

Research Article

A Control and Posture Recognition Strategy for Upper-Limb Rehabilitation of Stroke Patients

Xian Yu ¹, Bo Xiao ², Ye Tian ³, Zihao Wu ⁴, Qi Liu ⁴, Jun Wang ⁴, Mingxu Sun ⁵,
and Xiaodong Liu ⁶

¹NARI Group Co., Ltd. (State Grid Electric Power Research Institute Co., Ltd.), Nanjing 211000, China

²School of Chemistry and Materials, Nanjing University of Information Science and Technology, Nanjing, Jiangsu 210044, China

³China Information Communication Technologies Group Corporation (CICT), Wuhan, China

⁴School of Computer and Software, Nanjing University of Information Science and Technology, 210044 Nanjing, China

⁵School of Electrical Engineering, University of Jinan, China

⁶School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK

Correspondence should be addressed to Mingxu Sun; cse_sunmx@ujn.edu.cn

Received 18 November 2020; Revised 8 February 2021; Accepted 30 March 2021; Published 18 May 2021

Academic Editor: Jun Cai

Copyright © 2021 Xian Yu 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.

At present, the study of upper-limb posture recognition is still in the primary stage; due to the diversity of the objective environment and the complexity of the human body posture, the upper-limb posture has no public dataset. In this paper, an upper extremity data acquisition system is designed, with a three-channel data acquisition mode, collect acceleration signal, and gyroscope signal as sample data. The datasets were preprocessed with deweighting, interpolation, and feature extraction. With the goal of recognizing human posture, experiments with KNN, logistic regression, and random gradient descent algorithms were conducted. In order to verify the superiority of each algorithm, the data window was adjusted to compare the recognition speed, computation time, and accuracy of each classifier. For the problem of improving the accuracy of human posture recognition, a neural network model based on full connectivity is developed. In addition, this paper proposes a finite state machine- (FSM-) based FES control model for controlling the upper limb to perform a range of functional tasks. In the process of constructing the network model, the effects of different hidden layers, activation functions, and optimizers on the recognition rate were experimental for the comparative analysis; the softplus activation function with better recognition performance and the adagrad optimizer are selected. Finally, by comparing the comprehensive recognition accuracy and time efficiency with other classification models, the fully connected neural network is verified in the human posture superiority in identification.

1. Introduction

There are more than 10 million new strokes per year worldwide [1], and stroke is still the leading cause of death and disability among adults [2]. With the accelerating aging of the society and the prevalence of unhealthy lifestyles, stroke diseases have shown explosive growth and are getting younger. Strokes are characterized by high incidence and disability, with World Health Organization data showing that strokes have a disability rate of up to 80%. The economic burden is 10 times greater than that of myocardial infarction. Therefore, prevention and treatment are urgent, and the rehabilitation system for patients needs to be improved.

Stroke patients' recovery of limb function is one of the most important aspects of rehabilitation. At present, there are several different types of rehabilitation therapy in clinic, such as electromyographic feedback therapy, electrical stimulation therapy, and motor imagery mental training therapy, while the most highly regarded in clinical practice is functional electrical stimulation (FES), with stimulation electrodes worn on the limbs of stroke patients consisting of the controller send out stimulation signals to electrically stimulate specific muscles to enable the limb to perform various types of functional rehabilitation or to perform daily activity, which in turn leads to the recovery of limb function. Stroke patients need to perform specific

functional tasks in the process of rehabilitation, so an efficient control strategy needs to be designed. At the same time, due to the lack of existing public datasets, it is urgent to establish a database, design algorithms to analyze sensor device data, and identify the upper-limb posture movement of stroke patients. This can provide reference for the rehabilitation and rehabilitation effect of stroke patients. It provides an effective solution.

The paper is divided as follows: Section 2 presents the related work on this field. Section 3 and Section 4 demonstrate the methodologies. Section 5 shows the results and discusses the findings. Finally, Section 6 concludes the paper.

2. Related Work

The related work in this paper concerns FES control and upper-limb posture recognition, and the following sections will focus on these two components.

2.1. Related Work on FES Control. Functional electrical stimulation (FES) is often used for rehabilitation treatment of stroke or spinal cord injury. For individuals with motor nervous system damage, FES can activate the skeletal muscle of paralyzed patients by implementing low-level electrical pulses on motor neurons [3] and activate corresponding muscles according to different expected movements [4]. Since Liberson et al. first used FES to rehabilitate a prolapsed foot in 1960 [5], FES has been proved to be one of the important methods to treat stroke rehabilitation or spinal cord injury. Sabut et al. proposed a combination of FES and general rehabilitation program, which has a significant effect on improving the muscle strength of patients [6]. It is not easy to use FES to control the target skeletal muscle at a high level. When FES stimulates the muscle, the muscle response to the stimulation is nonlinear and time-varying, and individuals with nervous system damage are often accompanied with time delay [7]. There are open-loop, closed-loop, and state machine-based control strategies in FES system to deal with the above problems.

Open-loop control strategy is a simple but reliable control strategy, which is widely used in various control systems. However, due to the low precision of open-loop control and the lack of automatic correction ability, closed-loop control solves this problem [8]. The closed-loop FES control system usually consists of feedback signals, error detection and correction processes, and a model used to determine the output of the system. For example, Zhang et al. proposed an electromyography-based closed-loop torque control strategy of functional electrical stimulation [9], and FES-evoked electromyography (EMG) was used to reflect the state of the stimulated muscle, so as to compensate the muscle strength adaptively. Compared with open-loop control, the closed-loop FES system using surface electronics (sEMG) biases feedback from bilateral arms for enhancing upper-limb stroke rehabilitation [10]. Dodson et al. used a closed-loop controller to compensate for electromechanical delay (EMD) to increase the energy expenditure of the hybrid neural prosthesis and prolong the onset of muscle fatigue [11].

Compared to open-loop control, closed-loop control strategies have better automaticity and adaptivity.

Generally, a FSM controller is composed of a set of states, state transition conditions, input signals, and output functions [12]. Each “state” corresponds to a movement stage, and the “state transition condition” realizes the exit of FES from each movement. Condition, finite state control usually contains multiple states, each action corresponding to each state is predefined, and the transition between states is determined by the current state and artificial signals. Finite state machine has been proved to be an effective control method to realize the functional tasks of upper limbs. For example, the upper-limb auxiliary system designed by Wang et al., which combines FES with robotic exoskeleton, realizes the control of finite state machine based on embedded environment. The finite state machine is designed as an advanced controller, which sends commands to the embedded controller in real time to assist the grasping task realized by the assistant [13]. The experimental results proved the effectiveness of this method.

2.2. Related Work on Posture Recognition. The human body posture recognition mode is divided into vision-based human body posture recognition and sensor-based human body posture recognition [14, 15]. The first one mainly uses support vector machine [16, 17], hidden Markov, and other algorithms [18]. The recognition success rate or the efficiency of the algorithm is ideal, but it is more environment dependent, the conditions are limited, and the sensor used to capture the human body posture has the characteristics of small size, high sensitivity, and is easy for users to carry [19, 20].

Abobakr et al. proposed a holistic posture-based analysis model [21] that uses the Kinect. The sensor acquires the data, estimates the joint angle of the human body by inputting the depth image and uses a deep convolutional neural network model for the joint perspectives for regression [22], uses comprehensive training images to simulate different body movement tasks, and obtains highly generalized learning models to achieve higher attitude prediction rate [23]. In 2019, Xu et al. implemented depth information and skeletal tracking based on Microsoft Kinect V2 sensors to perform human posture recognition [24], and based on this, human fall detection was implemented. First, a Kinect V2 sensor was used to process the human joint data generated by the skeletal tracker, and then, the optimized BP neural network is used for posture recognition and based on this to detect falls, by training the neural network using a dataset generated by the Kinect tracker and using other body trackers for testing. Finally, posture recognition and fall detection were experimentally validated and tested in real time over the entire operating range of the sensor. The overall accuracy of the NITE tracker used for the drop test was experimentally 98.5%, and the worst accuracy was 97.3 percent. University of Brahem et al. mounted an accelerometer on the foot to track and identify foot movements [25]. University of Schwarz et al. used a MEMS sensor to capture and recognize hand movements, which in turn accomplished a medical office doctors’ human-computer operation with a computer

[26]. The feedback from the sensors effectively reduces the possibility of injury during jumping [27]. Lim et al. at Nanyang Technological University, Singapore, invented a wearable wireless human arm motion capture sensing system [28] that captures and recognizes human posture using acceleration sensors and bending sensors for human-computer interaction in medical applications for stroke patients in recovery training. Wang et al. analyzed the signal characteristics of accelerometers and gyroscopes on representative [29], the feature information is extracted, a DT model-based classifier is proposed, and the angle deviation is weighted by an improved PCA algorithm. On average, the experimental results proved that the average accuracy of the pose of other was close to 97.1%, improving the PCA-based angular bias method judgment accuracy.

In 2018, Cai et al. presented a process analysis and Fisher vector-based encoded human action recognition framework [30]; first, by applying Procrustes analysis and local retention projections, apply pose-based features extracted from silhouette images. The distinguishing shape information and the local manifold structure of the human pose are preserved and remain invariant for translation, rotation, and scaling. After the pose features are extracted, a recognition framework based on Fisher vector coding and multiclass support vector machines is used for the human motion classification, and the experimental results demonstrated the effectiveness of the method.

3. Control Strategies

The FES controller is generally composed of a series of preset states and state transition conditions, input signals, and output functions. In this case, each “preset state” corresponds to a movement phase, and the “output function” of each state performs a gradual change of muscle stimulation to its respective targets (the target can be zero) and finally maintains it on these targets. “Condition of state transition” means the precondition of exiting each movement stage. The potential “input signal” set of FSM controller can be the data measured by accelerometer units connected to different parts of the body, angle data, button status, and clock time.

3.1. FSM Controller for Upper-Limb FES. The general existence form of FSM controller is composed of a series of movement phases with time sequence (real rectangle) and natural transition (solid arrow) between each movement phase. Figure 1 shows the transition between states. There is a neutral phase in FSM, that is, the first state, which does not involve in stimulating any muscle parts. Users can customize the total number of stages of FSM controller, but not less than 2 states, which is determined by the selected execution task. FSM returns to the first phase, the neutral phase, whenever the transition conditions of the final phase are met. Therefore, the execution of functional tasks is always in the neutral state. Dashed arrows indicate transitions between special phases (such as default timeout and emergency stop), and any phase can transition to a neutral phase. It is specified that normal transition has higher priority than abnormal transition.

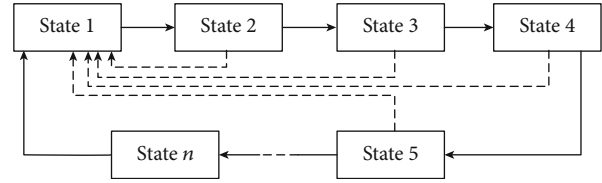


FIGURE 1: The transition between each state.

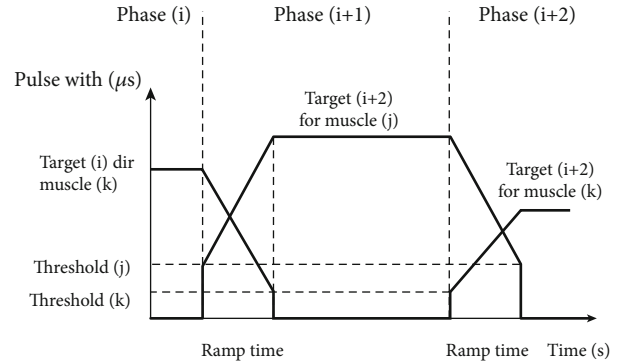


FIGURE 2: Rise to a threshold (before the rise) and fall from a threshold (after the ramp down).

The timing of the transition between phases is determined by the condition of the state transition, and the timing of the transition is expressed by the input signal and the current state. GUI can be used to customize the parameters of FSM, including the number of states, stimulus parameters of each state (stimulation thresholds, ramps, and target), and state transition conditions (timeout, combination logic, angle triggers, etc.).

Ramp time is a user-defined parameter in FES, which represents the ramp time from the current target to the new target. Figure 2 illustrates the variation of pulse width. The ramp rate is determined by the ramp time and two consecutive nodes in the stimulus curve. In this way, when the stimulus level is different, the shorter the ramp time, the higher the ramp rate.

The realization of ramp is determined by the frequency of FSM. In this paper, we decide to use 20 Hz; then, the minimum time step is 0.05 seconds. This prevents users from noticing any delay.

Phase conversion is determined by the current input signal and conversion conditions. The FSM controller can obtain up to four acceleration data to capture the motion of each part of the upper limb (i.e., hand, upper arm, and lower arm). In this case, the acceleration data of accelerometer will be transmitted to FSM controller in real time during the execution of functional tasks. In order to improve the flexibility of the system, this paper uses logical operators (N/A and OR) to combine two Boolean conditions to create conversion rules.

Let us discuss an example to show how FSM can customize settings for specific FES tasks. As shown in Figure 3, this FES task consists of five phases: “neutral,” “reach for door,” and “grass handle.” In stage 5, the forearm extensor is stimulated to reach the state of releasing the door handle. The

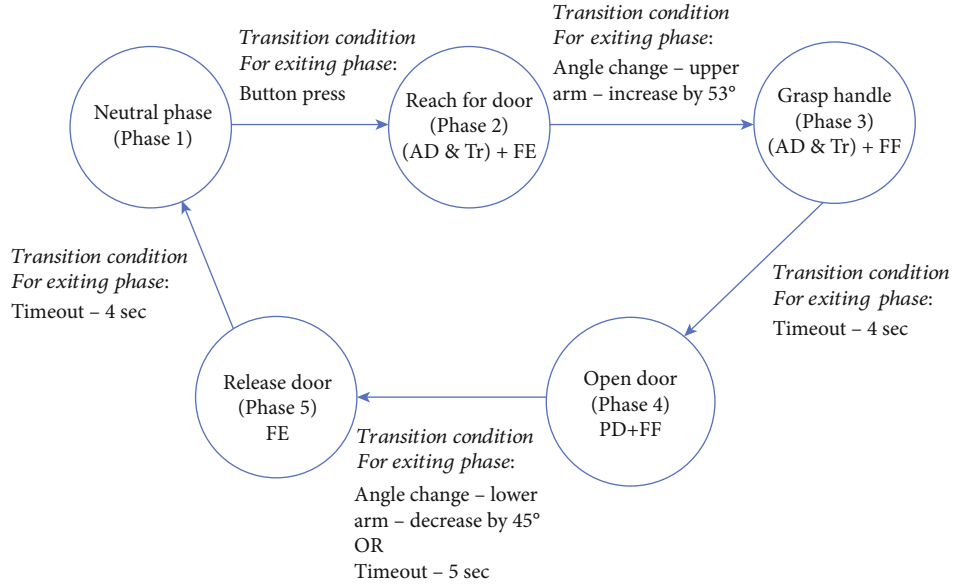


FIGURE 3: Transition between phases.

transition between states is an instantaneous event after the conditions are met. This example will be carried out in the experimental part, and the realization of each part of FSM is described in detail.

The execution of FES task is reflected in the FSM controller, which can be regarded as the state transition of each movement stage according to a certain time sequence and transition conditions. Table 1 lists the Boolean conditions of each phase transition in the door opening task. Two accelerometers are used to record the movements of the lower arm and the upper arm, respectively, which can be used as the transition of phases 2 to 3 and 4 to 5, and it can also be used as a condition for triggering phase transition. In the transition from stage 4 to stage 5 in this example, the logical operator is OR, which means that only one of condition A and condition B is satisfied and the phase transition be carried out, that is, the upper arm angle is reduced by 45° and the phase 4 is kept for 5 seconds, and the state transition can be triggered. It should be noted that the transition between phases depends not only on the state transition conditions, but also on the current state.

3.2. Implementation of the Finite State Machine Controller. This paper uses MATLAB and Simulink to implement a real-time FSM controller under the Windows platform. The real-time online data acquisition, processing, and stimulation parameter control are realized by Simulink. The components and input/output of the FES control system are described in Figure 4. The FSM controller can input the button pressing signal in real time, time-out clock and three-axis acceleration data, and real-time output of stimulus pulse width (μ sec), pulse amplitude (MA), and waveform. The waveform is pre-set and fixed; the Simulink simulation system runs at 20 Hz, implementing real-time angle tracking, angle triggering, FSM controller and security review.

TABLE 1: Transition rules of the “open a door” task.

Transition between phases	LO (logical operator)	Factor A	Factor B
1 to 2	Not/and	Press the space bar	Invalid
2 to 3	Not/and	Increase upper arm by 53°	Invalid
3 to 4	Not/and	Hold for 4 sec	Invalid
4 to 5	OR	Decrease upper arm by 45°	Hold for 5 sec
5 to 1	Not/and	Hold for 4 sec	Invalid

The real-time input of FSM controller includes the absolute angle value of x -axis and vertical direction measured by Xsens unit, the “space bar” button in GUI is used as the switch state button, the “enter” key is used as the emergency state button, and the time-out clock time is also included.

The design of FSM controller includes the design of state transition control, stimulation output control, and the research of improving the robustness of angle trigger. The output of FSM will be transmitted to the safety module in real time. The safety module is located between the controller and the stimulator. The safety block can prevent the pain caused by improper stimulation level. Because the safety block will limit the pulse width of a single pulse, the pulse amplitude, i.e., the total charge, and the maximum step size of the ramp, the safety block will stop the stimulation of the stimulator when any limit is exceeded to verify the security of the whole system. Figure 5 is the flow chart of FES control system of upper limb.

4. A Proposed Posture Recognition Algorithm

4.1. Data Acquisition Equipment. In this paper, the MPU6050 sensor module that satisfies the above

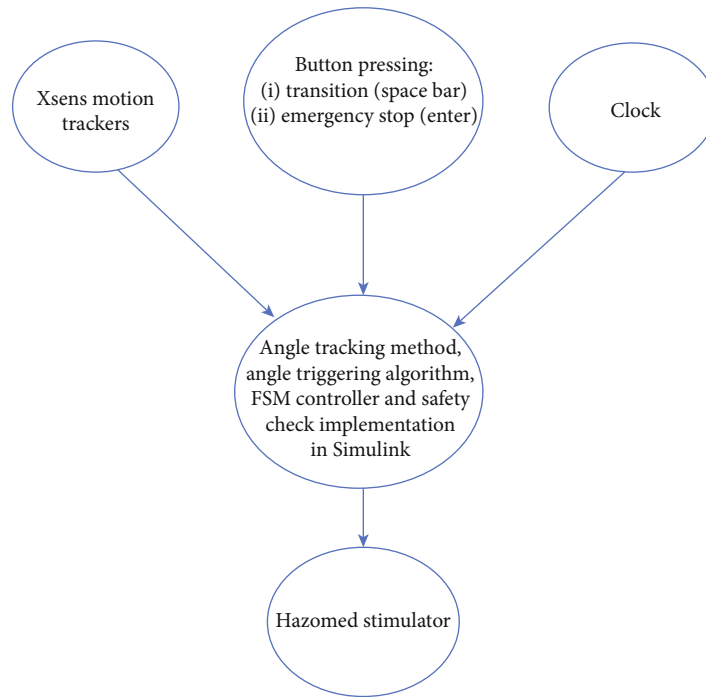


FIGURE 4: The data flow in FES control system.

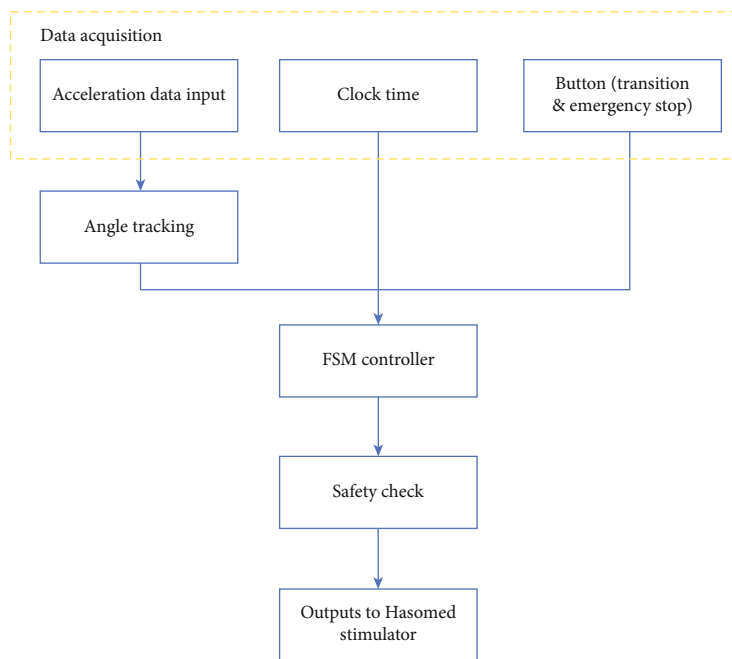


FIGURE 5: The flow chart of upper-limb FES control system.

characteristics is used as a data acquisition device to provide a reliable data source for subsequent research work.

The MPU6050 is a scalable digital motion sensor that integrates a 3-axis MEMS accelerometer and a 3-axis MEMS gyroscope processor, which accurately tracks fast and slow movements. The data collection device is shown in Figure 6. The measurement range of the sensor is user-definable, and the accelerometer can sense ranges of ± 2 g, ± 4 g, ± 8 g, and ± 16 g. The angular velocity can be sensed in the

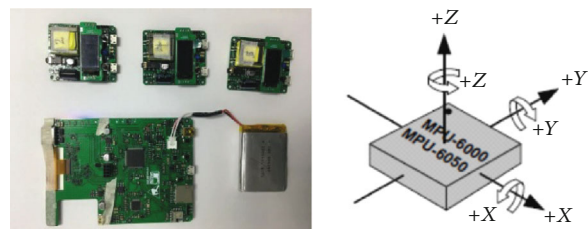


FIGURE 6: The data collection device.

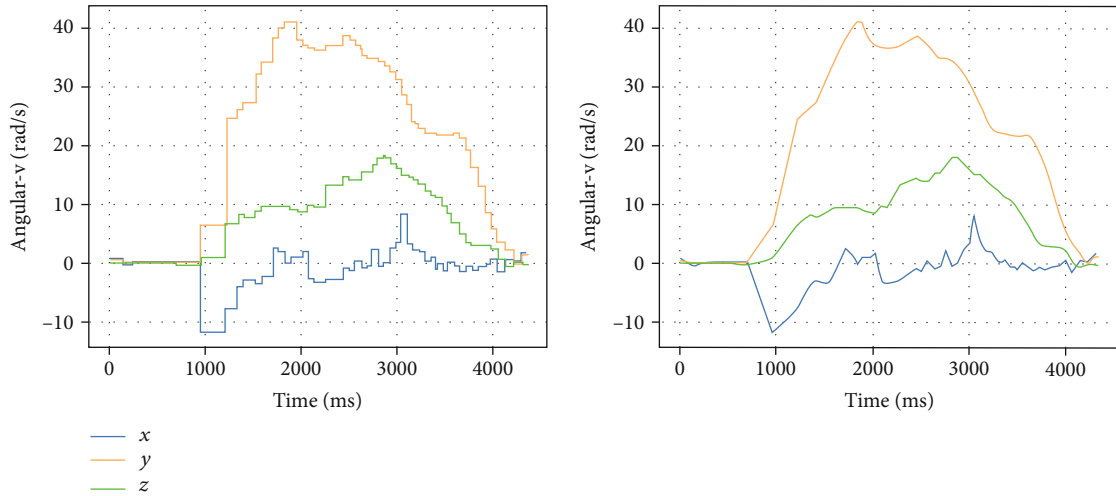


FIGURE 7: Sensor data waveforms before and after weight removal.

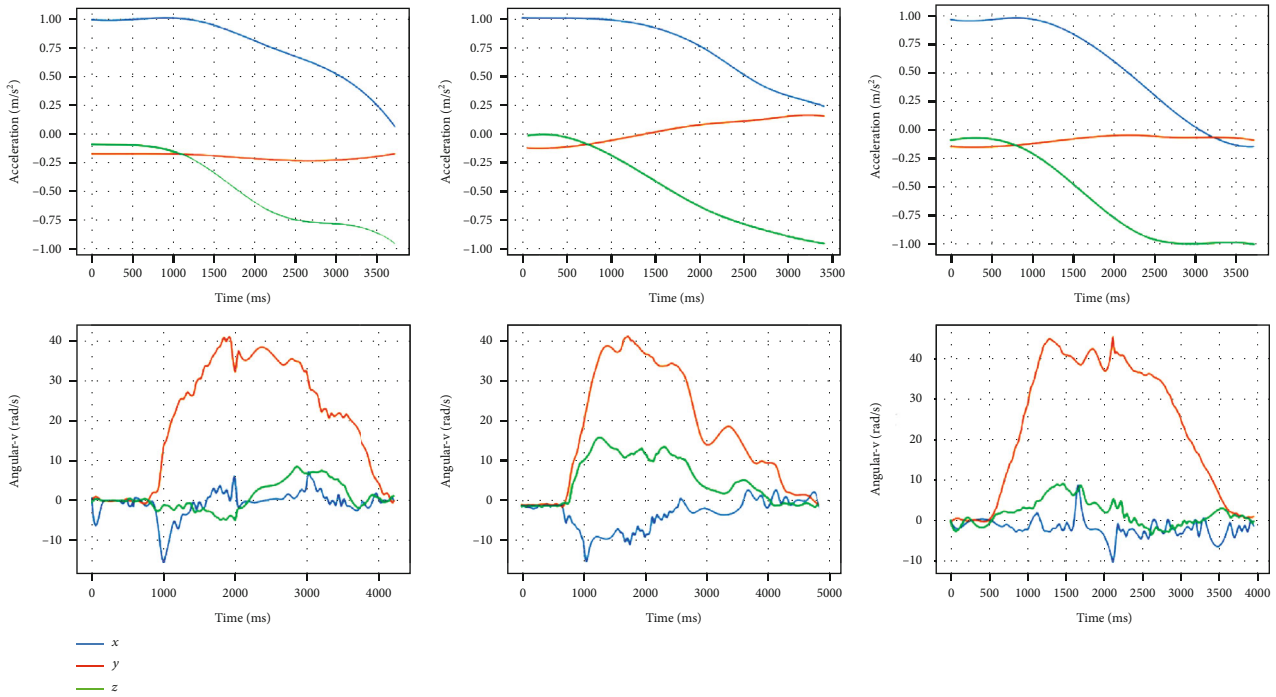


FIGURE 8: Waveform diagram after preprocessing of the side lift data.

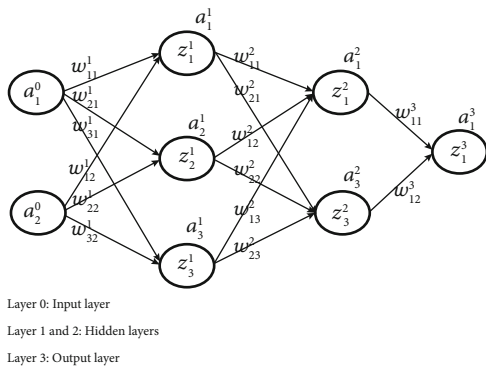


FIGURE 9: Structural diagram of a fully connected neural network.

range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$ (dps). In the data acquisition process, the MPU6050 first puts the calculated values into registers, and then, the microcontroller reads them via I2C.

4.2. Data Preprocessing. To further process the raw dataset, the dataset was deweighted, using the gyroscope data as an example, and the waveforms before and after deweighting are shown in Figure 7.

Data sawtooth has been eliminated, but still not smooth enough, in order to complete part of the missing value, the need to interpolate the dataset to get a smoother interpolation function and the use of three-sample interpolation on

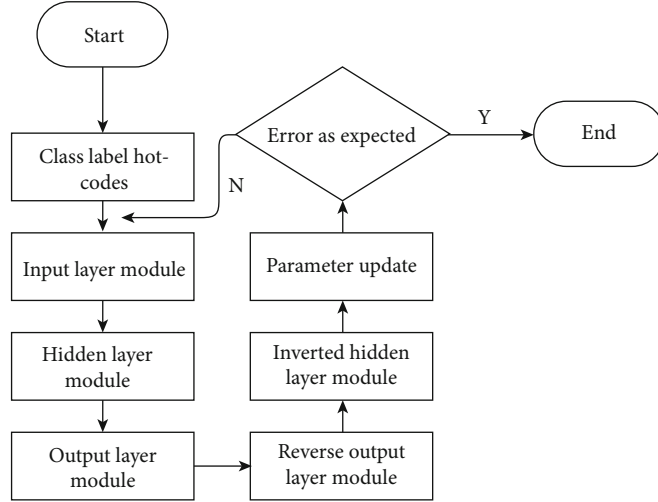


FIGURE 10: Posture recognition schemes for fully connected neural networks.

the dataset to deal with the processing of A, B, C three sensors of the attitude signal shown in Figure 8.

4.3. Fully Connected Neural Network Model. The experiments are mainly conducted using time domain analysis for feature extraction, with N denoting the number of rows of data in a time window and i denoting the row of data, and the selected variance, range, and interquartile range as features define as follows:

$$\sigma_X^2(t) = \lim_{n \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N [X_i(t) - \mu_X(t)]^2 = E[X^2(t)] - (E[X(t)])^2,$$

$$R = X_{\max} - X_{\min},$$

$$\text{IQR} = Q^*3 - Q^*1. \quad (1)$$

The simple structure of a fully connected neural network is shown in Figure 9,

where a_i^l denotes the output of the neuron, where l denotes the number of layers and i denotes the neuron number; z_i^l denotes the output of the inactivated neuron, where l denotes the number of layers and i denotes the neuron number; w_{ij}^l denotes the weighting factor of the neuron. The fully connected neural network obtains the output as the input of the next layer neuron through the multiplication and accumulation of the input data and the weight and then calculates the activation function to realize the forward propagation calculation. According to the error between the final output layer result and the expected result, the weight parameters are adjusted by the back propagation algorithm until the error between the output and the expected result are acceptable.

The experimental posture recognition scheme based on a fully connected neural network is shown in Figure 10.

Hot codes are performed on the labels of the posture dataset to convert the label variables into a form that the neu-

TABLE 2: Stimulation parameters for each channel at different phases.

Phase Channel	1	2	3	4	5
1	0	108	108	0	0
2	0	54	0	0	72
3	0	0	72	72	0
4	0	0	0	90	0

ral network can easily exploit to model operational efficiency as well as the nonlinear capabilities of the model.

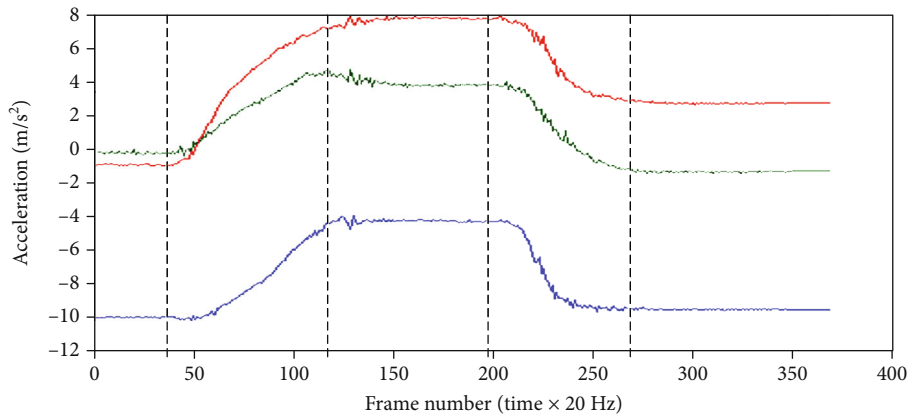
A fully connected neural network model is constructed. The fully connected network model constructed in this paper consists of four components. The first one is the input layer module, which is responsible for inputting the format of the posture data and the initialization task of neuron parameters at each layer during the first execution, setting for each reading of a set of 1590×6 pose matrix data. The hidden layer module consists of a hidden layer containing 30 neurons, the number of layers is determined by comparing the recognition rate and is responsible for the upper layer neurons, the output data are weighted and summed, and the activation function is used to generate the input values from the lower layer neurons. The output layer module is responsible for obtaining the predicted probability values for the six postures from the incoming data from the upper layer neurons. The tuning module is responsible for calculating the activation value for each neuron, the loss of each layer based on the activation value, and the parameter gradient from the output of the layers start to make parameter adjustments going forward. The posture dataset is trained by the above method to derive the final recognition model.

5. Experiment

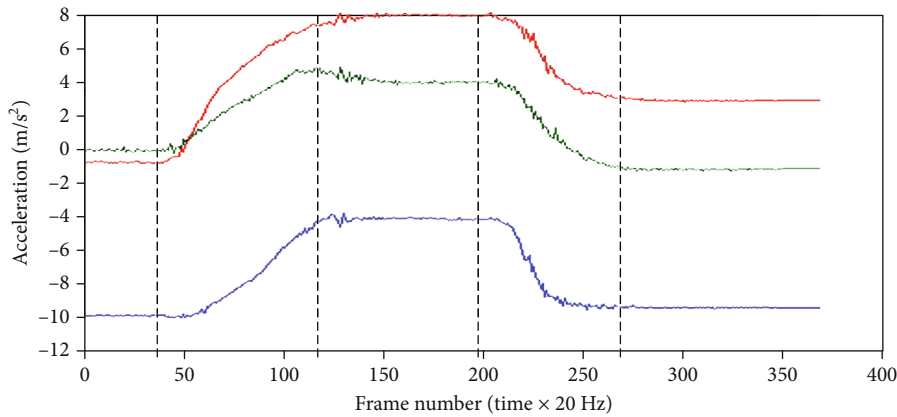
5.1. FES Control Test. This paper uses the “open door” task to test the FSM controller. The output data of the main monitoring controller includes the following: the accelerometer

TABLE 3: Stimulated muscles and their stimulation changes at various stages.

Phase	Output
Neutral	No stimulation
Reach for door	Anterior deltoid, triceps, and forearm extensors ramp from threshold to target and then stay on the targets.
Grasp	Both triceps and anterior deltoid rise to the target. Forearm flexors go from threshold to target. Both channels are at the target location. Forearm extensors shut down by climbing to a threshold and then decreasing to zero.
Open door	Forearm flexors ramp towards the next target. Posterior deltoid goes from threshold to target. Both channels stay at their target location. Anterior deltoid and triceps turn off by ramping to threshold and then decreasing to zero.
Release	Forearm extensors ramp from threshold to target and stay at the target. Posterior deltoid and forearm flexors turn off by ramping to threshold and then decreasing to zero.



(a)



(b)

FIGURE 11: Acceleration data. (a) Upper arm and (b) forearm.

signal of the Xsens unit of the upper arm and forearm, the change of the vertical angle after each state conversion, and the phase number and pulse width of each part of the muscle.

Before running FSM controller, it is necessary to install the Xsens motion tracking software. The software can directly collect the real-time acceleration data of Xsens inertial sensor unit on MTX hub from MATLAB. Xsens system samples the sensor data at the frequency of 100 Hz. Refer to Figure 3 for the muscle parts and transition conditions

involved in the transition of each stage. The specific stimulation parameters can be seen in Tables 2 and 3. The “stimulus threshold” and “maximum comfortable stimulation” are the default values, which are $360 \mu\text{s}$ and $0 \mu\text{s}$, respectively.

The data is collected by healthy subjects in real time when executing the “open a door” task. The dotted lines in the figure below indicate the transition between states. Two Xsens are, respectively, installed on the forearm and the upper arm to trigger at the starting angle. The corresponding acceleration data is shown in Figure 11.

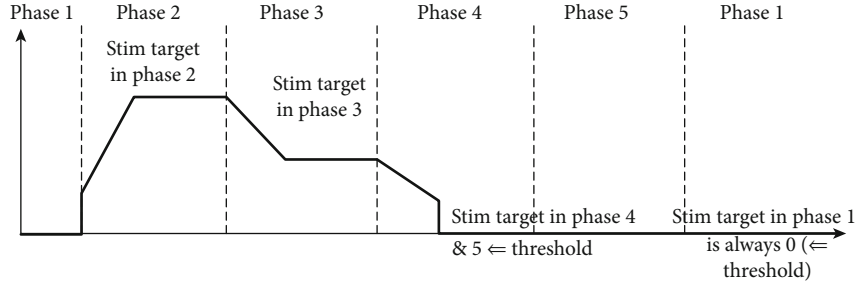


FIGURE 12: The stimulating curve of anterior deltoid and triceps at different phases.

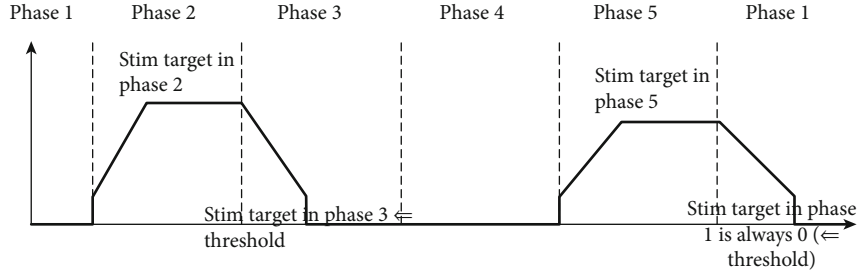


FIGURE 13: The stimulation curve of forearm extensors at each phase.

Figures 12 and 13 show the stimulation of part of the muscle tissue at all phases of movement in the example task of “opening the door” (see Figure 3). The stimulated muscles corresponding to each stage will ramp to the target level of that phase.

5.2. *Neural Network Test.* In order to verify the effectiveness of the fully connected neural network model for human posture recognition, this section takes the six human posture data collected above as an example and performs experimental validation.

The experimental dataset contains the six classical postures of forward flattening, lateral flattening, upward elbow bending, bent elbow backward, wrist upward bending, and horizontal elbow flexion MEMS sensor signals; in order for the pose dataset to be applied to the neural network model, the dataset needs to be preprocessed first. Since the completion time required for various postures varies, the length of the sensor signals collected for the posture samples is inconsistent, so as not to lose the original information of the attitude, it requires adding the original signal data to make the data window consistent. Before performing the experiments, this paper starts with a procedure to find the longest pose sample for the prelift, with a completion time of 5.3 seconds, and to add all the sensor data through three-sample interpolation for the dataset plus windows, interpolation is complete, splicing three sensor data, so that each attitude of the data sample then becomes 1590*6 in the form of a two-dimensional matrix. When solving multiclassification problems using neural networks, the labels need to be digitized, and the digitized class labels converted to binary matrix representation, such an operation is called creating dummy variables (one hot encoding) from categorical variables. As an example, the anterior flat-

TABLE 4: Identification results for different numbers of hidden layers.

Number of hidden layers	Average recognition rate	Time (seconds)
1	91.27%	14.576
2	81.34%	18.743
3	94.08%	20.492
4	84.18%	26.533
5	89.87%	23.33
6	79.41%	31.106

TABLE 5: Identification results for different activation functions.

Activation function	Relu	Softplus	Sigmoid	Tanh	Softsign
Average accuracy	91.31%	93.07%	37.84%	81.96%	90.33%

tened pose data used in this paper is transformed into the following labels: [0, 1, 0, 0, 0, 0].

In order to study the effect of the number of hidden layers on the recognition accuracy and recognition efficiency, this paper investigates the recognition accuracy of hidden layers 1 to 6 and the time taken; the number of neurons was all 30, and the judgment index was the recognition accuracy. The comparison results are shown in Table 4. In addition, the effects of different activation functions and optimizers on recognition accuracy are compared. The experimental results are shown in Tables 5 and 6.

To summarize the above comparative experiments, the fully connected neural network selected a 3-layer hidden layer structure with an activation function of softplus as well

TABLE 6: Identification results of different optimizers.

Optimizer	adam	rmsprop	sgd	adadelta	adagrad	adamax
Average accuracy	94.25%	96.01%	13.07%	93.17%	97.19%	94.35%

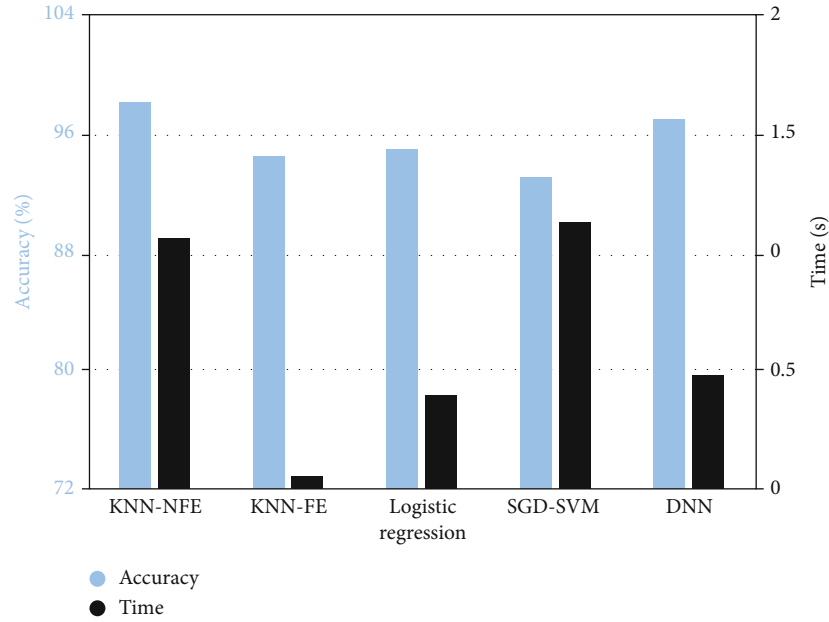


FIGURE 14: Accuracy and computation time of each algorithm (blue represents the accuracy, and black represents the time).

as an adaptive gradient descent optimizer. And three ten-fold cross-validation to take the mean value, the recognition rate, and duration of each algorithm are calculated as shown in Figure 14.

Known datasets without feature extraction retain good pose information, and the KNN model has a very good handle on such pose datasets, good recognition performance (KNN-NFE) with up to 98% recognition accuracy. However, due to the large sensor signal data, the resulting computation time is costly and takes as much as 1 second. In contrast, the calculation time of KNN classifier after feature extraction has been shortened by an order of magnitude and improved greatly, but due to the pose information was incomplete and the average recognition rate dropped to 94%. The logistic regression model outperformed the stochastic gradient descent SGD using a linear support vector machine classifier in terms of recognition rate and computation time classifier, and the recognition rate is also improved compared to the feature extracted KNN model. In addition, the fully connected neural network model has a similar recognition rate and takes less time to compute than the KNN-NFE, which has the highest recognition rate. Therefore, combining the recognition accuracy and time efficiency, fully connected neural networks still have some superiority in pose recognition.

FES control experiment designed the corresponding finite state machine control strategy for the door opening task. The experiment involved the muscle stimulation site, stimulation parameters, the transition of each stage, and the transition between each state. The experimental results

proved the effectiveness of the control strategy of finite state machine, which provided a powerful solution for the clinical design of rehabilitation plan and the implementation of rehabilitation training. The experiment of posture recognition is based on the recognition of six basic upper-limb movements. By comparing different classification methods, the practicability of fully connected neural network in posture recognition is finally determined. This discovery can be combined with the rehabilitation evaluation of patients in the later stage and can be used as a reference basis for the evaluation of patients' rehabilitation degree, which is of great significance to patients' rehabilitation.

6. Conclusions

This paper proposes a FSM controller model that supports clinical users to personalize settings according to different FES upper-limb functional tasks, which can be used as a powerful tool for clinicians to customize treatment plans for patients with different degrees of nerve injury. The implemented FSM controller was tested through the "door opening" task, and the experimental results proved its effectiveness and feasibility. The model is flexible and convenient, which greatly improves the convenience of the rehabilitation system for patients with upper-limb stroke.

In order to identify human posture, this paper starts with building a posture data acquisition platform and collects 6 of them in a three-channel data acquisition mode. The classical posture is recorded in the MEMS sensor data. Then, preprocessing such as deweighting and triple sample bar

interpolation was applied to the acquired dataset, and time domain analysis was applied from the sensor signal. Features useful for posture recognition are extracted. Subsequently, KNN, logistic regression, and random gradient descent were performed using an experimentally validated classification model with the goal of recognizing human posture experiments of the algorithms. To verify the superiority of each algorithm, the data window was adjusted to compare the recognition speed, computation duration, and accuracy of each classifier. In order to improve the accuracy of human posture recognition, a fully connected neural network-based model is established. In the process of constructing the network model, this paper investigates different activation functions and optimizers, and after experimental comparative analysis, it selects the better-performing softplus activation function as well as adagrad optimizer. Finally, by comparing the combined recognition accuracy and time efficiency with other classification models, the adjusted fully connected neural model in human is more effective and superior in posture recognition.

In this paper, based on small sample data, we establish a high-precision attitude recognition model, but there is still room for improvement, especially for the problem that the effect of small sample data in deep learning model is not as good as large-scale data. In the future, we will try to study a kind of attitude data that can generate typical attitude data by learning the characteristics of attitude data, so as to achieve the effect of expanding the sample data, and further improve in solving the problem of insufficient training samples.

Data Availability

The dataset is prepared by three-axis acceleration prototype nodes developed by ourselves for collection.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Xian Yu and Bo Xiao are the first authors due to equal contribution to this paper.

Acknowledgments

This work has received funding from Science and Technology Project of SGCC Research on Feature Recognition and Prediction of Typical Ice and Wind Disaster for Transmission Lines Based on Small Sample Machine Learning Method, National Natural Science Foundation of China (Nos. 41911530242 and 41975142), 5150 Spring Specialists (05492018012 and 05762018039), Major Program of the National Social Science Fund of China (Grant No. 17ZDA092), 333 High-Level Talent Cultivation Project of Jiangsu Province (BRA2018332), Royal Society of Edinburgh, UK and China Natural Science Foundation Council (RSE Reference: 62967_Liu_2018_2) under their Joint Interna-

tional Projects funding scheme, National Natural Science Foundation of China (Grant No. 41875184), Innovation Team of "Six Talent Peaks" in Jiangsu Province (Grant No. TD-XYDXX-004), and basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20191398 and BK20180794).

References

- [1] J. D. Pandian, S. L. Gall, M. P. Kate et al., "Prevention of stroke: a global perspective," *The Lancet*, vol. 392, no. 10154, pp. 1269–1278, 2018.
- [2] M. Ferrarin, F. Palazzo, R. Riener, and J. Quintern, "Model-based control of FES-induced single joint movements," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, no. 3, pp. 245–257, 2001.
- [3] J. J. Daly, E. B. Marsolais, L. M. Mendell, W. Z. Rymer, and A. Stefanovska, "Therapeutic neural effects of electrical stimulation," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 218–230, 1996.
- [4] G. You and T. Yan, "Functional electrical stimulation and its application in post-stroke hemiplegic patients," *Chinese Journal of Physical Medicine and Rehabilitation*, pp. 142–144, 2007.
- [5] W. Liberson, H. Holmquest, D. Scot, and M. Dow, "Functional electrotherapy: stimulation of the peroneal nerve synchronized with the swing phase of the gait of hemiplegic patients," *Archives of Physical Medicine and Rehabilitation*, vol. 42, pp. 101–105, 1961.
- [6] S. K. Sabut, P. K. Lenka, R. Kumar, and M. Mahadevappa, "Effect of functional electrical stimulation on the effort and walking speed, surface electromyography activity, and metabolic responses in stroke subjects," *Journal of Electromyography and Kinesiology*, vol. 20, no. 6, pp. 1170–1177, 2010.
- [7] S. Obuz, V. H. Duenas, R. J. Downey, J. R. Klotz, and W. E. Dixon, "Closed-loop neuromuscular electrical stimulation method provides robustness to unknown time-varying input delay in muscle dynamics," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2482–2489, 2020.
- [8] M. Lin, B. Porr, and H. Gollee, "Technical advances in functional electrical stimulation in gait function recovery," *Instrumentation Journal*, vol. 38, no. 6, pp. 1319–1334, 2017.
- [9] Q. Zhang, M. Hayashibe, and C. Azevedo-Coste, "Evoked electromyography-based closed-loop torque control in functional electrical stimulation," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2299–2307, 2013.
- [10] Y. Zhou, Y. Fang, K. Gui, K. Li, D. Zhang, and H. Liu, "sEMG bias-driven functional electrical stimulation system for upper-limb stroke rehabilitation," *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6812–6821, 2018.
- [11] A. Dodson, N. Alibejji, and N. Sharma, "Experimental demonstration of a delay compensating controller in a hybrid walking neuroprosthesis," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 465–468, Shanghai, 2017.
- [12] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines—a survey," *Proceedings of the IEEE*, vol. 84, no. 8, pp. 1090–1123, 1996.
- [13] Y. Wang, B. Metcalfe, Y. Zhao, and D. Zhang, "An assistive system for upper limb motion combining functional electrical

- stimulation and robotic exoskeleton,” *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 260–268, 2020.
- [14] S. M. Cerqueira, L. Moreira, L. Alpoim, A. Siva, and C. P. Santos, “An inertial data-based upper body posture recognition tool: a machine learning study approach,” in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 4–9, Ponta Delgada, Portugal, 2020.
- [15] Y. Xi, Y. Zhang, S. Ding, and S. Wan, “Visual question answering model based on visual relationship detection,” *Signal Processing: Image Communication*, vol. 80, article 115648, 2020.
- [16] Z. Zhang, Y. Liu, A. Li, and M. Wang, “A novel method for user-defined human posture recognition using Kinect,” in *2014 7th International Congress on Image and Signal Processing*, pp. 736–740, Dalian, China, 2014.
- [17] S. Neili, S. Gazzah, M. A. El Yacoubi, and N. E. Ben Amara, “Human posture recognition approach based on ConvNets and SVM classifier,” in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6, Fez, Morocco, 2017.
- [18] Y. Chen, L. Yu, K. Ota, and M. Dong, “Hierarchical posture representation for robust action recognition,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1115–1125, 2019.
- [19] G. Orenco, A. Lagati, and G. Saggio, “Modeling wearable bend sensor behavior for human motion capture,” *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2307–2316, 2014.
- [20] Y. Yan and Y. Ou, “Accurate fall detection by nine-axis IMU sensor,” in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 854–859, Macau, Macao, 2017.
- [21] A. Abobakr, D. Nahavandi, and J. Iskander, “RGB-D human posture analysis for ergonomie studies using deep convolutional neural network,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2885–2890, Banff, AB, Canada, 2017.
- [22] Y. Zhao, H. Li, S. Wan et al., “Knowledge-aided convolutional neural network for small organ segmentation,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1363–1373, 2019.
- [23] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, “A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning,” *Future Generation Computer Systems*, vol. 100, pp. 316–324, 2019.
- [24] Y. Xu, J. Chen, Q. Yang, and Q. Guo, “Human posture recognition and fall detection using Kinect V2 camera,” in *2019 Chinese Control Conference (CCC)*, pp. 8488–8493, Guangzhou, China, 2019.
- [25] M. B. Brahem, B. A. J. Ménélas, and M. J. D. Otis, “Use of a 3DOF accelerometer for foot tracking and gesture recognition in mobile HCI,” *Procedia Computer Science*, vol. 19, pp. 453–460, 2013.
- [26] L. A. Schwarz, A. Bigdelou, and N. Navab, “Learning gestures for customizable human-computer interaction in the operating room,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 129–136, Toronto, ON, Canada, 2011.
- [27] A. V. Dowling, J. Favre, and T. P. Andriacchi, “Inertial sensor-based feedback can reduce key risk metrics for anterior cruciate ligament injury during jump landings,” *The American Journal of Sports Medicine*, vol. 40, no. 5, pp. 1075–1083, 2012.
- [28] C. K. Lim, Z. Luo, I. M. Chen, and S. H. Yeo, “Wearable wireless sensing system for capturing human arm motion,” *Sensors and Actuators A: Physical*, vol. 166, no. 1, pp. 125–132, 2011.
- [29] B. Wang, X. Liu, B. Yu, R. Jia, and L. Huang, “Posture recognition and heading estimation based on machine learning using MEMS sensors,” in *International Conference on Artificial Intelligence for Communications and Networks*, pp. 496–508, Harbin, China, 2019.
- [30] J. X. Cai, R. X. Zhong, and J. J. Li, “Silhouettes based human action recognition in video via procrustes analysis and fisher vector odging,” *Journal of Donghua University*, vol. 2, pp. 4–9, 2019.