration count t
2:	Calculate the fitness value of each solution and retain the optimal solution.
3:	Set $t := 0$
4:	for each $i \in [1,n]$ do
5:	Generate initial population z_i^t;
6:	Evaluate the fitness function $f(z_i^t)$
7:	end for
8:	repeat
9:	Execute Eq. 8 to calculate b, generate a random number r, and select 3 positions in the nest randomly.
10:	if $r \leq 0.5$ then
11:	Execute the Eqs. 10–13 to generate a new z_i^{t+1}
12:	else
13:	Execute Eq. 9 to generate a new z_i^{t+1}
14:	end
15:	Evaluate the fitness function and keep the best solution.
16:	if ($f(z_i^{t+1}) > f(z_i^t)$) then
17:	Replace the solution z_i^t with the solution z_i^{t+1} the subscript j is used only to distinguish between the results of previous and subsequent steps};
18:	else if (random number $r_1 > Pa$)
19:	Build new nest with Eq.13
20:	Keep the best solution;
21:	Set $t = t + 1$.
22:	until ($t <$ iter$_{max}$).
23:	Output final solution.

2.3.3 ICS performance analysis

To comprehensively evaluate the optimization efficiency of the ICS algorithm, we selected several standard functions (Table 2) for benchmarking and comparative analysis. Among them, the Ackley, Rastrigin, and Griewank functions, which are characterized by multiple peaks and local minima, are used to evaluate the algorithm's ability to escape local extremes and prevent bogging down in suboptimal solutions, whereas the Schwefel's 2.2 function, the sphere function, and the sum of squares function are single-peak functions and primarily test the algorithm's optimisation accuracy and convergence speed.

Experiments were conducted on a Windows 10 operating system with an Intel® Xeon® Silver 4108 CPU @ 1.80GHz, and simulations were performed in the MATLAB 2022b environment. In terms of parameter settings, n = 30, iter_{max} was set to 500, and the probability parameter Pa was set to 0.25. Each algorithm was run independently 20 times to ensure the reliability of the results, and the results of the experiments are expressed as the mean and standard deviation of the 20 runs; the detailed data are shown in Table 3. The convergence curves of the algorithms are shown in Fig. 3.

By analyzing the data in Table 2 and Fig. 3, we found that the ICS algorithm significantly outperforms the traditional CS algorithm in terms of convergence speed, stability, and computational accuracy. In addition, the ICS algorithm exhibits a stronger global optimization search ability and can effectively avoid falling into local optimal solutions, which indicates that the ICS algorithm has potential advantages in solving complex optimization problems.

Table 2
Benchmark test functions.

Function	Expression	Domain	Best
Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=0}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sqrt{\sum_{i=0}^{n}\cos(2\pi x_i)}\right) + 20 + e$	[−32,32]	0
Rastrigin	$f(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	0
Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) + 1]$	[−600,600]	0
Schwefel's 2.2	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	[−10,10]	0
Sphere	$f(x) = \sum_{i=1}^{n}x_i^2$	[−100,100]	0
Sum square	$f(x) = \sum_{i=1}^{n}ix_i^2$	[−10,10]	0

Table 3
Comparison results between ICS and CS algorithms (mean ± standard deviation).

Function	ICS algorithm	CS algorithm
Ackley	$8.8818 \times 10^{-16} \pm 0$	0.6684 ± 0.6102
Rastrigin	0 ± 0	12.3984 ± 3.0452
Griewank	0 ± 0	0.1071 ± 0
Schwefel's 2.2	$6.0115 \times 10^{-75} \pm 0$	0.9196 ± 0.1215
Sphere	0 ± 0	8.6927 ± 4.0793
Sum square	$1.12 \times 10^{-142} \pm 0$	0.7917 ± 0.4251

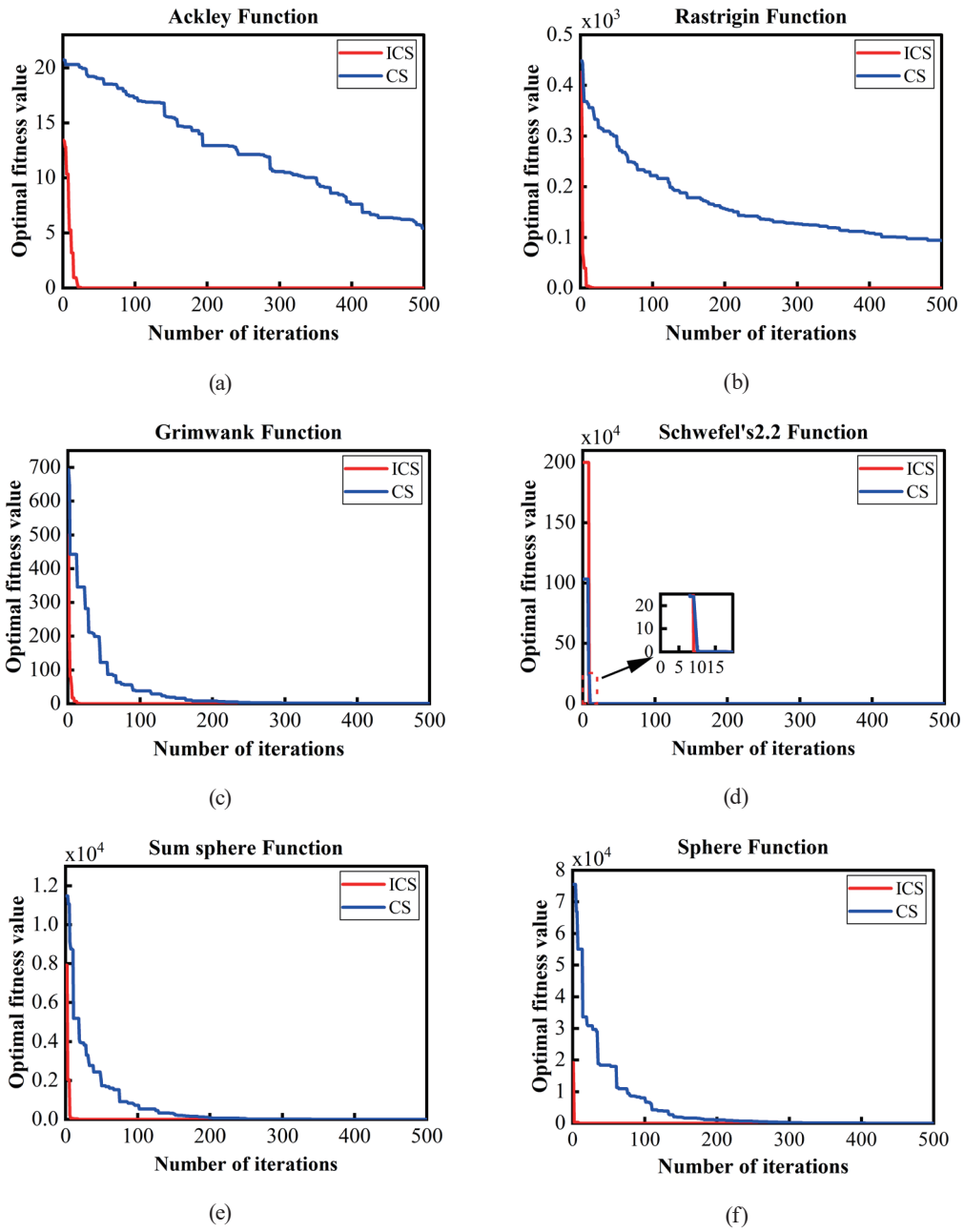

Fig. 3. (Color online) Convergence curve of the function optimization function.

2.3.4 ICS-SVM

First, the ICS algorithm is used to optimize the penalty and kernel function parameters of the SVM to obtain the optimal parameters (Fig. 4). Second, sEMG features are used to train the SVM with optimal parameters. Finally, the trained model is used to predict and recognize different gait phases.

The specific steps can be expressed as follows:

(1) The sEMG signal data from the lower limbs are collected and preprocessed, and time-domain features are extracted and then divided into the training and test datasets.
(2) The settings of C, σ, n, iter_{max}, and Pa are initialized.
(3) Training and error calculation are carried out using the training dataset to find and maintain the optimal nest position.
(4) When the random value r is evaluated against 0.5 and found to be less than or equal to this threshold, the nest position is updated using Eq. (9), followed by an error comparison with the previous nest to ensure that the most favorable nest location and its associated parameters are maintained. Conversely, if r is greater than 0.5, the nest position is updated using Eq. (13).
(5) If the random number $r_1 > Pa$, then we discard the worse nest and update the nest position using Eq. (14). Otherwise, we keep the current nest position unchanged, compute the optimal position of the nest, and compare the error with that of the previous nest to maintain the best possible position and its relevant parameters.

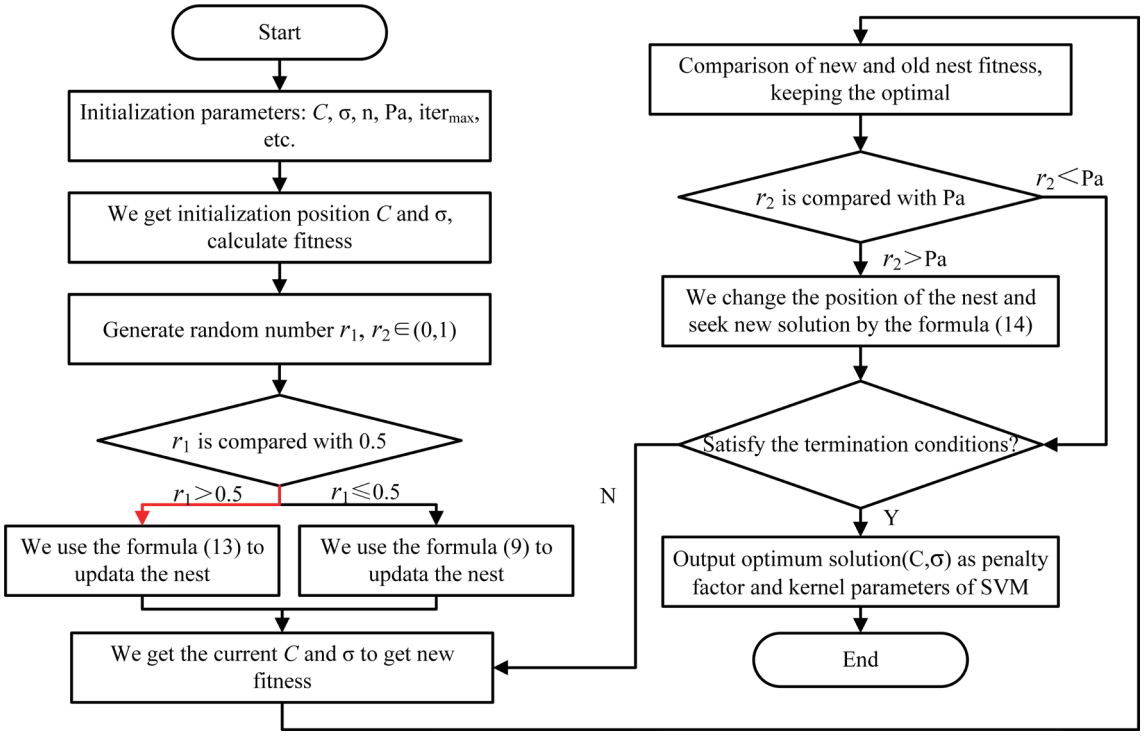

Fig. 4. (Color online) Flowchart of the ICS-SVM algorithm.

(6) The error of the optimal nest position is compared with the preset accuracy standard. If the error is within an acceptable accuracy range, the parameters C and σ are output. Otherwise, return to (5) and continue iteration.
(7) The SVM model is trained using the training dataset to obtain the prediction model.

3. Experimental Protocol

3.1 Experimental subjects

Ten male volunteers (age: 24 ± 3 years old, height: 173 ± 6 cm, weight: 67.5 ± 8.59 kg, and lower limb height: 900.30 ± 34.96 mm) were recruited from Tianjin University of Science and Technology for this study. All the volunteers had no history of joint or neurological disease or injury. All participants signed a written informed consent form in accordance with the policies of Tianjin University of Science and Technology before participating in the formal experiment.

3.2 Experimental setups and muscle selection

In this study, the raw sEMG signals were recorded using twelve electrodes (Noraxon, Scottsdale, USA). These sEMG signals were recorded at a sampling frequency of 1500 Hz. The Vicon motion capture system (Vicon Motion Systems Ltd., Oxford, UK), operating at a sampling frequency of 100 Hz, was used to capture foot trajectories. In addition, the experimental setup consisted of several electrodes, reflective markers, a treadmill, and a computer.

Human lower extremity muscle groups are critical for activities such as walking, running, and jumping, including rectus femoris (RF), biceps femoris (BF), semitendinosus, TA, gastrocnemius, and gastrocnemius medialis (GM).[48] In this study, seven key muscles were selected as targets for sEMG acquisition on the basis of their anatomical locations and biomechanical roles (Fig. 5). These target muscles included RF, vastus lateralis (VL), BF, TA, GM, gracilis, and right external oblique (REO). Their specific functions are as follows: RF and BF are key muscles primarily involved in the movements of knee and hip joints. VL plays a key role in maintaining knee stabilization and promoting its extension. TA is responsible for controlling ankle dorsiflexion and foot valgus. GM plays an important role in knee and ankle dorsiflexion. Gracilis muscles are mainly involved in hip adduction and internal rotation. REO is associated with lateral trunk flexion and rotation.

3.3 Experimental protocol

According to the requirements and purpose of the experiment, we placed the sEMG sensors and reflective markers on each subject according to the anatomical location. Ten subjects were asked to walk on a treadmill at three different speeds (0.5, 1.35, and 1.8 m/s). Each speed lasted 20 s and was repeated five times. To minimize possible fatigue-related errors in EMG signal acquisition, subjects were allowed to rest for 10 min before each experiment. Gait phase recognition in this study consisted of three main steps: signal collection, feature extraction, and

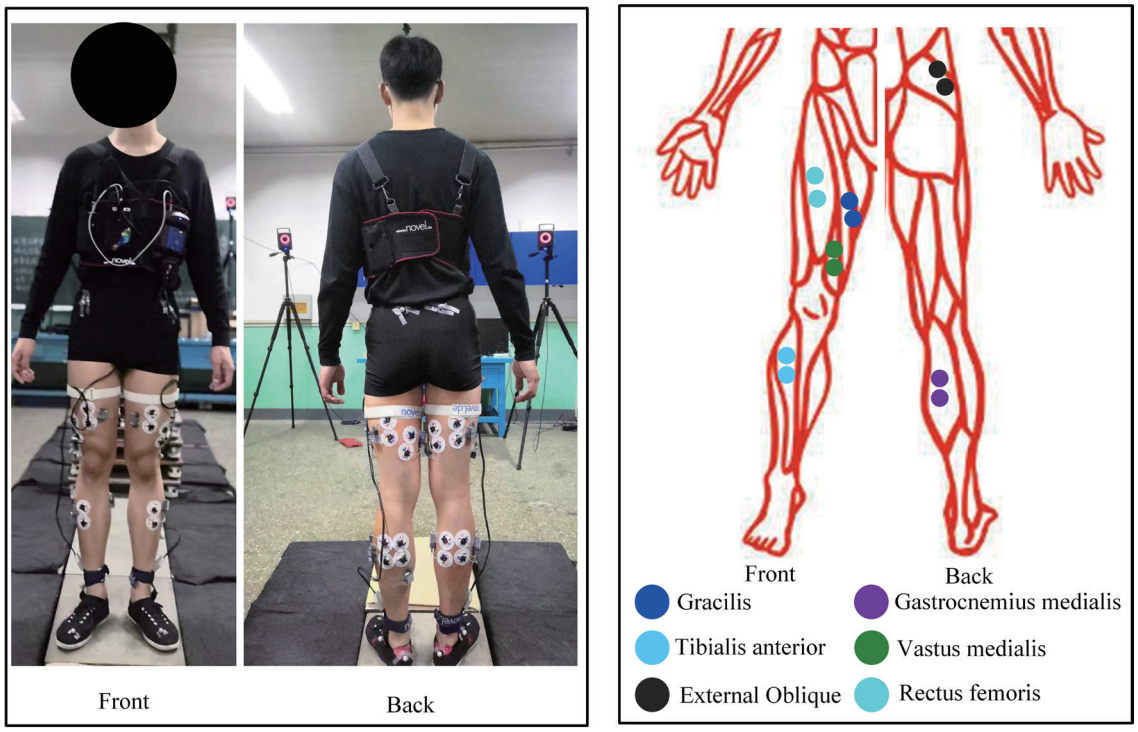

Fig. 5. (Color online) Layout diagram of the collection equipment.

gait recognition. First, sEMG and foot mark trajectory data were collected during the gait experiment. Then, four time-domain features were extracted from the preprocessed sEMG signals, i.e., *MAV*, *WL*, *VAR*, and *RMS*. Finally, these features were introduced into the ICS-SVM classifier for gait recognition. To increase the reliability of the experimental results, each model was independently trained three times. The entire process is shown in detail in Fig. 6.

4. Results

4.1 Recognition results of ICS-SVM model

In this study, 200 datasets were collected from each subject. These datasets were integrated into a total dataset of 10 × 200, which was segmented using the *k*-fold cross-validation method ($k = 10$) to improve the robustness of the model. At each iteration, the data sample of one subject was selected as the test dataset for the recognition model, and the data samples of the other nine subjects were used as the training dataset. The cross-validation was repeated 10 times, each subject was validated once, and the results of the average of 10 iterations were a single estimate. The kernel function of the SVM model was set to the Gaussian kernel function, the value range of the parameters σ and C was set to $[2^{-10},2^{10}]$, the number of iterations was set to 500, and the number of populations was set to 25. The ICS algorithm was used to optimize the σ and C of the SVM model. The ICS algorithm was set to 25 nests in this optimization, with a discovery

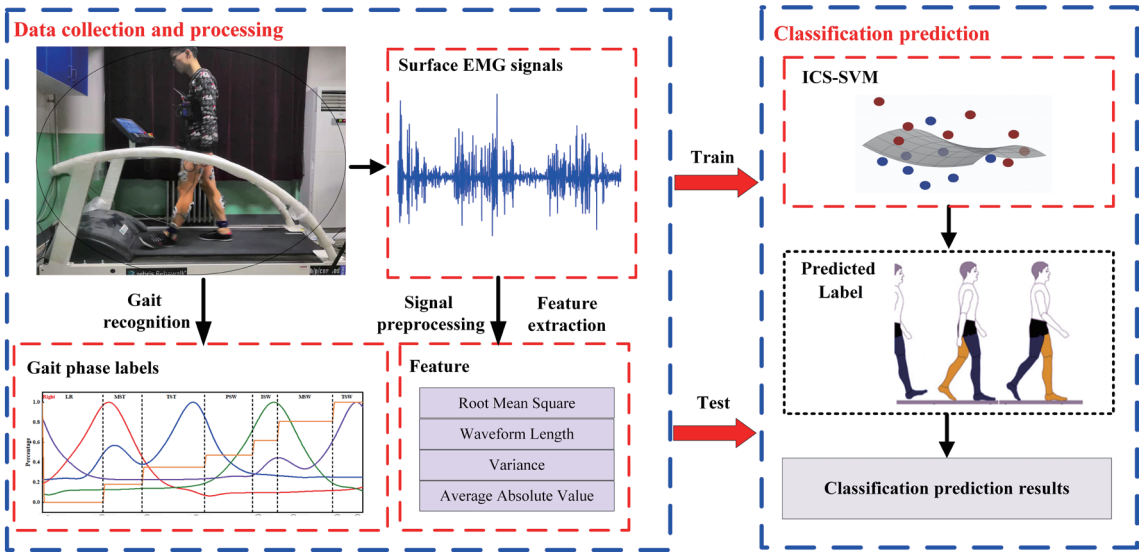

Fig. 6. (Color online) Overall flowchart of the experiment.

particular, the LR, MST, MSW, and TSW phases were recognized with the highest accuracy, with an average recognition accuracy of more than 95%. In contrast, the TST and PSW phases had lower recognition accuracies, whereas the ISW phase had the lowest recognition accuracy, and this phase was the most prone to misclassification. In addition, the model had the highest overall average recognition accuracy of 95.29% when the motion speed was 1.35 m/s.

4.2 Comparing model performance

Table 4 and Fig. 8 show the results of the comparative analysis of the ICS-SVM model with the SVM model and CS-based SVM (CS-SVM) models for the recognition of seven gait phases. We can see that when the SVM model is used directly for gait phase recognition, the accuracy is 92.177%. After the optimization of both the CS and ICS algorithms, the average accuracy rate became more than 94%, which indicates that SVM parameter optimization can significantly increase the accuracy of gait phase detection. Overall, the recognition performance of the ICS-SVM model in different gait phases is significantly higher than those of the SVM and CS-SVM models, and the average recognition accuracy is improved by 3.2 and 1.01%, respectively. In terms of gait phase recognition time, the ICS-SVM model has an average recognition time of only 15.6 ms, significantly shorter than those of both the SVM and CS-SVM models. According to the standard deviation data of recognition accuracy for each gait phase in Fig. 8, the ICS-SVM model shows a greater stability and precise adaptability to changes in gait phases. The experimental results show that the ICS-SVM model outperforms both the SVM and CS-SVM models in terms of both recognition accuracy and time. This makes it highly effective for gait analysis applications within lower limb rehabilitation exoskeletons.

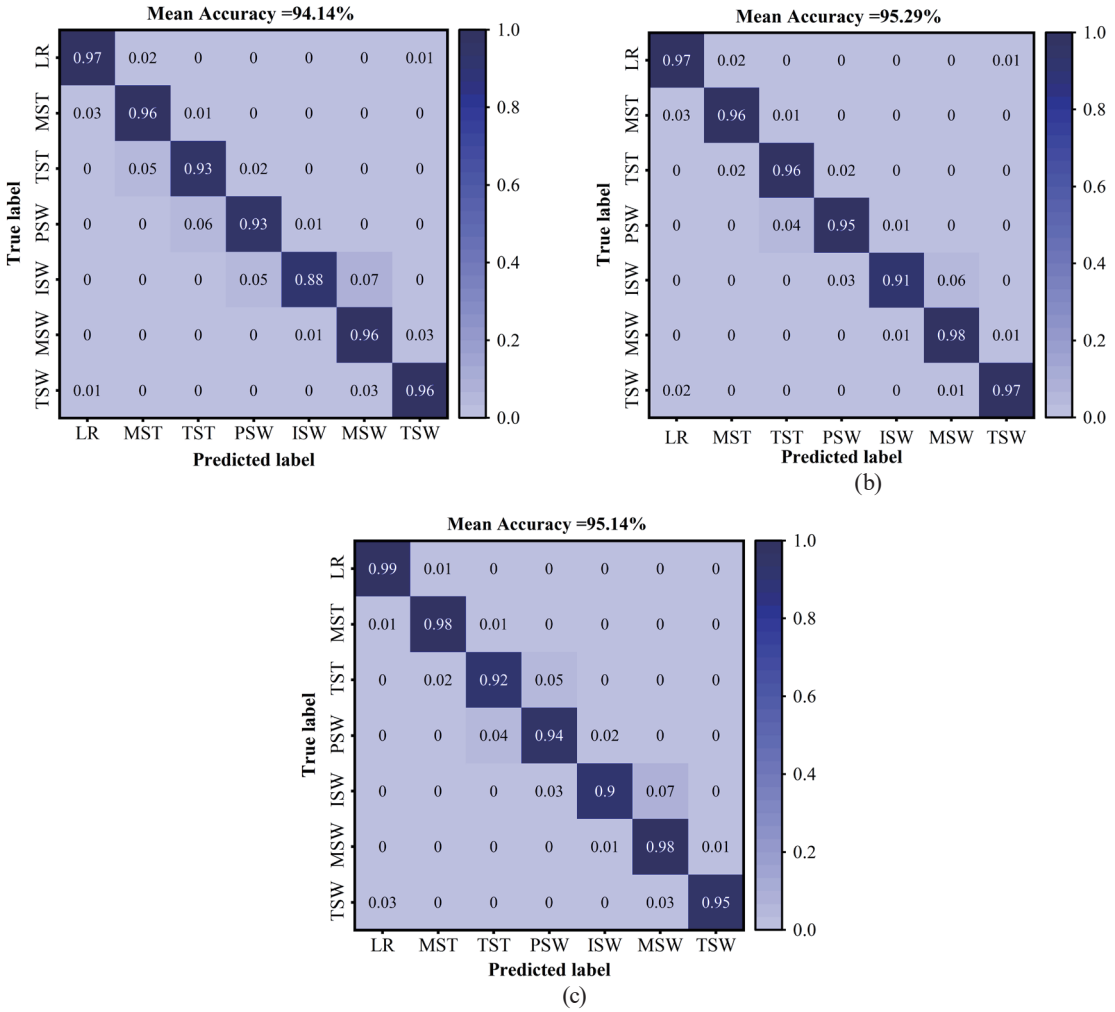

Fig. 7. (Color online) Recognition of gait phase confusion matrix of the ICS-SVM model: (a) 0.5, (b) 1.35, and (c) 1.8 m/s.

Table 4
Results of the performance comparison of the models.

Model	Recognition Accuracy (%)			Average Accuracy (%)	Recognition time (ms)
	0.5 m/s	1.35 m/s	1.8 m/s		
SVM	90.03	93.14	93.36	92.177	64.2
CS-SVM	92.66	95.02	94.83	94.170	36.7
ICS-SVM	94.39	95.53	95.45	95.125	15.6

probability of 0.25, and after a maximum of 500 iterations, the kernel function parameter was obtained as 2.475, and the penalty factor was obtained as 0.034. We then used the ICS-SVM model to perform gait phases at different walking speeds for recognition. The corresponding confusion matrix is shown in Fig. 7. We can observe that the ICS-SVM classifier successfully recognizes seven gait phases with an average recognition accuracy of more than 94%. In

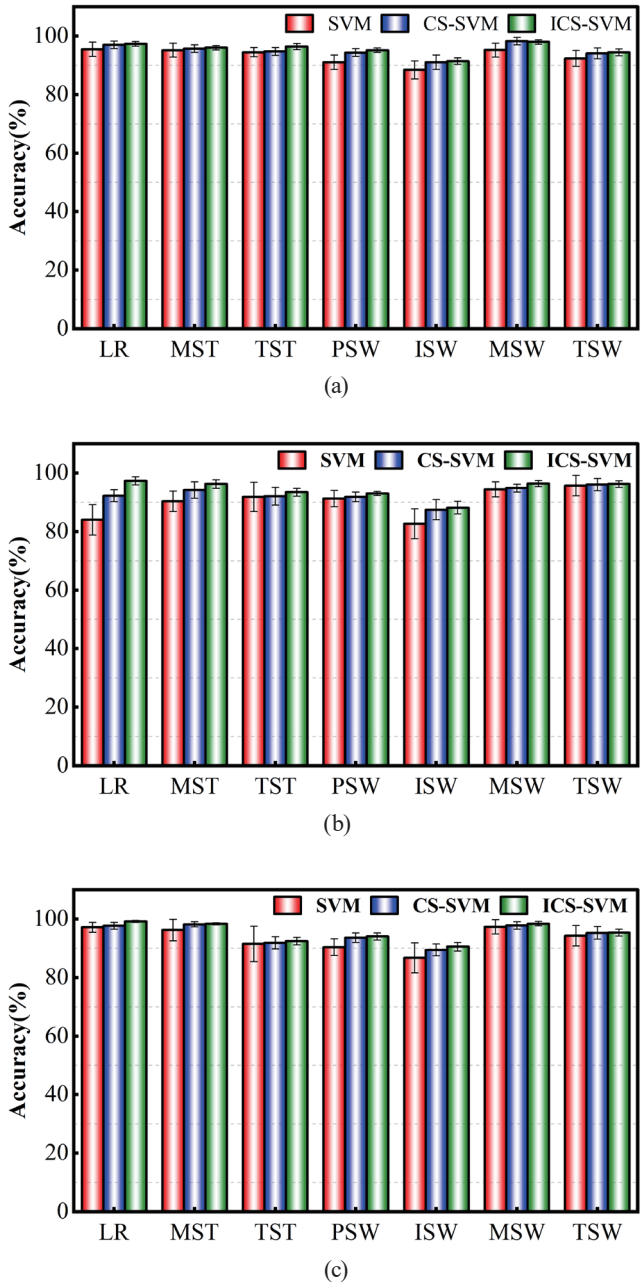

Fig. 8. (Color online) Recognition of gait phases (mean ± standard deviation): (a) 0.5, (b) 1.35, and (c) 1.8 m/s.

5. Discussion

In this study, we explored the use of the ICS-SVM model to recognize the gait phase at three walking speeds and obtained that the overall average recognition accuracy can reach 95.125%. The average recognition accuracies obtained were 94.39, 95.53, and 95.45% for walking speeds of 0.5, 1.35, and 1.8 m/s, respectively, and the highest recognition accuracies were obtained for the gait phases of LR, MST, MSW, and TSW (Fig. 8). To test the recognition performance of the

ICS-SVM model for seven gait phases, we compared it not only with the SVM and CS-SVM models, but also with the existing sEMG-based gait recognition models.

Zhang *et al.*[49] used a convolutional neural network model to detect and recognize five different gait phases, with an average recognition accuracy of 93.40%. Di Nardo *et al.*[50] completed the classification and recognition of gait phases using the MLP model with the feature values extracted from the sEMG signals with an accuracy of 93.4 ± 2.3%. Gao *et al.*[51] distinguished four different gait phases by using extreme learning machine (ELM) and LDA classification algorithms, and the average recognition accuracies of the gait phases were 87.60 and 92.105%, respectively. When using the separate time domain features of sEMG, the average recognition accuracies of the gait phases were 87.60 and 92.105%, respectively. Yun *et al.*[52] used CS-SVM to recognize seven different gait phases, and the average accuracy of gait phase recognition at non-voting was 94.05%. In addition, Yun *et al.* validated generalized regression neural networks, product-based neural networks, and ELM classification models for gait phase recognition and found that the recognition accuracies were all low (less than 75%). Zhang[53] introduced an SVM model for human posture recognition with an average accuracy of about 90.80%.

In contrast to the previous studies, the ICS-SVM model we present can capture more gait information. Moreover, the predictive results of this model are consistent with the pattern observed in the progression of gait phases. This model has a higher recognition accuracy than the existing traditional recognition classification models. Furthermore, an examination of Table 3 and Fig. 8 reveals that variations in walking speed significantly affect the accuracy of gait phase identification. Notably, the accuracy of gait phase recognition reaches its peak when the walking speed is set to 1.35 m/s. From the recognition results of each gait phase, we can see that the ISW phase has the lowest recognition accuracy and that the recognition error mainly occurs in the transition between the temporal gait phases.

To investigate the effectiveness of ICS for SVM optimization, we compared the differences between the ICS-SVM and CS-SVM algorithms (Table 3). We found that the accuracy of SVM models optimized with the ICS algorithm was generally higher than those of the CS-SVM models, with the average recognition accuracy increasing from 94.170 to 95.125%. This also indicates that the ICS-SVM model has some advantages in recognizing the gait phase of the lower limbs.

After analyzing the gait phase detection data, it was determined that there was no potential for further improvement in classification accuracy. This phenomenon may be due to the synchronization problem between the sEMG signal and the data acquisition of the Vicon motion capture system. Since the acquisition frequency of the Vicon motion capture system is 100 Hz and the sEMG signal is 1500 Hz, this resulted in a mismatch between each frame of optical data and up to 15 frames of EMG data. This mismatch inevitably affects the accurate identification of the gait phase when using the Vicon data to verify the sEMG output. In addition, by analyzing the confusion matrix of the ICS-SVM model, the recognition error was most significant in the ISW phase. Specifically, there is a miscalculation of the toe-off event when converting the PSW gait to the ISW gait. Inaccuracies were observed in the identification of the toe-off event during the transition from the PSW to the ISW phase. Second, a significant challenge was the transition

from the ISW to the MSW phase, where the complexity lies in determining the perpendicularity of the shank to the ground. However, there are greater challenges in accurately discriminating this transition event on the basis of EMG signals. Therefore, these two transition stages are the main reasons for the reduced accuracy of ISW gait recognition.

Despite the excellent performance of the ICS-SVM model in the gait recognition task, this study has several limitations. The motion capture technology is recognized as an effective tool for capturing accurate 3D motion data, but its accuracy is still affected by marker position, camera calibration, and soft tissue motion artifacts between the marker and the skin. Therefore, the focus of future research should be extended to testing more relevant field environments and further validating the accuracy of motion capture data. Finally, although the ICS-SVM algorithm shows potential in motion intent recognition experiments, further validation is needed to determine its superiority over other optimization methods. In addition, we explored a more precise synchronization of data acquired by the Vicon motion capture system with sEMG signals to reduce the negative effects of their asynchrony.

6. Conclusion

In this study, our objective was to improve the accuracy and efficiency of gait phase recognition. To achieve this, a classification recognition model based on ICS-SVM was proposed to complete the recognition of seven gait phases. Ten subjects performed a walking task on a treadmill at speeds of 0.5, 1.35, and 1.8 m/s. The results showed that the ICS-SVM model outperforms the traditional recognition algorithms in terms of accuracy, especially at a walking speed of 1.35 m/s, where it achieves the highest accuracy for gait phase recognition, reaching 95.53%. This result demonstrates the performance of the optimized SVM recognition algorithm. The gait phase recognition method based on the ICS-SVM model can be applied to the fields of intelligent medical treatment and lower limb rehabilitation training.

Acknowledgments

This research was partly supported by the Tianjin Graduate Research and Innovation Project under Grant No. 2021YJSB205, Tianjin Sci-Tech Projects under Grant Nos. 23YFYSHZ00280 and 2022ZD032, and Tianjin Municipal Education Commission's research project under Grant No. 2022ZD026.

References

1 J. H. J. Allum and M. G. Carpenter: Curr. Opin. Neurol. **18** (2005) 15. https://doi.org/10.1097/00019052-200502000-00005
2 C. Mei, F. Gao, and Y. Li: Sensors **19** (2019) 5499. https://doi.org/10.3390/s19245499
3 J. Ziegler, H. Gattringer, and A. Mueller: 2018 7th IEEE Int. Conf. Biomedical Robotics and Biomechatronics (Biorob) (2018) 978. https://doi.org/10.1109/biorob.2018.8487750
4 D. Shi, W. Zhang, W. Zhang, and X. Ding: Chin. J. Mech. Eng. **32** (2019). https://doi.org/10.1186/s10033-019-0389-8
5 I. Komnik, S. Weiss, C. H. Fantini Pagani, and W. Potthast: Gait Posture **41** (2015) 370. https://doi.org/10.1016/j.gaitpost.2015.01.019

6 K. Tong and M. H. Granat: Med. Eng. Phys. **21** (1999) 87. https://doi.org/10.1016/s1350-4533(99)00030-2
7 M. Boutaayamou, C. Schwartz, J. Stamatakis, V. Denoël, D. Maquet, B. Forthomme, J.-L. Croisier, B. Macq, J. G. Verly, G. Garraux, and O. Brüls: Med. Eng. Phys. **37** (2015) 226. https://doi.org/10.1016/j.medengphy.2015.01.001
8 O. Dehzangi, M. Taherisadr, and R. ChangalVala: Sensors **17** (2017) 2735. https://doi.org/10.3390/s17122735
9 H. Q. Chao, Y. W. He, J. P. Zhang, and J. F. Feng: AAAI Conf. Artif. Inte. (2019) 8126. https://doi.org/10.1609/aaai.v33i01.33018126
10 M. Grimmer, K. Schmidt, J. E. Duarte, L. Neuner, G. Koginov, and R. Riener: Front. Neurorob. **13** (2019). https://doi.org/10.3389/fnbot.2019.00057
11 W. T. Sheng, W. Guo, F. S. Zha, Z. Y. Jiang, X. Wang, and H. D. Zhang: 2019 IEEE 4th Int. Conf. Advanced Robotics and Mechatronics (Icarm 2019) (2019) 899. https://doi.org/10.1109/icarm.2019.8834082
12 W. Teufl, M. Lorenz, M. Miezal, B. Taetz, M. Fröhlich, and G. Bleser: Sensors **19** (2018) 38. https://doi.org/10.3390/s19010038
13 M. Roerdink, B. Coolen, B. E. Clairbois, C. J. C. Lamoth, and P. J. Beek: J. Biomech. **41** (2008) 2628. https://doi.org/10.1016/j.jbiomech.2008.06.023
14 T. C. Pataky, T. Mu, K. Bosch, D. Rosenbaum, and J. Y. Goulermas: J. R. Soc. Interface **9** (2011) 790. https://doi.org/10.1098/rsif.2011.0430
15 C. J. De Luca, S.-S. Chang, S. H. Roy, J. C. Kline, and S. H. Nawab: J. Neurophysiol. **113** (2015) 1941. https://doi.org/10.1152/jn.00555.2014
16 H. Xu and A. Xiong: IEEE Sens. J. **21** (2021) 13019. https://doi.org/10.1109/jsen.2021.3068521
17 X. Shi, P. Qin, J. Zhu, M. Zhai, and W. Shi: IEEE Access **8** (2020) 132882. https://doi.org/10.1109/access.2020.3008901
18 Y. Liu, S. Guo, Z. Yang, H. Hirata, and T. Tamiya: IEEE J. Biomed. Health. Inf. **25** (2021) 1529. https://doi.org/10.1109/jbhi.2020.3027303
19 G. I. Papagiannis, A. I. Triantafyllou, I. M. Roumpelakis, F. Zampeli, P. Garyfallia Eleni, P. Koulouvaris, E. C. Papadopoulos, P. J. Papagelopoulos, and G. C. Babis: J. Med. Eng. Technol. **43** (2019) 59. https://doi.org/10.1080/03091902.2019.1609610
20 C. Frigo and P. Crenna: Clin. Biomech. **24** (2009) 236. https://doi.org/10.1016/j.clinbiomech.2008.07.012
21 C. Morbidoni, A. Cucchiarelli, S. Fioretti, and F. Di Nardo: Electronics **8** (2019) 894. https://doi.org/10.3390/electronics8080894
22 Y. Chang, L. A. N. Wang, M. I. N. Li, M. Liu, L. Lin, B. O. Cui, and Q. Liu: J. Mech. Med. Biol. **23** (2023). https://doi.org/10.1142/s0219519423400663
23 X. Xi, C. Yang, J. Shi, Z. Luo, and Y.-B. Zhao: Neural Process. Lett. **50** (2019) 2265. https://doi.org/10.1007/s11063-019-10008-w
24 Y. Guo, R. Gravina, X. Gu, G. Fortino, and G. Z. Yang: Proc. 2020 IEEE Int. Conf. Human-Machine Systems (Ichms) (2020) 312. https://doi.org/10.1109/ichms49158.2020.9209449
25 B. Wei, Z. Ding, C. Yi, H. Guo, Z. Wang, J. Zhu, and F. Jiang: Front. Neurorob. **15** (2021). https://doi.org/10.3389/fnbot.2021.704226
26 N. Nazmi, M. A. Abdul Rahman, S.-I. Yamamoto, and S. A. Ahmad: Biomed. Signal Process. Control **47** (2019) 334. https://doi.org/10.1016/j.bspc.2018.08.030
27 M. Meng, Q. She, Y. Gao, and Z. Luo: 2010 IEEE Int. Conf. Information and Automation (2010) 852. https://doi.org/10.1109/icinfa.2010.5512456
28 C. D. Joshi, U. Lahiri, and N. V. Thakor: 2013 IEEE Point-of-Care Healthcare Technologies (PHT) (2013) 228. https://doi.org/10.1109/pht.2013.6461326
29 F. Wang, L. Yan, and J. Xiao: Sens. Mater. **31** (2019) 1335. https://doi.org/10.18494/sam.2019.2288
30 A. Dellacasa Bellingegni, E. Gruppioni, G. Colazzo, A. Davalli, R. Sacchetti, E. Guglielmelli, and L. Zollo: J. NeuroEng. Rehabil. **14** (2017). https://doi.org/10.1186/s12984-017-0290-6
31 E. Zheng and Q. Wang: IEEE Trans. Neural Syst. Rehabil. Eng. **25** (2017) 161. https://doi.org/10.1109/tnsre.2016.2529581
32 J. Zhang, L. Wang, and G. Wang: Ind. Lubr. Tribol. **76** (2024) 345. https://doi.org/10.1108/ilt-10-2023-0328
33 J. Bae and M. Tomizuka: Mechatronics **21** (2011) 961. https://doi.org/10.1016/j.mechatronics.2011.03.003
34 D. N. Condie: Int. J. Rehabil. Res. **15** (1992) 181. https://doi.org/10.1097/00004356-199206000-00017
35 J. Taborri, E. Palermo, S. Rossi, and P. Cappa: Sensors **16** (2016) 66. https://doi.org/10.3390/s16010066
36 J. Camargo, A. Ramanathan, W. Flanagan, and A. Young: J. Biomech. **119** (2021). https://doi.org/10.1016/j.jbiomech.2021.110320
37 A. Vijayvargiya, C. Prakash, R. Kumar, S. Bansal, and J. M. R. S. Tavares: Biomed. Signal Process. Control **66** (2021) 102406. https://doi.org/10.1016/j.bspc.2021.102406

38 O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas: Sensors **14** (2014) 6474. https://doi.org/10.3390/s140406474
39 B. Zhou, H. Wang, F. Hu, N. Feng, H. Xi, Z. Zhang, and H. Tang: Comput. Methods Programs Biomed. **193** (2020). https://doi.org/10.1016/j.cmpb.2020.105486
40 X. Sui, K. Wan, and Y. Zhang: Optik **176** (2019) 228. https://doi.org/10.1016/j.ijleo.2018.09.040
41 A. Phinyomark, P. Phukpattaranont, and C. Limsakul: Expert Syst. Appl. **39** (2012) 7420. https://doi.org/10.1016/j.eswa.2012.01.102
42 X. S. Yang and S. Deb: 2009 World Congr. Nature & Biologically Inspired Computing (NaBIC) (2009) 210. https://doi.org/10.1109/NABIC.2009.5393690
43 X. S. Yang and S. Deb: Int. J. Math. Modell. Numer. Optim. **1** (2010) 330. https://doi.org/10.1504/IJMMNO.2010.035430
44 W. Tan, L. Sun, F. Yang, W. Che, D. Ye, D. Zhang, and B. Zou: Optik **154** (2018) 581. https://doi.org/10.1016/j.ijleo.2017.10.090
45 U. Mlakar, I. Fister, and I. Fister: Swarm Evol. Comput. **29** (2016) 47. https://doi.org/10.1016/j.swevo.2016.03.001
46 X. Meng, J. Chang, X. Wang, and Y. Wang: Energy **168** (2019) 425. https://doi.org/10.1016/j.energy.2018.11.096
47 N. J. Cheung and H.-B. Shen: J. Mol. Graphics Modell. **54** (2014) 114. https://doi.org/10.1016/j.jmgm.2014.10.002
48 S. Valentin and R. R. Zsoldos: J. Electromyography Kinesiology **28** (2016) 167. https://doi.org/10.1016/j.jelekin.2015.12.005
49 X. Zhang, Y. Hu, R. Luo, C. Li, and Z. Tang: Sensors **21** (2021) 8365. https://doi.org/10.3390/s21248365
50 F. Di Nardo, C. Morbidoni, A. Cucchiarelli, and S. Fioretti: Biomed. Signal Process. Control **63** (2021) 102232. https://doi.org/10.1016/j.bspc.2020.102232
51 F. Gao, T. Tian, T. Yao, Q. Zhang, and P. A. Karjalainen: Comput. Intell. Neurosci. **2021** (2021) 1. https://doi.org/10.1155/2021/6693206
52 X. Yun, X. Ling, G. Lei, L. Zhanhao, and S. Bohan: Int. J. Adv. Comput. Sci. Appl. **14** (2023) 120. https://doi.org/10.14569/ijacsa.2023.0140113
53 L. Zhang: Int. J. Bioautomation **22** (2018) 179. https://doi.org/10.7546/ijba.2018.22.2.179-186