Sensors and Materials, Vol. 34, No. 3 (2022) 1017–1031
MYU Tokyo

S & M 2870

Human Motion Mode Recognition Based on Multi-parameter Fusion of Wearable Inertial Module Unit and Flexible Pressure Sensor

Tao Zhen, Hao Zheng, and Lei Yan*

Department of Mechanical Engineering, Beijing Forestry University,
35 Qinghua East Road, Haidian District, Beijing 100083, China

(Received November 30, 2021; accepted January 21, 2022)

Keywords: dynamic threshold, dynamic block matching, flexible pressure sensor, human motion modes, inertial module units

Aiming at the rapid recognition of human motion modes required in the intelligent control algorithm of exoskeleton robots, in this paper, on the basis of the characteristics of inertial data and pressure data collected by smart terminals carried by pedestrians, a dynamic block matching algorithm based on kinematics (DBMK) using motion mode recognition is proposed. This algorithm involves signal extraction and motion feature matching discrimination. More specifically, it first uses the method of periodic signal capture in adaptive motion mode to capture the heel touch event from the signal collected by a flexible pressure sensor mounted on the heel, and extracts the corresponding periodic signal. Finally, the DBMK algorithm uses a self-made lower limb motion information acquisition system to obtain human motion angle data. After kinematics preprocessing, the distance correlation coefficient based on Pearson weight proposed in this paper is used to identify the current human motion model category. The DBMK algorithm was used to identify five common human motion modes from the output data of inertial module units and flexible pressure sensors, and experimental results show that the proposed DBMK algorithm has an accuracy of 90.86% for the recognition of the five common motion modes.

1. Introduction

Exoskeleton robots for assistance of lower limbs can increase the convenience of human daily life. Exoskeletons of lower limbs have extremely important applications in the industrial,[1,2] military,[3,4] and medical[5,6] fields. The core problem of the exoskeleton robots used for assistance is predicting the position of the exoskeleton joint by recognizing the wearer's intended motion. The fusion of related sensor information, such as from inertial sensors, pressure sensors, and encoders, can predict the intended motion with high accuracy and has been widely used. With the large-scale popularity of smart terminals, sensors are used to improve the estimation of lower limb motion modes based on multi-sensing device fusion. An exoskeleton hybrid control

*Corresponding author: e-mail: mark_yanlei@bjfu.edu.cn
https://doi.org/10.18494/SAM3755

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

assistance system has four main parts: gait detection, motion mode estimation, motion intention recognition, and control strategy. The accurate estimation of motion modes is a prerequisite for the human–computer interactive control of exoskeletons.

Through the use of a wearable intelligent terminal to determine the current motion mode of the wearer and select the correct model parameters, the accuracy of exoskeleton joint movement can be improved. Therefore, motion mode recognition algorithms based on mobile smart terminal equipment have attracted research attention.[7–9] Ogata *et al.* proposed an effective technology for human motion recognition based on motion history images and feature space technology, and conducted experiments on the recognition of six human motion modes, which achieved satisfactory results.[10] However, the method of using images to detect motion modes is easily restricted by the test site, is expensive, and is not suitable for use in exoskeleton control strategies. Wang *et al.* proposed a method based on multiple sources and a general recurrent neural network (GRNN) to identify lower limb motion modes, using principal component analysis to extract features from plantar pressure information and surface electromyography (EMG), then the GRNN was used to recognize the actions of ascending stairs, descending stairs, and walking on flat ground, and the recognition rate reached 89.7%.[11] Song *et al.* studied the impact of surface electromyography (sEMG) signals on the accuracy of human motion pattern recognition, and used their designed multilayer perceptron and long short-term memory (LSTM) neural network to identify seven common human motion modes in daily life.[12] The use of EMG can overcome the space limitation problem caused by image technology. However, the installation of commercial EMG equipment and the attachment of the equipment are complicated, and the user is susceptible to sweat.[13] Moreover, the collected signal contains noise, which will affect the application of EMG equipment in exoskeleton robots. Khan *et al.* used a single three-axis accelerometer sensor to collect data as the input of an artificial neural network, and enhanced the features detected from the activity signal, achieving a recognition rate of above 99% for the four activities of lying, standing, walking, and running.[14] Song *et al.* used a multilayer BP neural network to identify 15 common motion modes based on the characteristics of multisource human lower limb acceleration and plantar pressure.[15]

Traditional motion mode recognition methods mostly use machine learning or deep learning algorithms,[16,17] which are unable to give a reasonable explanation based on the principle of the model. Also, the deep learning model is too complex and the model parameters are too numerous, increasing the amount of calculation and causing the model to run slowly. Therefore, deep learning algorithms with high complexity entail higher computing requirements for portable mobile devices.[18] The DBMK algorithm first establishes a mathematical model of the human body from the perspective of kinematics, and then uses the distance correlation coefficient based on Pearson's weight (PWD) to dynamically match the motion model in the database. The DBMK algorithm has few parameters and low computing power requirements. It can be processed in real time on an ordinary advanced RISC machine (ARM) processor, avoiding the high computational complexity of traditional deep learning algorithms. Experimental results show that the dynamic matching algorithm based on kinematics can effectively detect the wearer's motion pattern, and it is expected to be applied as a strategy for exoskeleton open hybrid control.

2. Materials and Methods

In the process of pedestrian motion, the angle data collected by an inertial sensor of a smart terminal change periodically. Although the stride length, stride height, and stride frequency produced by each pedestrian in the walking process are different, the inertial data generated by each pedestrian in the same motion mode have certain similarities in the waveform. In addition, the distance between the center of gravity of the two feet and the center of gravity of the human when a heel touch event occurs is different in different motion modes, which lays a solid foundation for the DBMK algorithm to realize motion mode recognition. A flow chart of the DBMK algorithm is shown in Fig. 1.

In the periodic signal extraction phase, the algorithm first obtains inertial data and foot pressure data in different motion modes through the inertial module unit (IMU). Then, a single-cycle segmented operation is performed after preprocessing the obtained foot pressure data and inertial data using the heel-to-ground event to generate sample data of the motion mode.

In the motion mode recognition stage, the inertial data collected by the IMU are preprocessed and single-period segmented, and then recognized by the human motion mode classifier generated by the DBMK algorithm. Specifically, the current motion mode is divided into three categories through the human kinematics model, then the sampled data are processed, and finally the sampled data are matched to the template data in the database using the PWD coefficients proposed in this paper, and the current motion mode is finally identified. The specific steps are as follows:

Step 1: First, the collected original inertial data and foot pressure data are preprocessed, and then the heel touch event is detected from the foot pressure data using the dynamic threshold method. The heel touch event detection method is used to continuously collect the motion data of two adjacent steps for single-cycle segmentation.

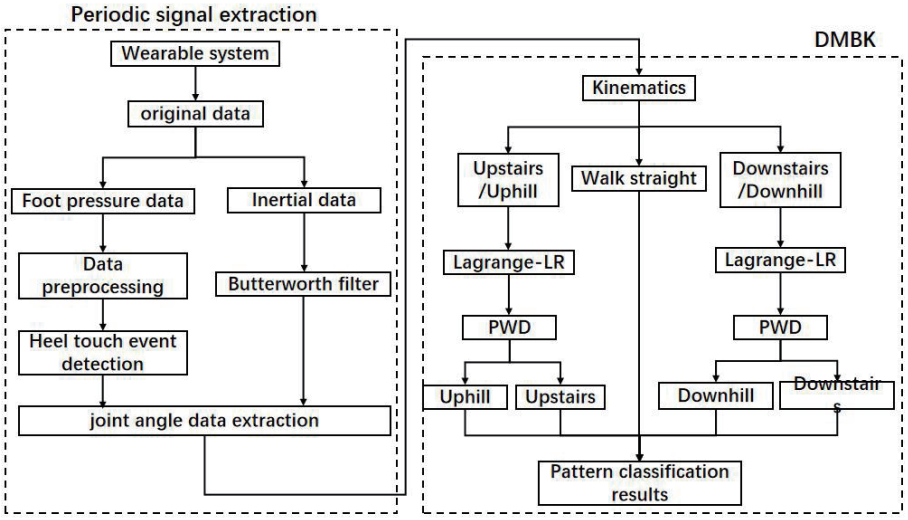
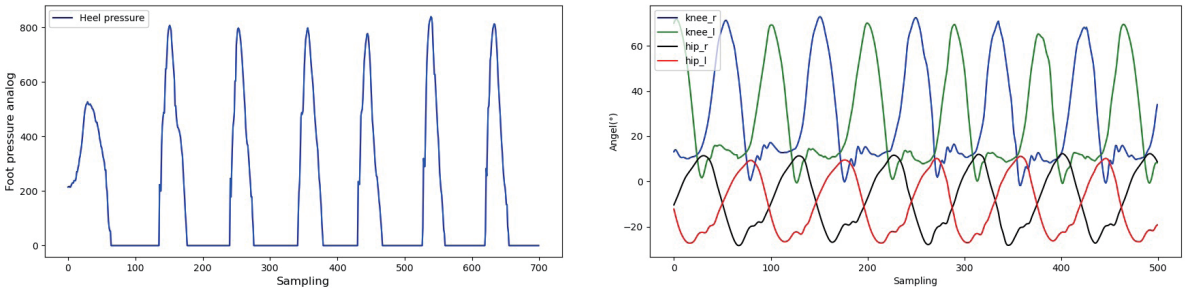


Fig. 1. Flow chart of motion mode recognition algorithm based on DBMK.

Owing to the interference of equipment and the environment, the data collected by inertial sensors and flexible pressure sensors are disturbed by noise with high probability, so a filter is required to preprocess the noise of the original data. Considering the real-time needs of the system and the characteristics of signal noise, the foot pressure data are processed as follows. When the amplitude of the foot pressure data is less than 40, the amplitude is set to zero to avoid misjudgment of heel contact events caused by noise signals; in this study, we use a second-order Butterworth filter to filter the inertial angle data.

As shown in Fig. 2, there are continuous gait cycles with foot pressure data and angle data. The foot pressure signal exhibits the characteristics of a pulse signal, so it is feasible to detect the rising edge signal of two adjacent steps and perform single-cycle segmentation in each motion mode.

The heel-to-ground phase detection method used in this paper is based on the dynamic threshold detection method. The processing flow of this method is shown in Table 1. The effective rising edges of the foot pressure signal must be satisfied: the current point is an upward trend, and the average value and the number of zero points in the sliding window S are less than



(a) (b)

Fig. 2. (Color online) Filtered angle data and foot pressure data. (a) Foot pressure data and (b) angle data.

Table 1
Processing flow of the heel-to-ground phase detection method.

Initialize $\alpha = 0.6$ *and* $\beta = 10$
For T iterations do:
$P(k) = \{ST(k-m), ST(k-m+1), ST(k-m+2), \ldots, ST(k-1), ST(k)\}$
$S(k) = \{ST(k-n), ST(k-n+1), ST(k-n+2), \ldots, ST(k-1), ST(k)\}$
$O(k) = \{ST(k-v), ST(k-v+1), ST(k-v+2), \ldots, ST(k-1), ST(k)\}$
$M = Min(S(k)) + \alpha \times [Max(S(k)) - Min(S(k))] + \beta$
$Av = Sum(S(k)) / n$
$Max\_p = Max(P(k)) / 10$
If $(ST(k) > M$ *and* $ST(k-1) < M$ *and* $Av < Max\_p$ *and* $N_0(S(k))>0.5*n)$:
If $(v > 40)$*:*
Output($O(k)$)
$O(k)$.clear()
k++
end

Here, α is the proportionality coefficient and β is the bias term. n is the length of sliding window S. m is the length of sliding window P. $N_0(S(k))$ represents the number of zeros in sliding window S. v represents the number of sampling points between two heel touch events.

1/10 of the peak value in the sliding window P and more than half of the total number of sampling points in the sliding window S, respectively, and the sampling point interval between two consecutive rising edges must exceed 40. Note that to ensure that the maximum value over a week is obtained, m must be greater than or equal to the sampling frequency.

Step 2: In this paper, human lower limb kinematic modeling is established from human sagittal plane kinematics, and a schematic diagram of the joint angle definition is shown in Fig. 3. When the end point of the thigh or calf is on the left (right) side of the frontal plane, the angle of the sensor is positive (negative). The angle between the knee joint and the hip joint can be derived from the IMU fixed on the thigh and calf. The principle of measuring the hip and knee angles is shown in Fig. 3, and the angles are expressed by Eqs. (1) and (2). The changes in the height of the lower extremities are mainly the result of the joint action of θ_{hip} and θ_s. When the dynamic threshold algorithm detects the right heel contact event, the height H of the right foot relative to the left foot can be obtained from Eqs. (3)–(9). When H is greater than the set threshold, the current motion mode is judged to be upstairs or uphill motion; when H is less than the set threshold, it is judged to be downstairs or downhill motion; when H is near zero, the current motion mode is judged to be walking straight.

$$\theta_{knee} = \theta_{rt} \text{ or } \theta_{lt} \tag{1}$$

$$\theta_{knee} = -\theta_{hip} + \theta_s \tag{2}$$

$$X_1 = X_0 + L_1 \cdot \sin(\theta_{hip}) \tag{3}$$

$$Y_1 = Y_0 + L_1 \cdot \cos(\theta_{hip}) \tag{4}$$

$$X_2 = X_1 + L_2 \cdot \sin(\theta_s) \tag{5}$$

$$Y_2 = Y_1 + L_2 \cdot \cos(\theta_s) \tag{6}$$

$$Y_{left} = Y_0 + L_1 \cdot \cos(\theta_{lt}) + L_2 \cdot \cos(\theta_{ls}) \tag{7}$$

$$Y_{right} = Y_0 + L_1 \cdot \cos(\theta_{rt}) + L_2 \cdot \cos(\theta_{rs}) \tag{8}$$

$$H = Y_{left} - Y_{right} \tag{9}$$

Here, (X_0,Y_0), (X_1,Y_1), and (X_2,Y_2) are the coordinates of the human hip joint, knee joint, and ankle joint, respectively. L_1 is the length of the thigh and L_2 is the length of the calf. θ_{rt} and θ_{lt} represent the right and left knee angles respectively. Y_{left} and Y_{right} represent the Y_2 coordinates of the left and right legs, respectively. H is the distance between the left-leg Y_2 and the right-leg Y_2. The knee angle θ_{knee}, hip angle θ_{hip}, and θ_s are shown in Fig. 3.

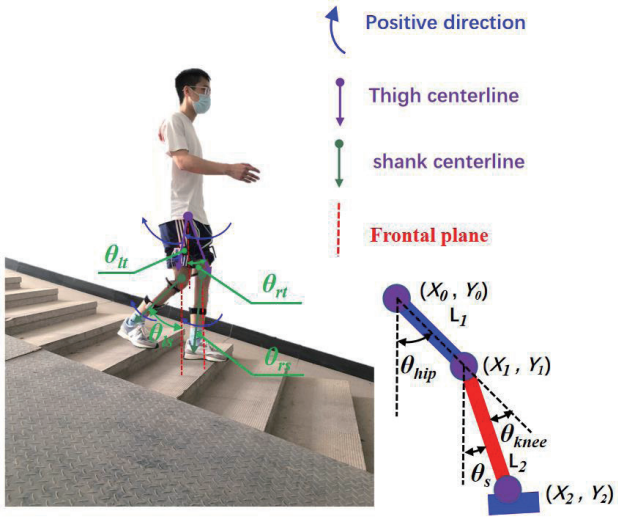


Fig. 3. (Color online) Joint angle definition.

Step 3: Step 1 is repeated multiple times to obtain the sampled knee joint angle sequence and template knee angle sequence T of one gait cycle in different human motion modes. Step 1 can be combined with Step 2 to calculate the height difference H between the centers of gravity of the left and right feet when the heel touch event occurs.

Step 4: After Step 3, Lagrange-LR processing is performed on the sampling sequence $S[i]$. Lagrange-LR is an interpolation operation. When the number of elements in sampling time sequence S is greater than that in template time sequence T in the database, the linear reduction method is used, for example, when T = [1, 3, 5] and S = [0.8, 1.4, 3.2, 3.9, 5.1], then S becomes [0.8, 3.2, 5.1] after being processed by the linear reduction method; when the number of elements in the sampling sequence S is less than the number of elements in the template sequence T in the database, the Lagrangian linear interpolation method is used to expand the sampling sequence, as shown in Eqs. (10)–(12). The performance of processing with the Lagrange-LR algorithm is shown in Fig. 4.

$$l_1(x) = (x - x_2) / (x - x_1) \tag{10}$$

$$l_2(x) = (x - x_1) / (x_2 - x_1) \tag{11}$$

$$L(x) = y_1 \cdot l_1(x) + y_2 \cdot l_2(x) \tag{12}$$

Here, y_1 and y_2 are the knee angle data collection point values at sampling times x_1 and x_2, respectively.

To improve the accuracy of upstairs/uphill and downstairs/downhill motion modes, the PWD correlation coefficient is proposed. Assume that the lengths of time series S and T are m and n, respectively, where m and n are not necessarily equal. The two time series must be aligned by

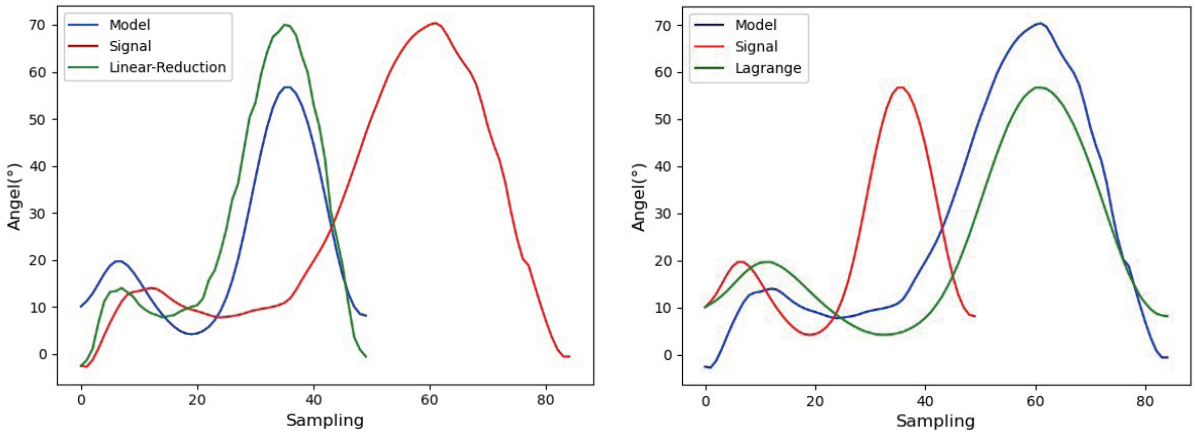


Fig. 4. (Color online) Diagrams showing sampling sequence, template sequence, and interpolation processing. (a) Linear reduction. (b) Second-order Lagrangian interpolation. In the legend, "Signal" represents the sampling sequence, "Model" represents the template sequence in the database, "Lagrange" represents the data sequence obtained by second-order Lagrangian interpolation of the sampling sequence, and "Linear-Reduction" represents the sampling sequence obtained by linear reduction.

Step 4, and the length of the processed time series S is changed to n. In matrix $L(n)$ in Eq. (13), each element represents the distance between $S[i]$ and $T[i]$, where the smaller the distance, the greater the similarity between two points. At the same time, it is necessary to consider the situation that the amplitudes of time series S and T maintain a certain distance, but the amplitude waveforms of the two time series are similar. In other words, $S[i]/T[i]$ is approximately constant over a continuous period of time. The correlation between the two time series will result in a large estimation error if only according to the distance matrix $L(n)$ in this case.

To better represent the correlation between two time series, in this paper, we propose a PWD correlation coefficient that divides matrix $L(n)$ into t parts. Then, the Pearson correlation coefficient is used to obtain the weight w_i of each matrix $L_i(n)$, as shown in Eq. (16), and finally, the distance coefficient of the two time series is obtained using Eq. (17). The motion mode of the sample is determined as that corresponding to the template knee angle sequence with the smallest difference.

$$\begin{aligned} S &= [s_1, s_2, s_3, \cdots, s_n] \\ T &= [t_1, t_2, t_3, \cdots, t_n] \\ L(n) &= [s_1 - t_1, s_2 - t_2, s_3 - t_3, \cdots, s_n - t_n] \end{aligned} \tag{13}$$

$$Sum(|L(n)|) = |s_1 - t_1| + |s_2 - t_2| + \cdots + |s_n - t_n| \tag{14}$$

$$Pear(S,T) = \frac{\sum_{i=1}^{n}(S_i - \overline{S})(T_i - \overline{T})}{\sqrt{\sum_{i=1}^{n}(S_i - \overline{S})^2}\sqrt{\sum_{i=1}^{n}(T_i - \overline{T})^2}} \tag{15}$$

$$w_i = 1 - Pear(S_i, T_i) \tag{16}$$

$$D = \sum_{i=1}^{t} Sum((|L_i(n)|) \cdot w_i) \tag{17}$$

Here, S is the collected sampling time series, T is the template time sequence in the database, and $Pear(S,T)$ is the Pearson correlation coefficient between time series S and T.

Step 5: First, the raw sample is segmented by Step 1 to obtain the sample sequence S to be identified, then the relevant height difference H is calculated by Step 2. S and H are continuously updated after Step 3, and finally, the sample sequence S is matched to the template sequence T in the database after the processing in Step 4 using the proposed PWD correlation coefficient, so as to recognize the motion mode.

3. Results

In this paper, a motion information collection system for the lower limbs is designed as a smart terminal device for users (wearers). This system can collect inertial signals and foot pressure signals of the wearer's lower limbs at the same time. An MD30-60 (China Shenzhen Jinke Electronic Technology Co., Ltd.) flexible pressure sensor with a range of 20 kg is used in this collection system. The IMU in the experiment adopts a JY901 nine-axis inertial sensor with a built-in Kalman filtering algorithm (China Gansu Youxin Electronics Co., Ltd.), where the return frequency is set to 200 Hz and the dynamic angle accuracy is 0.1°. The IMU module is fixed in the sagittal plane of each leg with a magic strap, and the flexible pressure sensor is installed in the wearer's insole.

In the same experimental environment, three wearers (male, height: 173 ± 5 cm, weight: 65 ± 6 kg) performed downstairs, upstairs, downhill, uphill, and straight walking experiments in sequence. The walking experiment was repeated three times in each motion mode. The number of steps in each experiment was 42 steps, the length of the slope was 17 m, and the length of the level road was 27 m. To prevent the participants from continuing to exercise after each experiment, affecting their subsequent motion gait, all participants were required to rest for 5 min after completing each walking experiment to reduce the possible impact of exercise fatigue on the walking gait. In the experiment, the width and height of the steps were 400 and 100 mm, respectively, and the inclination angle of the slope was 15.5°.

To verify the single-cycle segmentation performance of the dynamic threshold method in each motion mode, the subjects were required to perform experiments involving walking on flat ground, experiments ascending and descending ladders, and experiments walking up and down slopes. Figure 5 shows the segmentation results in the downstairs motion mode, which indicates that even if the signal amplitude changes significantly, the dynamic threshold algorithm still performs well in the detection of heel touch events. Then about 100 periodic signals in each motion mode were randomly chosen for analysis, and the experimental results shown in Table 2 were obtained. From these results, we can see that the accuracy of the dynamic threshold

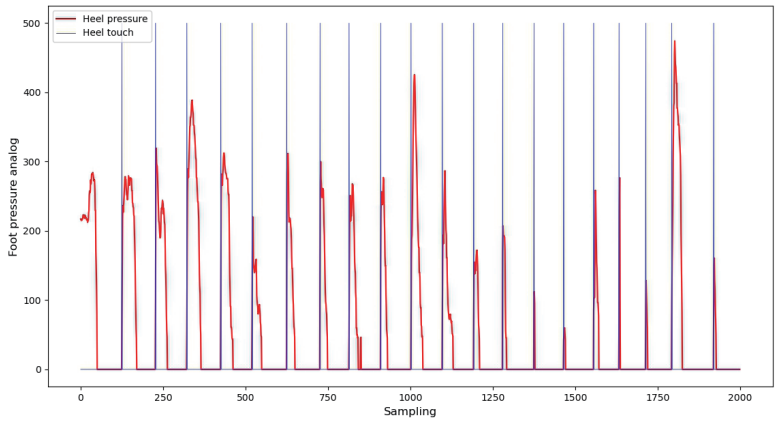


Fig. 5. (Color online) Schematic diagram of single-loop segmentation in downstairs motion mode.

Table 2
Recognition results of dynamic threshold method.

Classification	Walking straight	Upstairs	Uphill	Downstairs	Downhill
Number of true heel touches	117	103	100	107	106
Number of successful recognitions	117	103	100	107	105
Number of misjudgments	0	0	0	0	1
Number of unidentified modes	0	0	0	0	1
Recognition success rate	100%	100%	100%	100%	98.13%

algorithm for heel contact detection in all motion modes is above 98%, and the probability of misjudgment and recognition failure is extremely small.

The dynamic threshold algorithm was used to detect heel touch events with good results. Next, kinematic modeling was used to identify the motion modes of walking straight, upstairs/uphill, and downstairs/downhill. Kinematic modeling was used to build a human body model, and the height of the right foot relative to the left foot when the heel touches the right foot was calculated using Eqs. (1)–(9). From Fig. 6, it can be seen that in the downstairs experiment, 75% of the sampling points were concentrated between −97 and −72 mm, and two abnormal sampling points appeared during the whole performance. In the downhill experiment, except for one abnormal sampling point, all sampling points were concentrated between −58 and −40 mm. In the upstairs experiment, all sampling points were concentrated between 70 and 100 mm, and no abnormal points were found. In the uphill experiment, 75% of the sampling points were above 65 mm and one abnormal point appeared. In the straight walking experiment, 75% of the sampling points were distributed between −10 and 40 mm and one abnormal point appeared. Next, thresholds were determined to identify the motion mode of walking straight, upstairs/uphill, and downstairs/downhill. Through the analysis of Fig. 6, we set the following threshold: when the output result of the kinematic modeling was between −10 and 40, the output result was evaluated as the walking straight mode; when the output result of the kinematic modeling was greater than 40, the output result was evaluated as the upstairs/uphill mode; when the output result of the

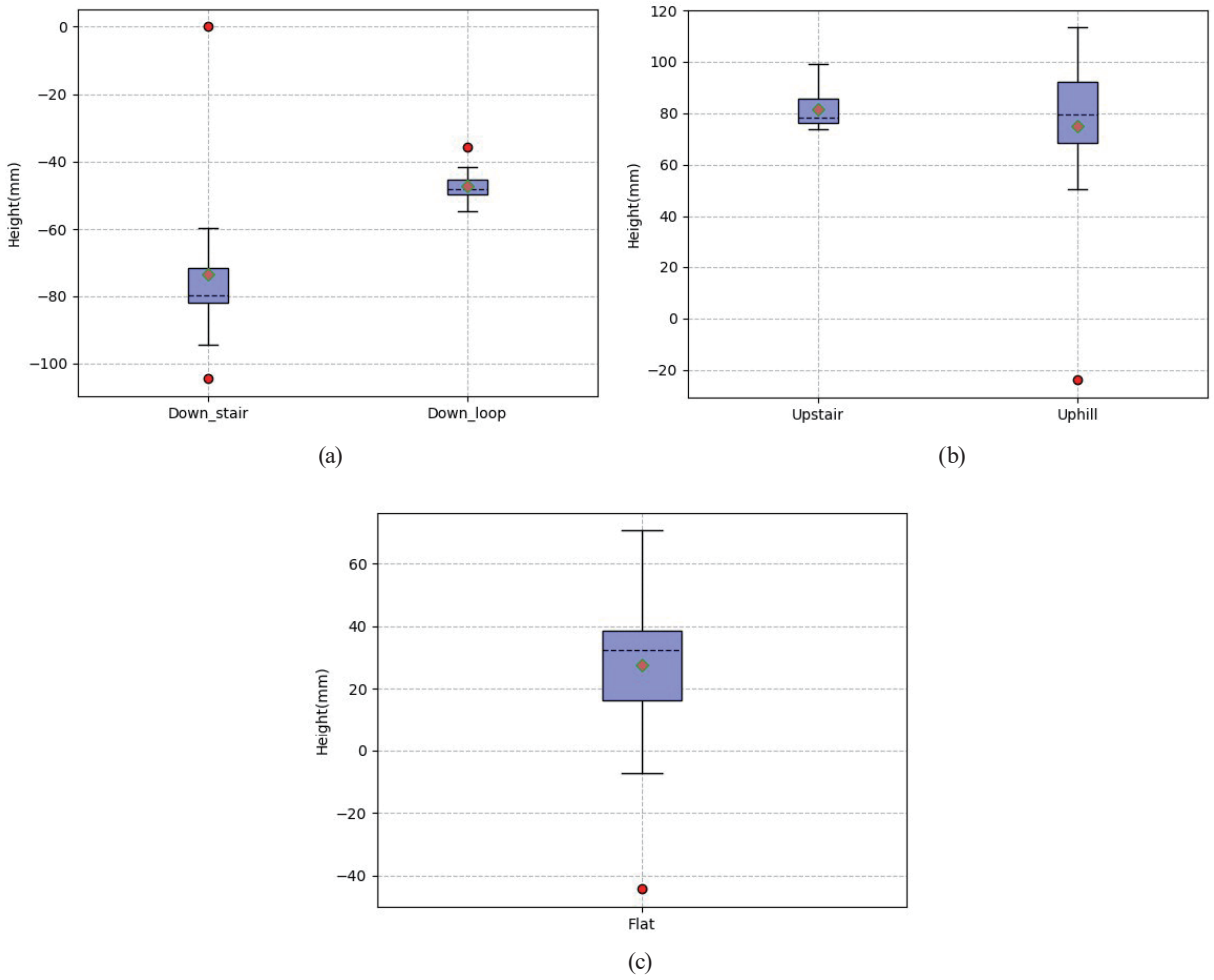


Fig. 6. (Color online) Box and whisker diagrams showing distributions of the relative height difference between the two feet when the heel touch event occurred. (a) Downstairs/downhill. (b) Upstairs/uphill. (c) Flat ground.

kinematic modeling was less than −20, the output result was evaluated as the downstairs/downhill mode.

In this study, the experimental conditions were limited to a step height of 100 mm and a slope angle of 15.5°. If the slope angle is less than 15.5° or the step height is less than 100 mm, the identification accuracy of the DBMK algorithm may decrease markedly. It is still necessary to perform further experiments to evaluate the influence of the experimental conditions on the accuracy of the DBMK algorithm.

After evaluating the pattern recognition of walking straight, upstairs/uphill and downstairs/downhill, the task of recognizing the upstairs, uphill, downstairs, and downhill modes was addressed. For this purpose, we propose the PWD correlation coefficient [Eq. (17)]. Note that the Pearson correlation coefficients between the knee angles for the uphill and upstairs modes and between the knee angles for the downhill and downstairs modes are relatively high; thus, the Pearson correlation coefficient alone cannot solve the problem of motion pattern recognition. To

verify the effectiveness of the PWD correlation coefficient, we carried out upstairs, uphill, downstairs, and downhill experiments.

From Fig. 7, it can be seen that the average recognition rate of the DBMK algorithm for the five motion modes is 87.55%. Among them, the average accuracy of the PWD correlation coefficient matching is 90.86%, and the average accuracy of the dynamic threshold algorithm for detecting heel touch events is 95.99%. It can also be seen from Fig. 7 that the DBMK algorithm has the lowest performance for the uphill motion mode, with a recognition accuracy of only 66.13%, which is mainly caused by the low matching accuracy of the PWD correlation coefficient. Table 3 shows the confusion matrix of the DBMK algorithm. The DBMK algorithm misjudged 15.32 and 7.26% of cases of the uphill motion mode as upstairs and straight walking motion modes, respectively, and the dynamic threshold method failed to detect 11.29% of the heel touch events. In general, the DBMK algorithm can accurately identify the five most common daily activities of the human body, especially those of walking straight, upstairs, downstairs, and downhill.

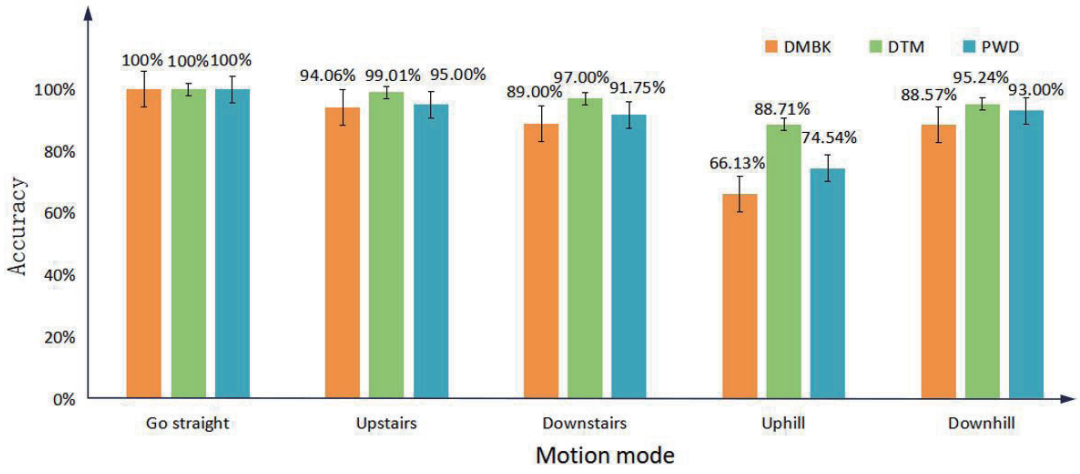


Fig. 7. (Color online) Recognition accuracy rate in different motion modes. "AMC" denotes the success rate of the dynamic threshold algorithm for detecting the heel touch event.

Table 3
Confusion matrix of experimental results.

True label \ Predicted label	0	1	2	3	4	X
0	100%	0	0	0	0	0
1	0	94.06%	0.99%	2.97%	0.99%	0.99%
2	1.02%	0	90.82%	0	7.14%	1.02%
3	7.26%	15.32%	0	66.13%	0	11.29%
4	3.0%	0	4.0%	0	93.0%	0
X	0	0	2.0%	0	4.76%	

0: Walking straight; 1: Upstairs; 2: Downstairs; 3: Uphill; 4: Downhill; X: No heel touch event occurred.

4. Discussion

The dynamic matching algorithm model based on kinematics mainly includes two parts: periodic signal extraction and sample template matching. In periodic signal extraction, two heel touch events are mainly used to extract a period of knee angle data. The higher the detection speed of the heel touch event, the more accurate the detection position, which is beneficial for the subsequent recognition performance. The dynamic threshold algorithm has three important parameters: α, β, and the number of elements n in the time series $S(k)$ in Table 1. We carried out experiments to explore the influence of the coefficients on the algorithm, and the results obtained are shown in Table 4. It can be seen that α should be set between 0 and 1.0, and when α is set to 0.6, the dynamic threshold algorithm achieves the best results. From Table 5, it can be seen that β has little effect on the recognition performance of the algorithm, although when β is set to 10, the recognition performance is highest. Table 6 shows that the dynamic threshold algorithm is more sensitive to the coefficient n than to the other two parameters, and when $n = 7$, set in the experiment in this paper, is replaced with $n = 11$, the detection performance of the algorithm is improved. We utilized the dynamic threshold algorithm (DT) to obtain a heel touch event accuracy rate of up to 95.19% in the five modes, verifying its effectiveness.

In recent years, the rapid development of deep learning technology has also encouraged many researchers to use it for motion pattern recognition.[19,20] Although deep learning has achieved

Table 4
Influence of coefficient α on DT performance.

α	Straight walking	Upstairs	Downstairs	Uphill	Downhill
0.2	100.0%	99.01%	97.05%	88.71%	90.83%
0.4	100.0%	99.01%	97.05%	88.71%	91.67%
0.6	100.0%	99.01%	97.05%	88.71%	94.28%
0.8	100.0%	99.01%	96.08%	87.70%	94.28%
1.0	0	0	0	0	0

Table 5
Influence of coefficient β on DT performance.

β	Straight walking	Upstairs	Downstairs	Uphill	Downhill
0	100.0%	100.0%	99.01%	97.00%	88.71%
5	100.0%	100.0%	99.01%	97.00%	88.71%
10	100.0%	100.0%	99.01%	97.00%	88.71%
15	100.0%	100.0%	99.01%	97.00%	87.71%
20	100.0%	100.0%	99.01%	96.00%	86.29%

Table 6
Influence of coefficient n on DT performance.

n	Straight walking	Upstairs	Downstairs	Uphill	Downhill
3	100.0%	99.01%	89.11%	62.90%	90.00%
7	100.0%	99.01%	97.0%	88.71%	94.29%
11	100.0%	100.0%	100.0%	100.0%	95.19%
15	100.0%	100.0%	98.99%	100.0%	93.4%
3	100.0%	99.01%	89.11%	62.90%	90.00%

Table 7
Performance of DBMK algorithm for different values of coefficient *m*.

m	10%	30%	50%	70%	90%	100%
Go straight	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
Upstairs	74.0%	97.0%	98.0%	95.0%	95.0%	98.0%
Downstairs	86.60%	51.55%	89.69%	91.75%	83.51%	98.97%
Uphill	78.18%	75.45%	68.18%	74.55%	72.73%	0.0%
Downhill	80.00%	92.00%	89.00%	93.00%	63.00%	0.0%

Table 8
Accuracy of motion pattern recognition of different models.

Classifier	KNN	SVM	MLP	DTW
Accuracy rate	69.2%	76.8%	80%	85.6%

very good results in motion pattern recognition, it is difficult to apply it to exoskeleton hybrid control models due to the complexity and ambiguity of the deep learning framework. In this paper, we propose the DBMK algorithm, an important part of which is the PWD correlation coefficient. The performance of this coefficient directly determines the final recognition performance of the motion model, where the coefficient *m* has a great impact on the matching result of the PWD correlation coefficient. In the above experiment, *m* was set to 70%. To explore the influence of *m* on the recognition performance of the DBMK algorithm, we performed six sets of control experiments with *m* set to 10, 30, 50, 70, 90, and 100%. The experimental results are shown in Table 7, from which it can be seen that the straight walking motion mode is not sensitive to *m*, whereas the other four motion modes are more sensitive to *m*, especially the downhill motion mode. On the whole, the DBMK algorithm has the lowest performance for uphill pattern recognition, for which the highest recognition accuracy rate is only 78.18%, indicating that further research is required. In combination with Table 3, it can be seen that the DBMK algorithm misjudges the uphill motion mode as the upstairs motion mode. It can be seen from Fig. 4 that the knee angle waveform in the uphill motion mode is very similar to that in the upstairs motion mode. This may also be the reason why the DBMK algorithm is not sensitive to the uphill motion mode. In addition, it can be seen from Table 4 that when *m* is 70%, the DBMK algorithm has the best comprehensive recognition performance for the five motion modes, and the average recognition accuracy reaches 90.86%.

Elhoushi *et al.* designed a decision tree model to recognize eight different motion modes with recognition accuracy of up to 87.94%.[21] Li *et al.* used a variety of algorithm models to identify five different motion modes,[22] and their recognition results are shown in Table 8. The DBMK algorithm proposed in this paper has an average recognition accuracy of up to 90.86% for the five motion modes, suggesting that it can be applied to motion pattern recognition.

5. Conclusion

In this paper, a dynamic matching algorithm based on kinematics was proposed to solve the problems of the poor interpretability and computational complexity of motion mode recognition

methods based on traditional machine learning and deep learning; the DBMK algorithm was used to recognize motion modes through a kinematic model and template matching. The algorithm consists of two main parts: heel touch event detection and motion pattern recognition. First, the dynamic threshold algorithm is used to detect heel touch events from a pressure signal collected by a plantar membrane pressure sensor, and the accuracy of the DT algorithm for detecting heel touch events in the five motion modes was as high as 95.19% after modifying the parameters α, β, and n. The DBMK algorithm combines kinematic modeling and the PWD correlation coefficient to identify the upstairs, uphill, downstairs, and downhill motion modes, wherein the coefficient m has a great impact on the final recognition performance of the PWD algorithm. For example, when m was set to 100%, the DBMK algorithm did not identify any uphill and downhill motion modes, whereas when m was set to 70%, the DBMK algorithm performed well and the average recognition accuracy of the five dynamic motion modes was as high as 90.86%. Experimental results show that the dynamic matching algorithm based on kinematics can effectively identify five common motion modes.

References

1 S. Jatsun, A. Malchikov, and A. Yatsun: Int. Russian Automation Conf. (IEEE, 2020) 470–475. https://doi.org/10.1109/RusAutoCon49822.2020.9208173
2 A. Yatsun and S. Jatsun: 2018 Global Smart Industry Conf. (IEEE, 2018) 1–5. https://doi.org/10.1109/GloSIC.2018.8570092
3 M. Hong, G. Kim, and Y. Yoon: Robotica **37** (2019) 2209. https://doi.org/10.1017/S0263574719000845
4 J. Proud, D.. Lai, K. Mudie, G. Carstairs, D. Billing, A. Garofolini, and R. Begg: HFES (2020). https://doi.org/10.1177/0018720820957467
5 M. Nikolenko and D. Kotin: 2017 18th Int. Conf. Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM, 2017). https://doi.org/10.1016/S0011-5029(62)80013-3
6 P. Tran, S. Jeong, K. Herrin, and J. P. Desai: IEEE Trans. Med. Rob. Bionics **3** (2021) 606. https://doi.org/10.1109/TMRB.2021.3100625
7 M. Elhoushi, J. Georgy, A. Noureldin, and M. Korenberg: IEEE Trans. Instrum. Meas. **65** (2015) 208. https://doi.org/10.1109/TIM.2015.2477159
8 M. Elhoushi, J. Georgy, A. Noureldin, and M. Korenberg: IEEE Trans. Intell. Transp. Syst. **18** (2017) 1662. https://doi.org/10.1109/TITS.2016.2617200
9 I. Klein, Y. Solaz, and G. Ohayon: IEEE Sens. J. **18** (2018) 7577. https://doi.org/10.1109/JSEN.2018.2861395
10 T. Ogata, J. Tan, and S. Ishikawa: IEICE Trans. Inf. Syst. E89-D (2006) 281. https://doi.org/10.1093/ietisy/e89-d.1.281
11 X. Wang, S. Yang, Y. Cao, Y. Ding, and Z. Zhang: 2020 IEEE Int. Conf. Information Technology, Big Data and Artificial Intelligence (IEEE, 2020) 407–411. https://doi.org/10.1109/ICIBA50161.2020.9277504
12 J. Song, Zhu, Y. Tu, H. Huang, M. Arif, Z. Shen, X. Zhang, and G. Cao: Appl. Sci. **10** (2020) 3358. https://doi.org/10.3390/app10103358
13 T. Zhen, J. Kong, and L. Yan: Complexity **2020** (2020) 17. https://doi.org/10.1155/2020/8672431
14 Khan, Lee, and Kim: 2008 30th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society (IEEE, 2008) 5172–5175. https://doi.org/10.1109/IEMBS.2008.4650379
15 J. Song, A. Zhu, Y. Tu, Y. Wang, M. A. Arif, H. Shen, Z. Shen, X. Zhang, and G. Cao: Sensors **20** (2020) 537. https://doi.org/10.3390/s20020537
16 R. Zhao, X. Ma, X. Liu, and F. Li: IEEE Sens. **21** (2021) 5022. https://doi.org/10.1109/JSEN.2020.3033278
17 Z. Lu, A. Narayan, and H. Yu: 2020 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE, 2020) 4091–4097. https://doi.org/10.1109/IROS45743.2020.9341183
18 L. Huang,, H. Li, B. Yu, X. Gan, B. Wang, Y. Li, and R. Zhu: Sensors **20** (2020) 2263. https://doi.org/10.3390/s20082263
19 S. Wang, Q. An, S. Li, G. Zhao, and H. Sun: IEEE Sens. **21** (2021) 10007. https://doi.org/10.1109/JSEN.2021.3057592

20 K. Pawlowski, S. Chakravarty, Y. Xie, and A. K. Joginipelly: 2020 IEEE Int. Conf. Big Data (IEEE, 2020) 5381-5386. https://doi.org/10.1109/BigData50022.2020.9377733
21 M. Elhoushi, J. Georgy, A. Noureldin and M. J. Korenberg: IEEE Trans. Instrum. Meas. **65** (2016) 208. https://doi.org/10.1109/TIM.2015.2477159
22 G. Li, K. Yao, E. Geng, J. Lin, and Y. Pang: 2021 Int. Wireless Communications and Mobile Computing. (IEEE, 2021) 1931–1935. https://doi.org/10.1109/IWCMC51323.2021.9498672

About the Authors

Tao Zhen received his B.S. degree in mechanical engineering from Shandong Jianzhu University, China, in 2014. Since 2018, he has been studying for a Ph.D. degree in mechanical engineering at the School of Engineering, Beijing Forestry University. His current research interests are in exoskeleton robots and wearable intelligent systems design. (zhentao@bjfu.edu.cn)

Hao Zheng received his B.S. degree in automation from Beijing Forestry University, Beijing, China, in 2019. He is now a master's student in the School of Engineering, Beijing Forest University, majoring in mechatronic engineering. His current research interests include exoskeleton robots and smart electrical equipment. (zhenghao@bjfu.edu.cn)

Lei Yan received his M.S. degree from Jilin University, China, in 2004 and his Ph.D. degree from Kyungpook National University, Daegu, in 2010. Since 2010, he has been a professor at Beijing Forestry University, China. He has authored more than 60 papers published in various journals and completed 10 scientific research projects. His research interests are in MEMS, exoskeleton robots, and machine vision. (mark_yanlei@bjfu.edu.cn)