

Assistive Exoskeleton based on sEMG-based Motion Intention Recognition Study

Boyang Zhang*

School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, China

* Corresponding Author Email: boyangzhang@nuaa.edu.cn

Abstract. To address the issue of imprecise recognition of human motion in upper extremity assistive exoskeletons, a method for predicting the motion intention of the human upper extremity using surface electromyography (sEMG) signals is proposed, tailored for a 6-degree-of-freedom (6-DOF) upper extremity exoskeleton system. First, a three-dimensional model of the exoskeleton was built through simulation experiments, and the exoskeleton's motion equation was established. The inverse kinematics (ikine) function was utilized to compute joint variables, while the ctraj and jtraj functions were applied to generate a smooth joint trajectory from the initial position to the target position, determining the motion process of the exoskeleton through trajectory planning. In terms of signal processing, a bandpass filter was constructed to filter the sEMG signals, removing noise and irrelevant information to obtain more accurate muscle signal features. Based on this filtered data, a convolutional neural network (CNN) model was constructed to predict and recognize the filtered sEMG signals. By leveraging the CNN's multilayer convolutional architecture, deep features were extracted from the signals, enhancing the model's ability to recognize human motion intentions. The simulation results demonstrated that the CNN model achieved an average recognition accuracy of over 90% for predicting human upper limb motion intentions, validating the feasibility and effectiveness of the proposed approach. This method not only improves the accuracy of motion intention recognition for exoskeleton systems but also provides significant technical support for future applications in upper limb rehabilitation and human-machine collaboration.

Keywords: Exoskeleton, surface EMG signal, neural network, bandpass filter.

1. Introduction

With the sharp increase in the number of stroke patients in recent years, the damage to the brain motor area and hemiplegia caused by the disease lead to a large number of patients with hand function and movement disorders [1]. Studies have shown that among stroke patients with limb dysfunction, patients with hand dysfunction account for about 80% [2]. According to the discovery that the nervous system has plasticity [3], timely hand rehabilitation training can activate the plasticity of the nervous system, so that the movement disorders of patients can be further relieved [4].

In order to make patients get more effective treatment, the auxiliary exoskeleton can be used for human-computer interaction [5]. Through real-time acquisition of the patient's Surface Electromyography signal (Surface Electromyography) [6], Matlab is first used to implement band-pass filter for filtering and noise reduction of the collected signal [7], improve the signal-to-noise ratio (SNR), and enhance the information in a specific frequency range to enhance the detectability of surface myoelectric signal [8]. The Convolutional neural network model is used to predict the signals generated by the patient's palm movements, and drive the motor of the hand exoskeleton while understanding the patient's intention of the next movement [9], so as to assist the patient in the movement of daily life.

2. Working Principle of the Auxiliary Exoskeleton

The upper limb auxiliary exoskeleton is mainly composed of six degrees of freedom (6 DOF), corresponding to the interphalangeal joint, metacarpophalangeal joint, pronator teres joint, middle carpal joint, radiocarpal joint, and elbow joint of the human body. It mainly uses lightweight and high-strength materials (such as aluminum alloy and carbon fiber) to meet the required flexibility of

the exoskeleton during rotation. Rotary joints and sliding joints are used to simulate the natural joint movements of the human body, and the maximum movement limit of each joint is limited to protect the patient. A rotary motor is placed at the main rotary joint of the exoskeleton, and the motor is controlled to rotate by accepting the real-time sENG signal of the human body to predict the approximate movement trend of the arm and achieve the function of assisting the patient in daily life.

2.1. DH Parameter Establishment

The Denavit-Hartenberg (DH) parameter method can accurately describe the geometry of the exoskeleton [10]. With the use of the Link function, the relative position between each link can be defined through the link length a and the angle. Each link's properties are described by four parameters: link length a , link twist α , link offset d , and joint angle θ . By establishing six links to form the corresponding DH parameter table, we can calculate the position and orientation of the robotic arm's end-effector in matrix form. This method allows us to obtain the three-dimensional model of a six-degrees-of-freedom exoskeleton. The detailed DH parameters corresponding to the 6-degree-of-freedom joint model are shown in Table 1 and the 3D model established after simulation is shown in Figure 1. This table illustrates the DH parameters used in exoskeleton modeling.

Table 1. DH parameter table

Link(i)	$\theta_i(\text{rad})$	$d_i(\text{m})$	$a_i(\text{m})$	$\alpha_i(\text{rad})$
1	0	0.1	0.5	0
2	$-2/\Pi$	0.3	5	$-2/\Pi$
3	0	0.1	2	$-2/\Pi$
4	0	0.1	0.8	$-2/\Pi$
5	0	0.1	0.8	$2/\Pi$
6	0	0.2	0.8	0

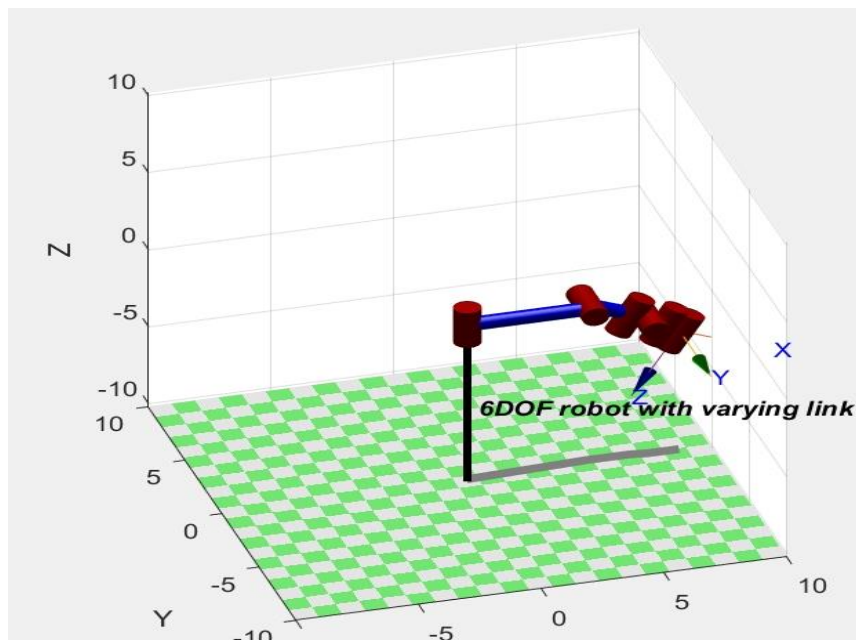


Fig. 1. Linkage model with six degrees of freedom (Photo credited: Original)

2.2. Motion Model of the Auxiliary Exoskeleton

In order to enable the exoskeleton to move from the initial matrix to the target matrix through trajectory planning, we first need to obtain the position and orientation of the end effector using the homogeneous transformation matrix in forward kinematics [11]. Suppose there is a link (i) connected to link i+1, we can use the following matrix to describe the transformation between link (i) and link (i+1):

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

Suppose there are a total of (i) links, by multiplying all the transformation matrices $A(i)$ in sequence, we can obtain the homogeneous transformation matrix T from the base to the end effector.

$$T = A_1 \cdot A_2 \cdot A_3 \quad (2)$$

2.3. Trajectory Planning

When assisting patients in performing arm movements with an exoskeleton (see Figure 2), specific actions may need to be executed. In this context, trajectory planning can be conducted using MATLAB. After determining the starting and ending points of the motion, multiple sampling points can be set to define the motion curve of the exoskeleton. First, the (ikine) function is used to compute the joint variables at each sampling point, returning a joint angle vector q , which ensures that the robotic arm's end-effector can reach the position and orientation corresponding to the target matrix. Finally, Cartesian space trajectory planning is employed, where the (ctrj) and (jtraj) functions in MATLAB's Robotics Toolbox are used to assist in generating Cartesian paths and the corresponding joint trajectories.

To enable the exoskeleton to move according to position commands, assume the initial joint vector of the link relative to the base coordinate system is $x = [0, 0, 0, 0, 0, 0]$. According to the position command, the target vector that needs to be reached is 'target' = $[0, 1, 0, \pi/3, \pi/7, \pi/5]$. By setting 200 sampling points, the jtraj function can be used to generate a smooth trajectory from the initial joint angles x to the target joint angles 'target' through polynomial interpolation.

The action instructions for the downward bending of the wrist joint of the exoskeleton are completed. Each subfigure in a, b, c, respectively shows the change curves of the angular displacement, angular velocity and angular acceleration of each joint with time during the trajectory planning process (see Figure 3).

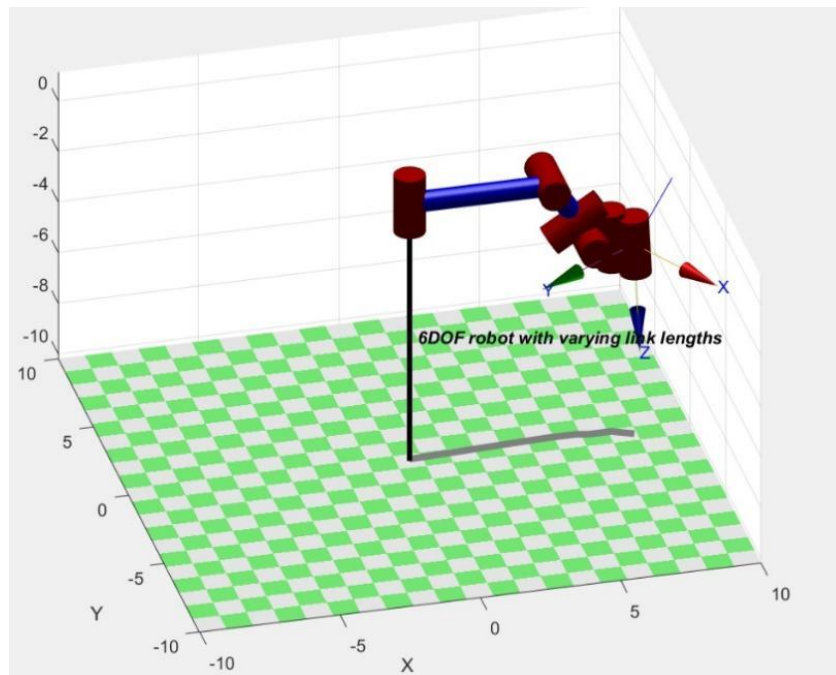


Fig. 2. Trajectory planning diagram of the auxiliary exoskeleton (Photo credited: Original)

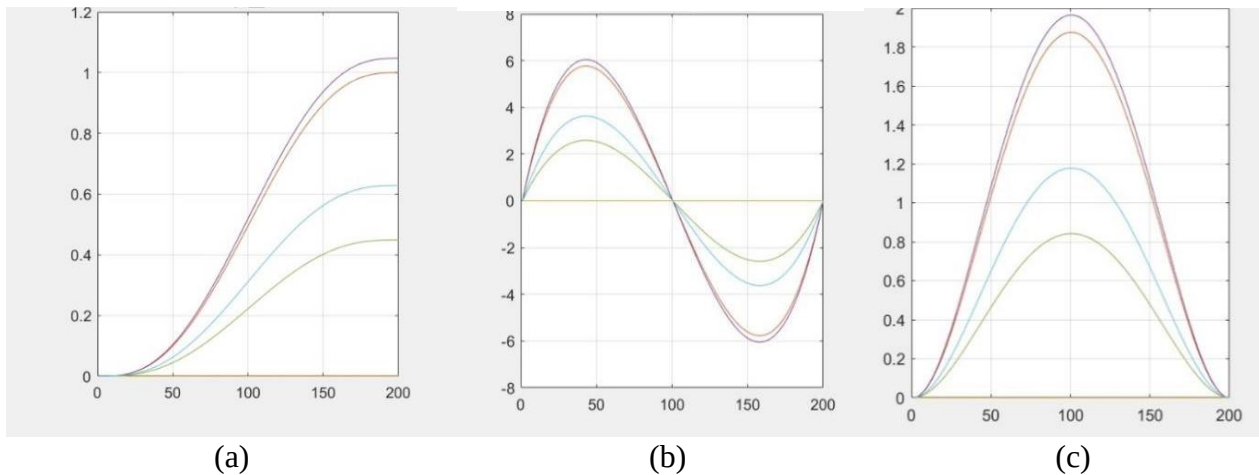


Fig. 3. The total time of exoskeleton movement from the initial position to the target position is divided into 200 sample points. (a), (b) and (c) depict the angular displacement, angular velocity and angular acceleration of the exoskeleton model corresponding to each sample point (Photo credited: Original)

It can be observed that the motion of each joint is smooth and without special fluctuations, the motion trajectory has continuity, and the velocity and acceleration of the joint transition smoothly to zero at the start and end of the trajectory. It can be found that there are no sudden fluctuations in the changes of angular velocity and acceleration, which indicates that the manipulator is stable and reliable during operation, and there is no jitter phenomenon. This smooth movement ensures that the exoskeleton does not cause unnecessary impact or injury to the patient's arm while assisting in daily life, thus providing a safe and effective rehabilitation environment.

3. Surface EMG Data Acquisition

In this study, patient data collected from the Ninapro (<http://ninapro.hevs.ch/>) open source dataset, which currently contains DB [1-10] completed datasets containing sEMG signal files of amputees and non-amputees, was selected as the study objects. Among them, the NinaproDB1 dataset used in this study included surface EMG and kinematic data of 27 complete subjects during 52 hand movements and rest positions.

sEMG data was obtained by 10 Otto Bock MyoBock 13E200 electrodes (see Figure 4), while kinematic data was obtained using Cyberglove 2 data glove.

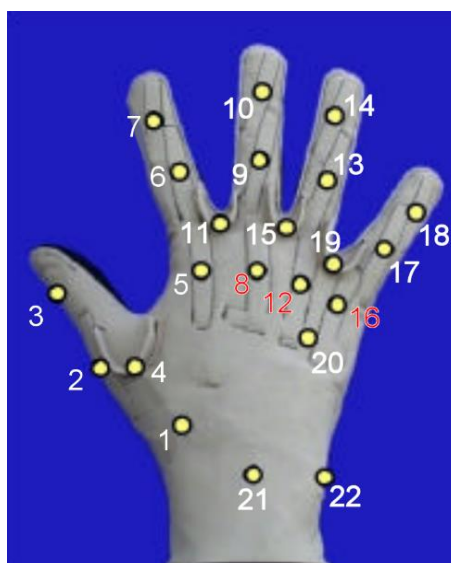


Fig. 4. Cyberglove 2 data glove [12]

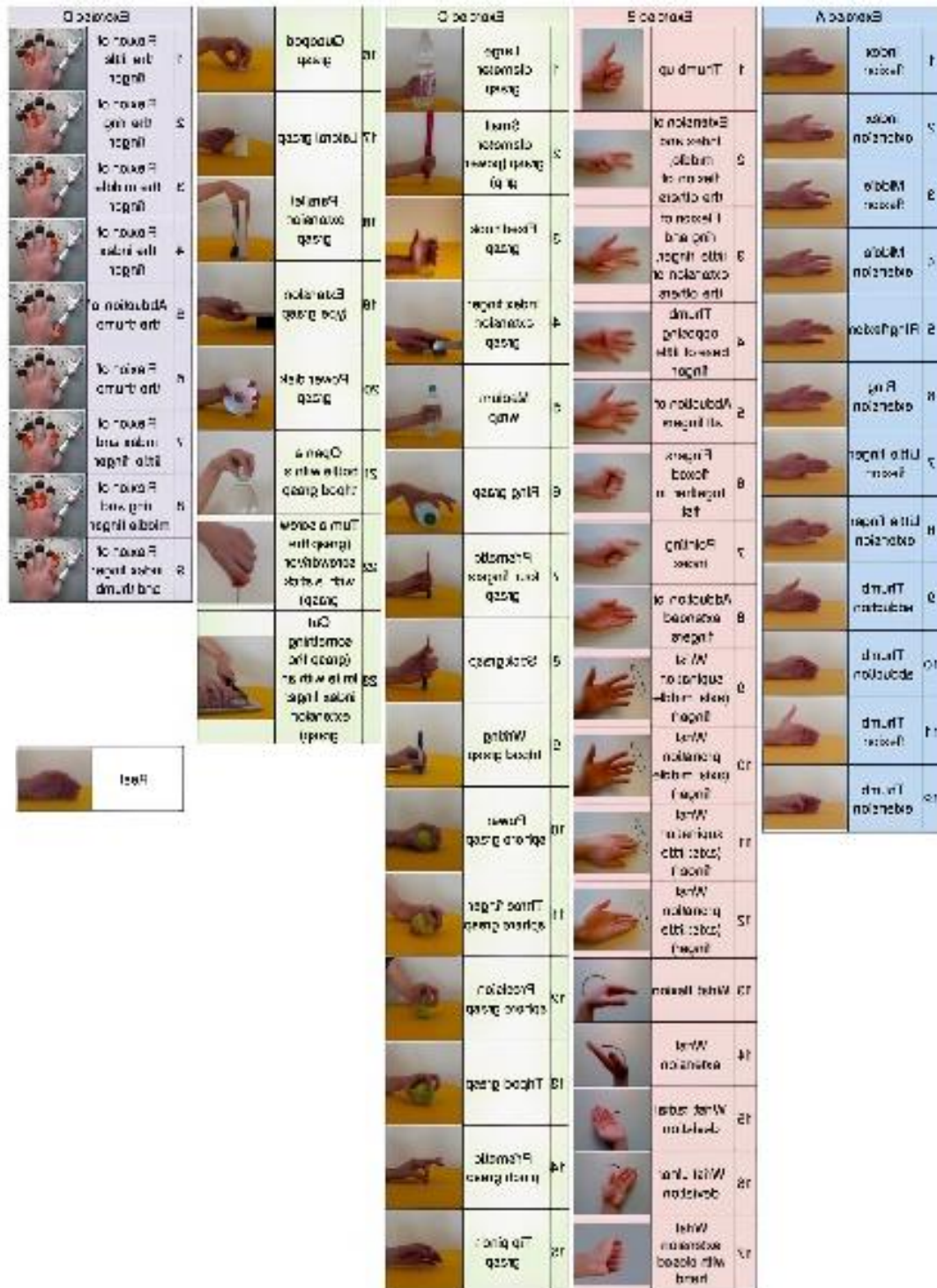


Fig. 5. The whole action [12]

The DB1 dataset contains three sets of exerciseA-C movements. In this paper, the 13th sEMG data from exerciseB about wrist downward bending (see Figure 5) was mainly captured as the main research object in the filtering and neural network models.

4. sEMG Signal Preprocessing

Since the collected sEMG signal is susceptible to noise interference from external electronic instruments, direct use of unprocessed signal data may lead to errors in experimental results, so it is necessary to pre-process the signal before motion intention recognition, the entire procedure includes normalization processing, signal filtering and denoising.

4.1. Normalized Processing

sEMG signals are characterized by multiple features and may have different levels of data. Direct reference of real-time data acquisition may lead to the problem that features with larger values have too much influence weight on the model, while features with smaller values may be ignored, and gradient disappearance or gradient explosion may occur in the subsequent training process of neural networks. Therefore, it is necessary to normalize the collected data. In this study, Min-max normalization model is used to solve this problem.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3)$$

The x' is the normalized value and x is the original value. What's more, $\min(x)$ and $\max(x)$ are the maximum and minimum values in the original data respectively, by bringing the original data into the formula, the original data can be distributed in the interval of $[0, 1]$, where the minimum value is mapped to 0, and the maximum value is mapped to 1. With Min-Max Scaling, all the data collected will be compressed to the same magnitude so that each eigenvalue has the same effect on the weight of the model.

4.2. Bandpass Filter

In order to filter and de-noise the signal, the low-pass filter (LPF) and high-pass filter (HPF) can be connected to form a bandpass filter (BPF) to filter the sEMG signal as shown in the Figure 6. The center frequency is set to 100hz and the passband bandwidth is set to 50hz, so that the signal in the range of $[50, 150]$ is retained as much as possible, and the frequency signal outside the range is attenuated.

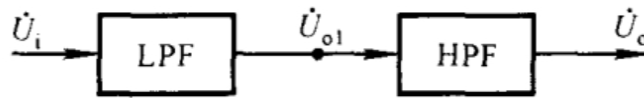


Fig. 6. Bandpass Filter [7]

$$B(u, v) = L(u, v) \cdot H(u, v) \quad (4)$$

In order to obtain the transfer function of the bandpass filter, we can first set up the ideal low-pass filter transfer function, where s_0 is only related to the cutoff frequency.

$$H_{LP}(s) = \frac{1}{\frac{s}{s_0} + 1} \quad (5)$$

Similarly, the transfer function of an ideal high-pass filter can be expressed as:

$$H_{HP}(s) = \frac{\frac{s}{s_1}}{\frac{s}{s_1} + 1} \quad (6)$$

Then, according to the formula (0), the formula in (1) and (2) can be brought into it:

$$H_{BP}(s) = \left(\frac{\frac{s}{s_1}}{\frac{s}{s_1} + 1} \right) \cdot \left(\frac{1}{\frac{s}{s_0} + 1} \right) \quad (7)$$

Assuming that the cut-off frequencies of the low-pass and high-pass filters are f_0 and f_1 respectively, the transfer function of the bandpass filter can be obtained according to the Laplace transform ($s=j\omega$) :

$$H_{BP} = \frac{\frac{s}{2\pi \int_1} \cdot \frac{1}{\frac{s}{2\pi \int_2}}}{\frac{s}{2\pi \int_1} + 1} \quad (8)$$

In this study, sEMG signals of the wrist joint flexion were collected from four subjects. At each time point, seven groups of feature points (signal1-7) were selected, and the signals were denoised through filtering. A band-pass filter built in MATLAB was used to perform the filtering and denoising. The time-domain waveforms of the signals before and after filtering and denoising are shown in Figure 7 and 8.

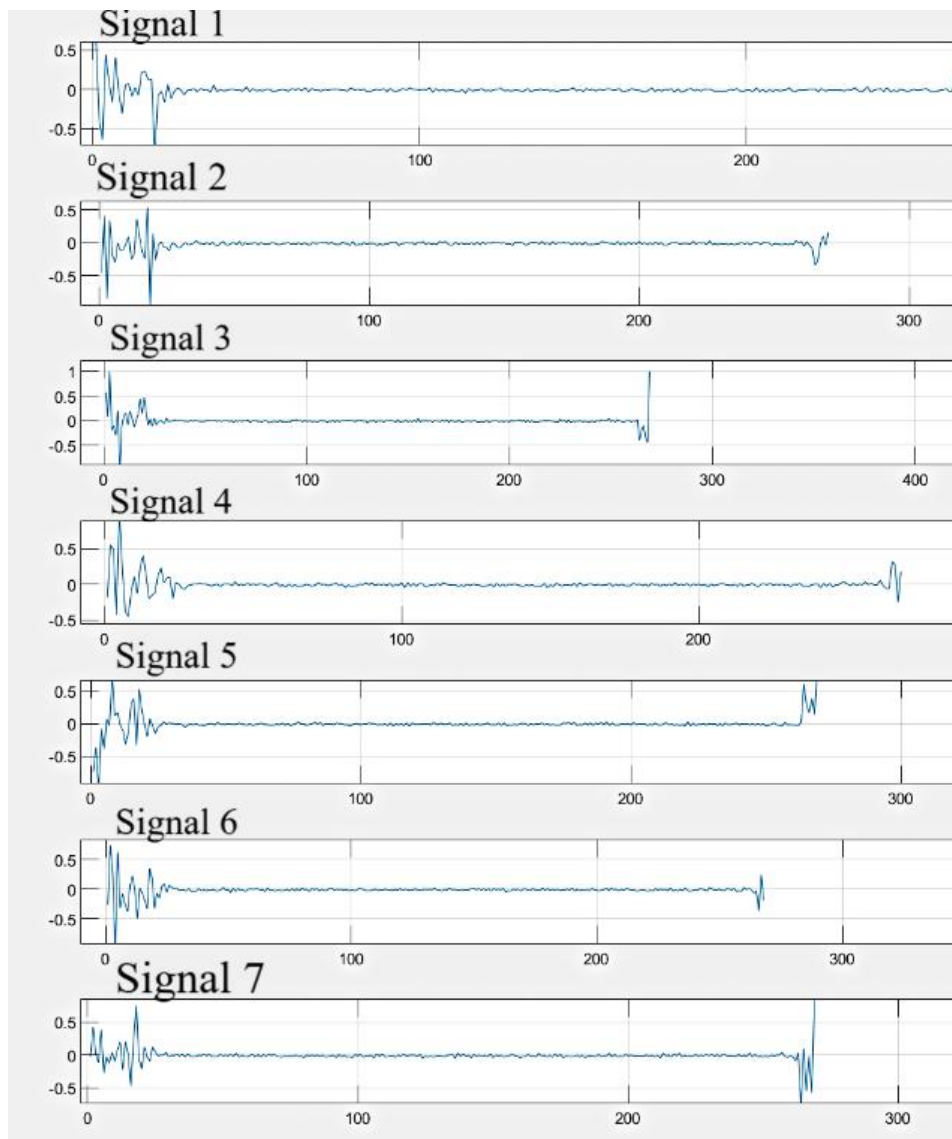


Fig. 7. Time domain waveform before filtering (Photo credited: Original)

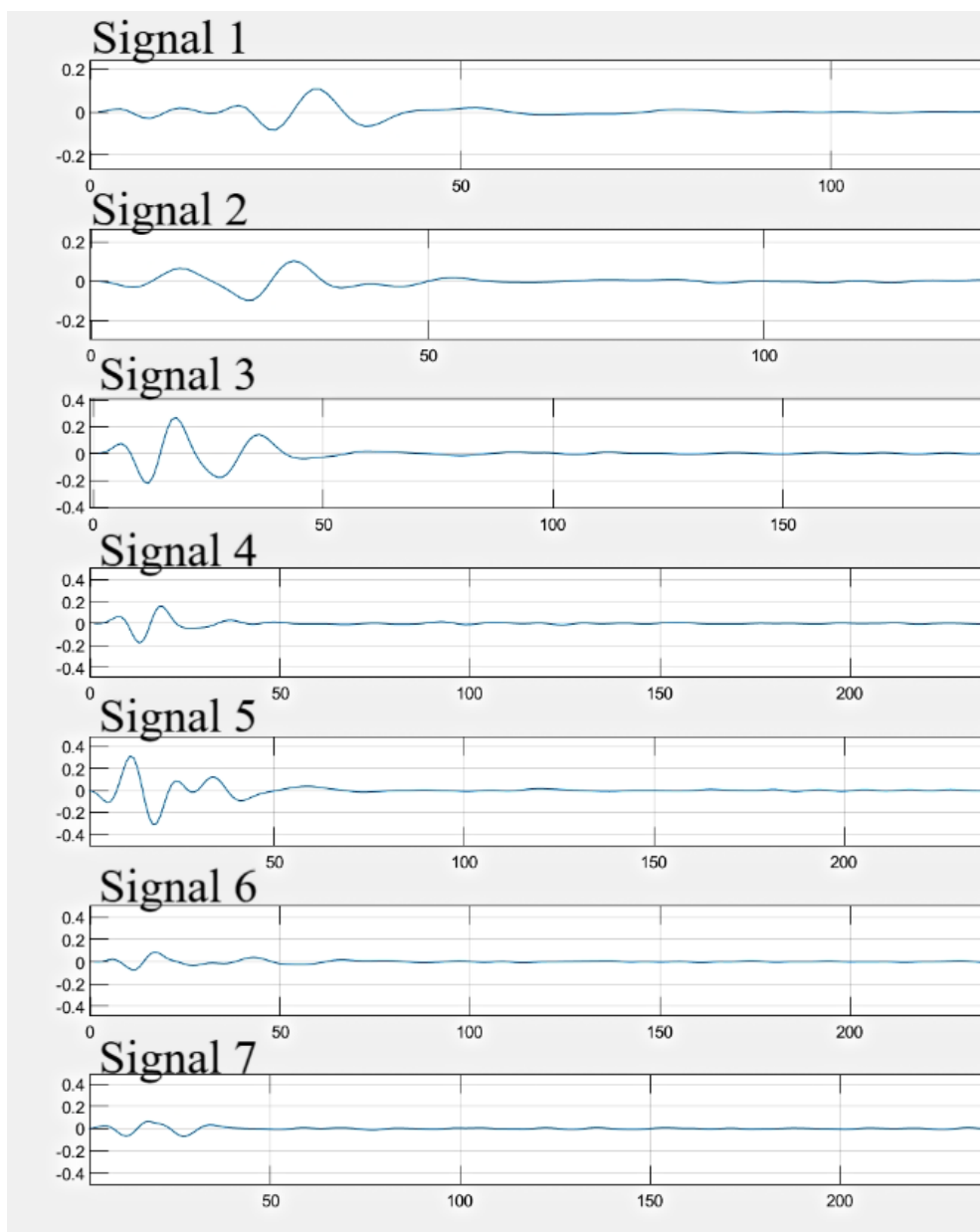


Fig. 8. Time domain waveform after filtering (Photo credited: Original)

From the comparison of the above figure, it can be clearly found that the time domain waveform becomes clean and clear under the filter of the bandpass filter, which greatly reduces the inherent noise generated by the electronic components in the myoelectric equipment and the impact caused by other environmental noises, and the signal-to-noise ratio is improved.

5. Convolutional Neural Networks

To achieve motion intent recognition, this study primarily uses a convolutional neural network (CNN) to predict the filtered and denoised sEMG signals. CNNs have the advantage of autonomously learning data features by introducing convolution operations and hierarchical feature extraction methods. The network mainly consists of convolutional layers, pooling layers, and fully connected layers, with the structure shown in Figure 9. The convolutional neural network begins by using convolutional layers to perform local dot product operations on the data to generate feature maps. These feature maps are then downsampled and reduced in computational complexity through pooling layers. Finally, the fully connected layers at the end of the network map the high-level features learned by the CNN to the predicted values, thus achieving motion intent recognition.

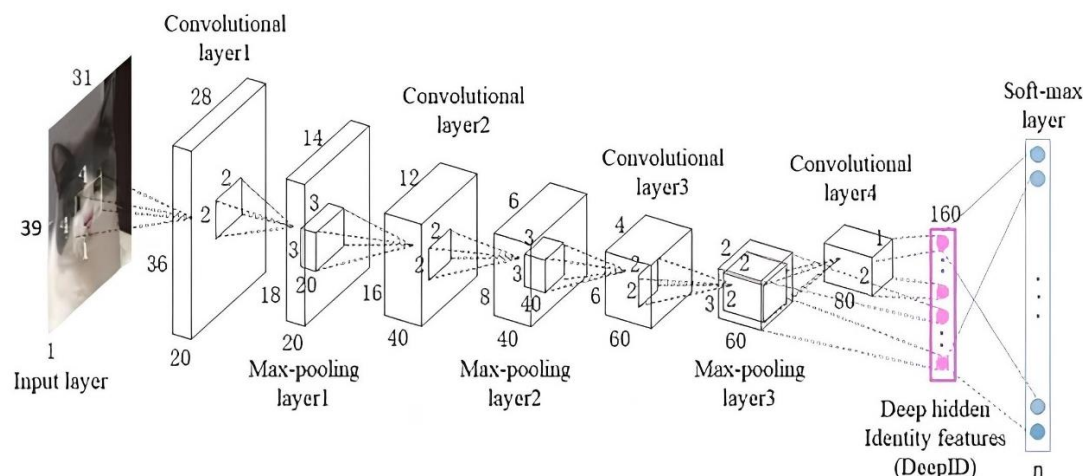


Fig. 9. The structure of CNN [9]

5.1. Construct Neural Network Prediction Model

First, the wrist motion data extracted from the Ninapro DB1 dataset was filtered. Then, 500 sample points were selected from the filtered data, with each sample containing seven feature values and one target value. 80% of the samples were used as the training set, and the remaining 20% as the test set. To ensure consistency between the structure of the CNN input layer and the data, each sample was converted into a tensor format of $7 \times 1 \times 1$.

For the convolutional layer, a 3×1 kernel was used. Since CNNs are typically applied to image data, the sEMG signals were segmented into fixed windows, each containing seven time points and multiple corresponding feature values. In the code, this was represented in the form of $[7, 1, \text{channel}]$. The first convolutional layer outputs 16 feature maps, while the second layer outputs 32 feature maps. The data is then passed through a batch normalization layer for normalization and subsequently into an activation layer. A ReLU activation layer ('reluLayer') was implemented in MATLAB to enable the CNN to handle non-linear problems. A dropout layer was added with a retention rate of 0.8 to prevent overfitting. In the fully connected layer, the output unit was set to 1 to ensure that only one value is produced.

The model was trained using Stochastic Gradient Descent with Momentum (SGDM), with a maximum of 800 iterations, an initial learning rate of 0.01, and a learning rate decay strategy. The performance of the model was evaluated using Root Mean Square Error (RMSE), R-squared (R^2), Mean Absolute Error (MAE), and Mean Bias Error (MBE). The approximate structure of each layer in the model is shown in Table 2.

Table 2. The approximate structure of each layer in the model.

	Name	Type
1	Imageinput 7×1×1 image: 'zerocenter' normalization	Image Input
2	conv_1 16 3×1 convolution: stride [1 1]	2D Convolution
3	batchnorm_1	Batch Normalization
4	relu_1	ReLU
5	conv_2 32 3×1 convolution: stride [1 1]	2D Convolution
6	batchnorm_2	Batch Normalization
7	relu_2	ReLU
8	Dropout 20% Dropout	Dropout
9	fc 1 Fully Connected Layer	Fully Connected
10	regressionoutput	Regression Output

5.2. Simulation experiment and comparison of results

As shown in Figure 10 in the early stage of training, RMSE is high and the training error is large, which indicates that the model is still in the data learning stage. With the increase of iterations, the RMSE value tends to be stable and is around 0.1-0.2, which indicates that the model has basically converged and the training of the model has achieved good results. The change trend of training loss and RMSE is similar, both tend to be stable, which shows the effectiveness and convergence of the training process.

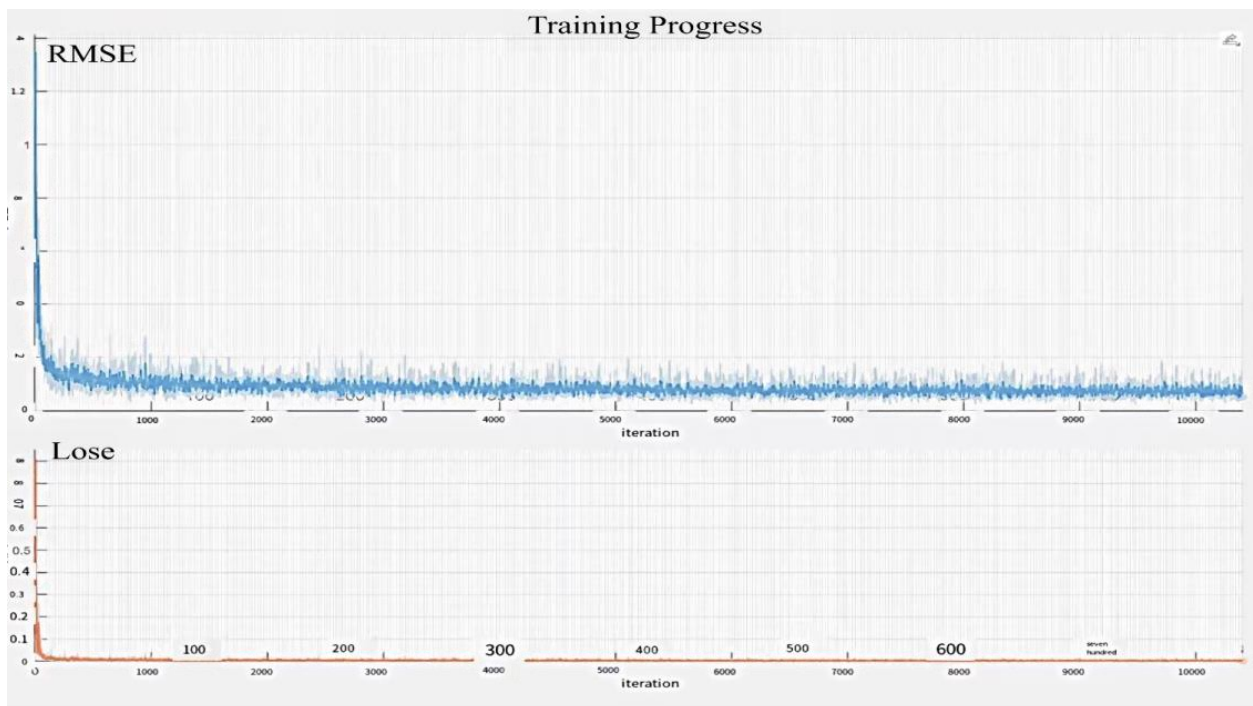


Fig. 10 RMES and Lose change in the training process (Photo credited: Original)

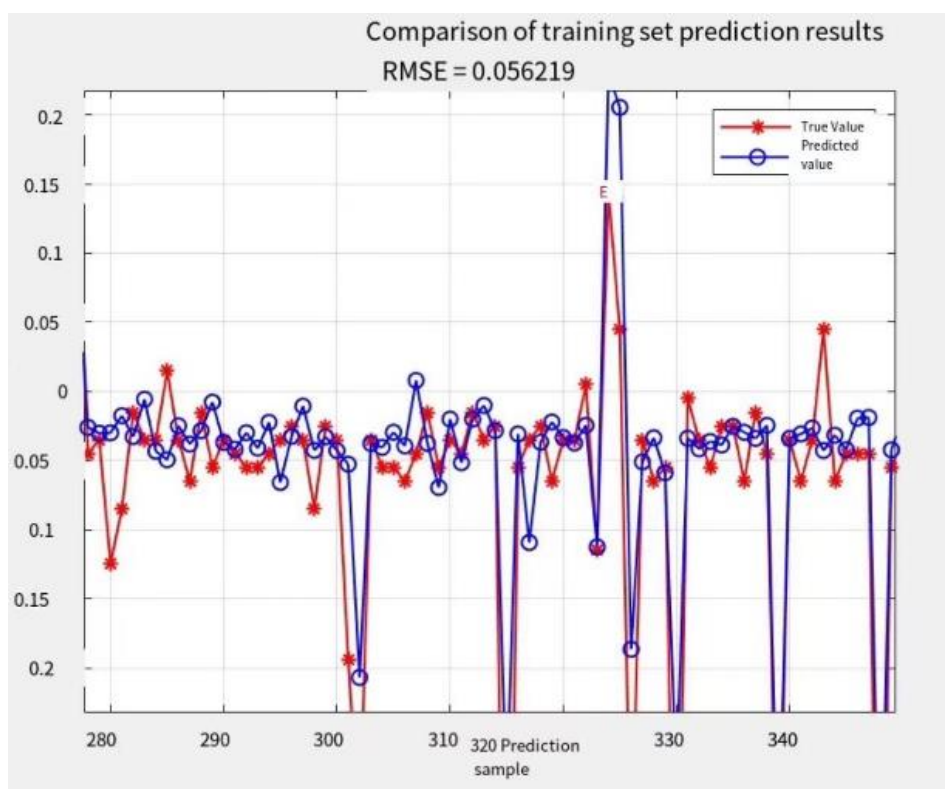


Fig. 11. Comparison of training set prediction results (Photo credited: Original)

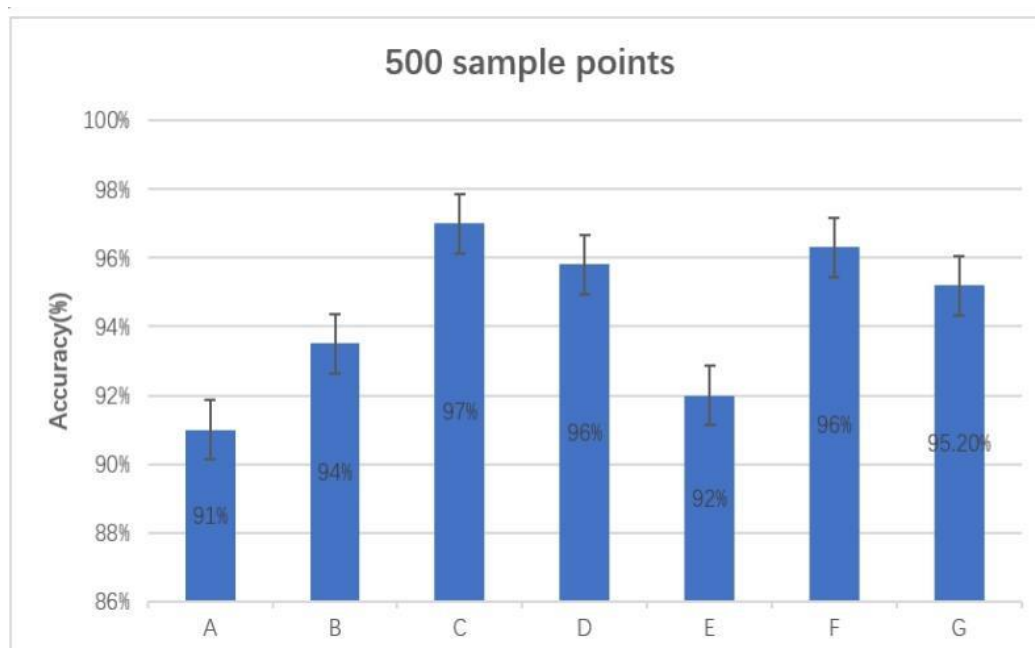


Fig. 12. Prediction accuracy of all patients (Photo credited: Original)

Figure 11 captured the comparison between the real value and the predicted value in part of the predicted samples. The prediction accuracy rate in this group of data reached 94.5%, and according to the observation on the figure, the real value and the predicted value fit well. At the same time, sEMG signal predictions for the same movement were performed on seven patients. As shown in Figure 12, the prediction accuracy for all patients reached 90%, with the highest accuracy at 97% and the lowest at 91%. This demonstrates that the CNN can accurately predict sEMG signals and effectively achieve the intended motion recognition.

6. Conclusion

The importance of an assistive exoskeleton for patients with upper limb mobility impairments was analyzed. A six-degree-of-freedom 3D model of the exoskeleton was established based on the DH parameter method. By utilizing trajectory planning, a Cartesian path and corresponding joint path were generated, enabling the exoskeleton to move along a reasonable trajectory to the target position, assisting patients in their daily activities while conforming to the characteristics of human-machine interaction. The raw data from the sEMG database was normalized, compressing the data to the same scale. A band-pass filter was used to denoise the normalized data, improving the signal-to-noise ratio of the filtered signals. Finally, a convolutional neural network model was designed to predict the filtered signals. The prediction results, including comparison graphs and RMSE values, were obtained. The sEMG signals of multiple patients in the DB1 dataset were predicted, ensuring that the prediction accuracy exceeded 90%, thus potentially aiding patients in their daily activities.

References

- [1] Chen S. and Jia J. and Shu X. EEG evaluation of stroke patients with hand dysfunction. *Annals of Physical and Rehabilitation Medicine*. 2018, 61: e446-e446.
- [2] Jiang Rongrong, Chen Yan, Pan Cuihuan. Research Progress on the Rehabilitation Assessment of Upper Limb and Hand Motor Function After Stroke. *Chinese Journal of Rehabilitation Theory and Practice*, 2015, 21(10): 1173-1177.
- [3] Stolbkov Y. K. and Yu. P. Gerasimenko. Neurorehabilitation Based on Spinal Cord Stimulation and Motor Training". *Neuroscience and Behavioral Physiology*. 2024: 1-12.
- [4] Zeiler SR, Krakauer JW. "The interaction between training and plasticity in the poststroke brain." *Curr Opin Neurol*, 2013; 26(6): 609-16.

- [5] Peng Liang, Hou Zengguang, Wang Chen, et al. Rehabilitation Assistive Robots and Their Physical Human-Machine Interaction Methods. *Acta Automatica Sinica*, 2018, 44(11): 2000-2010.
- [6] Ergeneci Mert et al. An Embedded, Eight Channel, Noise Canceling, Wireless, Wearable sEMG Data Acquisition System With Adaptive Muscle Contraction Detection." *IEEE transactions on biomedical circuits and systems*. 2018. 12(1): 68-79.
- [7] Yi Le et al. A high-selective multiple-mode bandpass filter design, applicating in both millimeter-wave 5G and WiFi systems. *Microelectronics Journal*. 2024, 149: 106250.
- [8] Yan Huixin, Yang Lu, Li Shaocong, et al. "Design and Simulation of Digital Bandpass Filter Based on FPGA." *Modern Electronics Technique*, 2024, 47(02): 7-10.
- [9] Zeng Ziniu. "Research on Motion Intention Prediction Method Based on Surface Electromyography and Upper Limb Rehabilitation Exoskeleton Control." *South China University of Technology*, 2022.
- [10] Dabao Lao et al. "Error Modeling and Parameter Calibration Method for Industrial Robots Based on 6-DOF Position and Orientation." *Applied Sciences*. 2023, 13:19.
- [11] Niu Yuanhui, Cheng Guangming, and Yang Zhigang. "Forward Kinematics Analysis of Manipulators Under the D-H Coordinate System." *Mechanical Engineer*, 2006, 7: 3.
- [12] Work partially supported by the Swiss National Science Foundation ([www. snf.ch](http://www.snf.ch)) through the Sinergia projects #132700 NinaPro and #160837 MeganePro.