

Torque Prediction of Ankle Joint from Surface Electromyographic Using Recurrent Cerebellar Model Neural Network

Hai-Yan Jiang^{1,2}, Shou-Yan Yu¹, Chih-Min Lin^{3,*}, Yan Chen^{1,2},
Shu-Ping Huang^{1,2}

¹ College of Electrical Engineering and Automation, Fuzhou University, Fuzhou, 350108, Fujian, China, jianghaiyan@fzu.edu.cn, n170127071@fzu.edu.cn, n180127014@fzu.edu.cn, n190127024@fzu.edu.cn

² Fujian Key Lab of Medical Institute and Pharmaceutical Technology, Fuzhou University, Fuzhou, 350108, Fujian, China

³Department of Electrical Engineering, Yuan Ze University, Tao-Yuan 320, Taiwan, cml@saturn.yzu.edu.tw

*Corresponding author

Abstract: Joint torque prediction plays an important role in quantitative limb rehabilitation training and the exoskeleton robot. The Surface electromyography signal (sEMG) with the advantages of non-invasive and easy collection can be applied to the prediction of human muscle force. By utilizing the sEMG, the recurrent cerebellar model neural network (RCMNN), which has better generalization and computational power than the traditional neural network has been used to predict the joint torque. In this work, a smooth function with adaptive coefficient is employed to polish the results of RCMNN, the proposed method shows great performance on torque prediction with the correlation coefficient between the torque and the estimation result up to 98.43%, such advanced model paves the way to the application on the quantitative rehabilitation training.

Keywords: Torque prediction; ankle joint; sEMG; RCMNN

1 Introduction

Joint torque prediction plays an important role in quantitative limb rehabilitation training and exoskeleton robots [1]. A number of kinematics and non-invasive methods have been proposed to estimate the joint torque, but most of these require special measuring apparatus, such as isokinetic dynamometers, which made these methods unsuitable for application in patients' daily lives and operations outside the lab. The Surface electromyographic (sEMG) signals are a kind of biological

electrical signals recorded on the skin surface, containing movement intentions of the human body [2]. Generally, compared with electromyographic (EMG) signals collected by needle electrodes, sEMG signals are widely used in rehabilitation medicine and sports medicine [3-6] because of their advantages in terms of simple operation, non-invasive and multi-point measurement. Moreover, due to the transmission time, sEMG signal is generated 30-150 ms earlier than human muscle movements, which makes it a valuable signal source for man-machine interaction technology [7]. Therefore, it is of significance to use the sEMG signals in the joint torque prediction.

Several nonlinear models have been proposed for estimation of ankle joints torque from the sEMG signals, such as artificial neuron network (ANN) and fuzzy model (FCM) [8-13], which actually obtained good results in various nonlinear modeling. Kim *et al.* [14] using deep neural network to estimate the torque by the sEMG signal. Lu *et al.* [15] developed an sEMG-based torque estimation control strategy for a soft elbow exoskeleton. Xu *et al.* [16] proposed an sEMG-based joint torque estimation strategy combining with hill-type muscle model by using radical basis function neural network to make the results of muscular movement digitized. Currently, artificial neuron network (ANN) as a nonlinear model is still the most popular model used in torque prediction [17-19]. However, the learning of the neural network is slow since all the weights are updated during each learning cycle. The cerebellar model neural network (CMNN) has been widely used in high-precision control fields due to its simple structure, good generalization, rapid learning speed and good convergence [20-26]. A recurrent unit, which considers the effect of the previous moment on the current moment, is added to the CMNN to make it a recurrent CMNN (RCMNN). Since the prediction problem is relative to time sequence, the prediction performance of RCMNN is better than CMNN. Therefore, utilizing the sEMG and RCMNN to predict joint torque is an essential project in a wide range of real-time applications.

In this paper, an sEMG-driven RCMNN is introduced to predict the torque of the ankle joint. A group of processed signals consisting of the joint angular velocity, accelerometer signals, and sEMG signal related to the joint torque are sent to the predictor and then the prediction output result is obtained. At last, a smoothing function with the adaptive coefficient is used to process the output result of the RCMNN to obtain the final torque prediction result.

2 Data Acquisition

To estimate the ankle joint torque, Delsys Trigno Wireless System was used for collecting sEMG signals, and BIODEX System 4 Pro Strength Testing System was applied for collecting angular velocity, accelerometer, and torque signals

(Fig. 1(a)). Five Trigno sEMG sensors were placed on the subject muscles, including gastrocnemius, tibialis anterior, peroneus Longus, extensor hallucis longus, and extensor digitorum longus, for detecting the sEMG signals on the disinfected skin surface. Then the subjects were fixed on BIODEX System 4 Pro Strength Testing System to swing ankle joint eversion and inversion as described in Fig. 1 (b). The raw data from the sEMG and position sensors are sampled at rates of 2000 Hz and 418 Hz using a 16-bit A/D converter, respectively. Note that all the subjects, ranging in age from 17-42 years participated in this study, are in ankle health, and none of them has a history of injuries.



(a)

(b)

Figure 1

The system setup for the experiment

After data acquisition, a serial of raw sEMG data were collected and shown in Fig. 2.

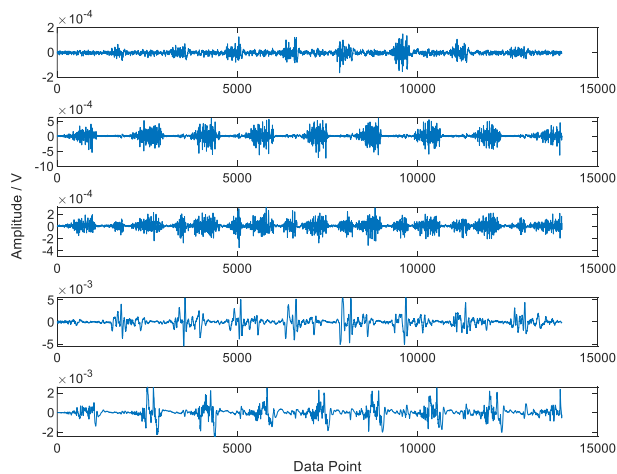


Figure 2

The raw sEMG signals

In Fig. 2, from top to bottom are the amplitudes of gastrocnemius, tibialis anterior, longus tibialis, extensor pollicis, extensor digitalis longus.

3 Data Processing

3.1 Outlier Processing and De-noising

Due to the undesirable noise and disturbances, the raw sEMG signals are hardly estimated the ankle joint torque. As known, the surface muscle telecommunications energy is mainly concentrated between the frequency bands of 20 Hz and 500 Hz. For this reason, the raw signal can be filtered by removing the uncorrelated frequency band and the 50 Hz power frequency. In this study, the signal is processed by Fourier transform using a filter with the amplitude of the incoherent frequency at zero. Then, the inverse transform is performed to obtain the processed data of tibialis anterior shown in Fig. 3.

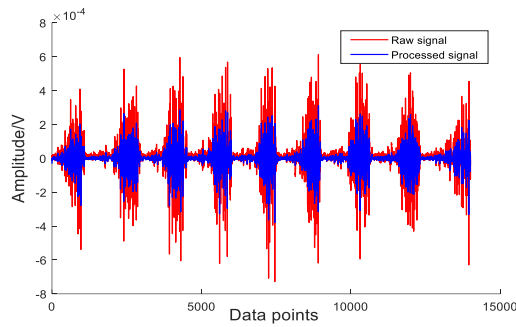


Figure 3
Contrast between raw signal and denoised signal

It can be seen that the processed signal effectively restrains the noise and preserves the peaks and the abrupt part so that it can retain the characteristics of the raw signal.

3.2 De-redundancy

However, redundancy may exist between the sEMG signals of five muscles [27], which will limit the prediction accuracy of the neural network and increase the calculation amount and the prediction time of the model.

In this section, a correlation coefficient is employed to determine the degree of coupling among different sEMG signals of muscles. The formula is described as follows

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{n=1}^N (f_i(n) - \bar{f}_i)^2}, n = 1, 2, \dots, N \quad (1)$$

$$C_{ij} = \frac{1}{N} \sum_{n=1}^N (f_i(n) - \bar{f}_i)(f_j(n) - \bar{f}_j), n = 1, 2, \dots, N \quad (2)$$

$$\rho_{ij} = \frac{C_{ij}}{\sigma_i \cdot \sigma_j} \quad (3)$$

where σ_i is the standard deviation of the sEMG signal on the i -th muscle surface, and C_{ij} and ρ_{ij} are the covariance and correlation coefficients of the sEMG signals on the i -th and j -th muscle surfaces.

Finally, the correlation coefficient of the five muscles is shown in the following table.

Table 1
Correlation coefficient between muscles

Muscle 1 \ Muscle 2	Gas	T-A	L-T	E-P	E-D-L
Gas	1	0.0277	-0.0868	0.2175	0.0589
T-A	0.0277	1	-0.4539	-0.0684	0.1451
L-T	-0.0868	-0.4539	1	0.0140	-0.1261
E-P	0.2175	-0.0684	0.0140	1	-0.0898
E-D-L	0.0589	0.1451	-0.1261	-0.0898	1

P.S. Gas, T-A, L-T, E-P, and E-D-L represent gastrocnemius, tibialis anterior, Longus tibialis, extensor pollicis, extensor digitalis longus respectively.

Furthermore, to determine whether there is redundancy between the data, a method shown as Fig. 4 is used.

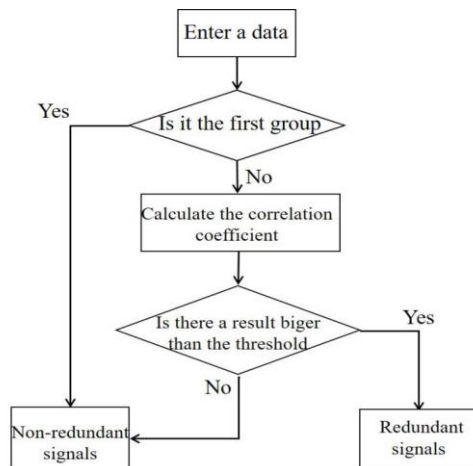


Figure 4
Redundancy decision flowchart

In this flowchart, all the data is divided into redundant signals and non-redundant signals. The first group of data, the gastrocnemius signal, is put into non-redundant information and a threshold (absolute value is 0.4) is set. Then, the second group data is input, and calculate the correlation coefficient between group 2 and non-redundant information. The correlation coefficient of the data is compared with the threshold value. When the correlation coefficients are bigger than the threshold value, the data is considered as redundant data, otherwise for the non-redundant data. Repeat the second step until all the data is judged.

According to Table 1 and Fig. 4, the signals of the tibialis anterior and longus tibialis are redundant.

3.3 Resampling

Because the sampling frequency of sEMG (2,000 Hz) is higher than that of joint angle and joint force. In order to match the length of the sEMG signal the in time-domain, the joint torque data are also re-sampled and are then smoothed. To test the effect of redundancy, in this paper, the data set is divided into a redundant group and a non-redundant group. The redundant data set includes processed 5 channels of sEMG, joint angular velocity and accelerometer signals, and the non-redundant data group includes 4 channels of sEMG except gastrocnemius, as well as joint angular velocity and accelerometer signals. The redundant group and non-redundant group data after resampling are 7-dimensional and 6-dimensional feature vectors respectively, and they will be inputted into the RCMNN.

4 Review of RCMNN

4.1 Structure of RCMNN

The schematics of the RCMNN are shown in Fig. 5 with the input space, the association memory space, the receptive-field space, the weight memory space, and the output space [28].

Input Space For a given input data I_i , $i=1,2,\dots,m$, each input variable I_i is quantized into n discrete regions (called “elements” or “neurons”) according to given control space. The value of neurons n is called resolution. Based on this n element, n layers are defined in total.

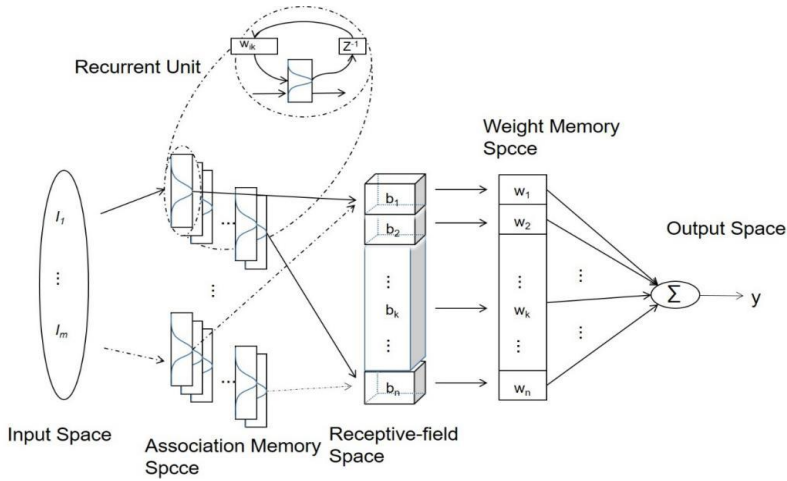


Figure 5
Architecture of a RCMNN [28]

Association Memory Space In the space, the Gauss function is used as the receptive-field basis function, which can be expressed as

$$r_{ik} = \exp\left[-\frac{(I_{ik} - m_{ik})^2}{\sigma_{ik}^2}\right], \quad i = 1, 2, \dots, m, \quad k = 1, 2, \dots, n \quad (6)$$

where r_{ik} represents the output of the k -th receptive-field basis function for the i -th input, and the mean m_{ik} and variance σ_{ik} represent the center point and width of the Gaussian function. In addition, I_{ik} is the result of the recurrent unit and can be represented by

$$I_{ik}(t) = I_i(t) + w_{ik} r_{ik}(t-1) \quad (7)$$

where t denotes the time step, and $r_{ik}(t-1)$ is the value of r_{ik} through a time delay; w_{ik} is the recurrent gain, and the size of the value represents impact of the information of one previous point to the current moment.

To further understand w_{ik} here, a linear function instead of the Gaussian function is chosen as basis function and is defined as

$$r_{ik}(t) = I_{ik}(t) \quad (8)$$

Then, (7) can be expressed as

$$r_{ik}(t) = I_i(t) + w_{ik} r_{ik}(t-1) \quad (9)$$

Since

$$r_{ik}(t-1) = I_i(t-1) + w_{ik} r_{ik}(t-2) \quad (10)$$

Equation(9) becomes

$$\begin{aligned} r_{ik}(t) &= I_i(t) + w_{ik} r_{ik}(t-1) \\ &= I_i(t) + w_{ik} I_i(t-1) + w_{ik}^2 r_{ik}(t-2) \\ &= I_i(t) + w_{ik} I_i(t-1) + \dots + w_{ik}^{n-1} I_i(t-n+1) + w_{ik}^n r_{ik}(t-n). \end{aligned} \quad (11)$$

Obviously, the output of RCMNN in the associative memory space contains the information of all previous parameters.

Receptive-Field Space The k -th multidimensional receptive-field function is defined as

$$b_k = \prod_{i=1}^m r_{ik}, \quad k = 1, 2, \dots, n, \quad i = 1, 2, \dots, m \quad (12)$$

The output of the space indicates the excitation intensity for the last space.

Weight Memory Space The weight of the receptive-field space with the output space is stored in the space and denoted in a vector as

$$w = [w_1, \dots, w_k, \dots, w_n]^T \quad (13)$$

Output Space The output of RCMNN is the algebraic sum of the Receptive-Field Space in the weight memory space and is shown as

$$y = \sum_{k=1}^n (w_k \cdot b_k) \quad (14)$$

4.2 Learning Algorithm of RCMNN

In this section, backpropagation (BP) is applied to update the parameters m_{ik} , σ_{ik} , w_{ik} and w of RCMNN. While BP is the iterative gradient decent algorithm designed to minimize the mean square error of an objective function defined as

$$E = \frac{1}{2}(y_0 - y)^2 = \frac{1}{2}e^2 \quad (15)$$

where $e = y_0 - y$, y_0 is the previous known value for the testing data and the expected output of neural network.

The parameter updating rule based on the gradient descent algorithm is derived according to

$$P(N+1) = P(N) + \Delta P = P(N) + \eta_p \frac{-\partial E(N)}{\partial P} \quad (16)$$

where η_p is a learning rate, and if P is replaced by m_{ik} , σ_{ik} , w_{ik} and w then it denotes the updating rule for mean, variance, recurrent gain, and output weight, respectively.

The changed values of the above parameters are calculated by the chain rule represented as

$$\begin{aligned} \Delta w_k &= -\eta_w \frac{\partial E}{\partial w_k} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial w_k} \\ &= \eta_w (y_0 - y) \cdot b_k \end{aligned} \quad (17)$$

$$\begin{aligned} \Delta m_{ik} &= -\eta_m \frac{\partial E}{\partial m_{ik}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial b_k} \frac{\partial b_k}{\partial r_{ik}} \frac{\partial r_{ik}}{\partial m_{ik}} \\ &= \eta_m (y_0 - y) \cdot w_k \cdot b_k \cdot \frac{2(I_{ik} - m_{ik})}{\sigma_{ik}^2} \end{aligned} \quad (18)$$

$$\begin{aligned} \Delta \sigma_{ik} &= -\eta_\sigma \frac{\partial E}{\partial \sigma_{ik}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial b_k} \frac{\partial b_k}{\partial r_{ik}} \frac{\partial r_{ik}}{\partial \sigma_{ik}} \\ &= \eta_\sigma (y_0 - y) \cdot w_k \cdot b_k \cdot \frac{2(I_{ik} - m_{ik})^2}{\sigma_{ik}^3} \end{aligned} \quad (19)$$

$$\begin{aligned} \Delta w_{ik} &= -\eta_{wik} \frac{\partial E}{\partial w_{ik}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial b_k} \frac{\partial b_k}{\partial r_{ik}} \frac{\partial r_{ik}}{\partial I_{ik}} \frac{\partial I_{ik}}{\partial w_{ik}} \\ &= \eta_{wik} (y_0 - y) \cdot w_k \cdot b_k \cdot \frac{2(m_{ik} - I_{ik})}{\sigma_{ik}^2} \cdot r_{ik}(t-1) \end{aligned} \quad (20)$$

4.3 Convergence Analyses [29]

Theorem 1 Let η_p be the learning-rates for the P in (16). Then, the convergence of tracking error is guaranteed if η_p is chosen as

$$0 < \eta_p < \frac{2}{Y^2} \quad (21)$$

where $Y = \frac{\partial y(N)}{\partial P}$.

Proof Define a Lyapunov functions as

$$V(N) = \frac{1}{2} e(N)^2 \quad (22)$$

Then, the change of the Lyapunov function is obtained as

$$\begin{aligned}\Delta V(N) &= V(N+1) - V(N) \\ &= \frac{1}{2}(e(N+1)^2 - e(N)^2) \\ &= \frac{1}{2}(e(N+1) + e(N)) \cdot \Delta e(N)\end{aligned}\quad (23)$$

According to the Taylor formula and the chain rule, the error function variation can be expressed as

$$\Delta e(N) \cong \frac{\partial e(N)}{\partial P} \cdot \Delta P \quad (24)$$

Among these,

$$\frac{\partial e(N)}{\partial P} = \frac{\partial e(N)}{\partial y(N)} \cdot \frac{\partial y(N)}{\partial P} = -1 \cdot \frac{\partial y(N)}{\partial P} \quad (25)$$

and

$$\Delta P = -\eta \frac{\partial E(N)}{\partial P} = -\eta \cdot (-e(N)) \cdot \frac{\partial y(N)}{\partial P} = \eta \cdot e(N) \cdot \frac{\partial y(N)}{\partial P}. \quad (26)$$

Using (25) and (26), (24) becomes

$$\begin{aligned}\Delta e(N) &\cong \frac{\partial e(N)}{\partial P} \cdot \Delta P \\ &= (-1 \cdot \frac{\partial y(N)}{\partial P}) \cdot (\eta \cdot e(N) \cdot \frac{\partial y(N)}{\partial P}) \\ &= -\eta \cdot e(N) \cdot (\frac{\partial y(N)}{\partial P})^2 \equiv -\eta \cdot e(N) Y^2.\end{aligned}\quad (27)$$

From (27), $\Delta V(N)$ can be represented as

$$\begin{aligned}\Delta V(N) &= \frac{1}{2}(2e(N) + \Delta e(N)) \cdot \Delta e(N) \\ &= \frac{1}{2}\eta e(N)^2 Y^2 (\eta Y^2 - 2)\end{aligned}\quad (28)$$

If η_p is chosen as $0 < \eta_p < \frac{2}{Y^2}$, $\Delta V(N) < 0$ is guaranteed obviously. And easily, for any $0 \leq y_0(N) \leq 1$, $0 \leq y(N) \leq 1$, there is $V(N) \geq 0$. Therefore, the Lyapunov stability of $V(N) \geq 0$ and $\Delta V(N) < 0$ is guaranteed, which can explain the convergence of tracking error. The derivation results prove that the convergence of RCMNN model is guaranteed.

5 Simulation Result

In this experiment, the goal of the model training is to find the values of the parameters m_{ik} , σ_{ik} , w_{ik} and W that minimizes the overall error. We take the 7-dimensional redundant data set and 6-dimensional non-redundant data set as the input data of the RCMNN (Fig. 5) respectively, and the joint signals as output data. We randomly selected 80% data for training and verification, 20% data for testing. To compare the RCMNN with other methods, we use CMNN, and the most popular artificial neural network BP. The performance of trained models was subsequently tested by comparing predicted torque values from the model and the measured torque values. RMSE (Root Mean Square Error), NMAE (Normalized Mean Absolute Error), NRMSE (Normalized Root Mean Square Error), and CC (Correlation Coefficient) between the predicted joint torque and the measure joint torque data are used to evaluate network performance.

5.1 The Updated RCMNN

In the training of RCMNN, in order to make the RCMNN converge faster, the learning-rate η_p is set as

$$\eta_p = \frac{c}{t} \quad (29)$$

In which c is a constant, and t is the training times. Obviously, the equation is in line with the law of error convergence. Because as the number of training increases, the error will continue to decrease, and when the error is small enough, an excessive learning rate will cause the error to oscillate. Thereby a decreasing learning rate accelerates the rate of convergence during model training, making the model meet the standard faster.

After 50 times training, the results are drawn in Fig. 6 (a). Note that the output of RCMNN is generally close to the true value with a slight vibration, which is obvious at the bottom of the waveform. To smooth the waveform, the smooth function is applied in RCMNN, which is represented as follows

$$f(t) = c_1 f(t) + c_2 f(t-1) + c_3 f(t-2) \quad (30)$$

where $f(t)$ is the data point at time t , and c_1 , c_2 , c_3 are set as 0.5, 0.3, 0.2, respectively.

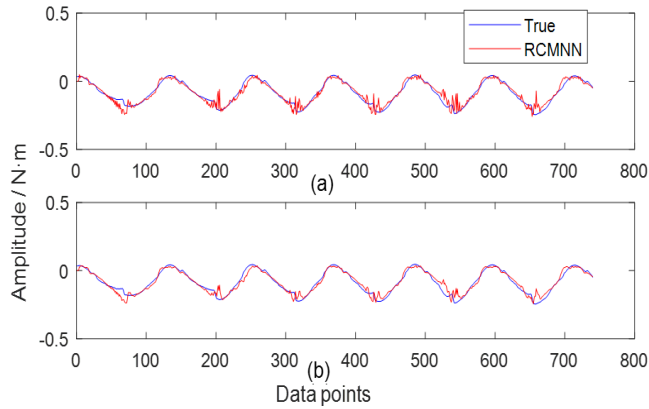


Figure 6
Result of training data

Table 2
Comparison of the three cases

Results	RMSE	NMAE	NRMSE	CC
(a)	0.0273	0.1937	0.2299	0.9454
(b)	0.0239	0.1828	0.2009	0.9582

P.S. RMSE (Root Mean Square Error) = $\sqrt{\frac{1}{n} \sum_1^n (y_0 - y)^2}$

NMAE (Normalized Mean Absolute Error) = $\frac{\sum_1^n |y_0 - y|}{\sum_1^n |y_0|}$

NRMSE (Normalized Root Mean Square Error) = $\sqrt{\frac{\sum_1^n (y_0 - y)^2}{\sum_1^n (y_0)^2}}$

CC (Correlation Coefficient) shown on formula (2)

As shown in Fig. 6 (b) and Table 2, the frequency and amplitude of oscillation have a great reduction compared to the original output (Fig. 6 (a)).

In the testing of RCMNN, the input data is sent to the predictor, and the prediction results and smoothed results are shown in Fig. 7 and the specific parameters are shown in Table 3.

Table 3
Comparison of the two cases

Results	RMSE	NMAE	NRMSE	CC
a	0.0281	0.2333	0.2483	0.9555
b	0.0254	0.2188	0.2248	0.9640

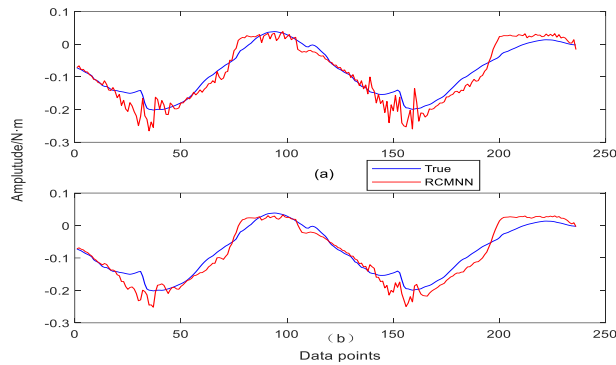


Figure 7

Result of testing data

Results show the peaks and troughs of the output of the model appear significantly earlier than the sample output, while the smoothed RCMNN greatly improves the situation of oscillation, proving the validity of RCMNN.

5.2 Results

The results obtained by the RCMNN, CMNN, and BP network of the redundant group and the non-redundant group are shown in Table 4 and Table 5 respectively. Table 4 and Table 5 show that the performance of the RCMNN is better than that of BP and CMNN, due to its recurrent unit. In addition, according to Table 4 and Table 5, although the redundant group has one more dimensionality information related to the torque of ankle than the non-redundant group, the latter generally has better performance, like RMSE, NMAE, NRMSE, and CC. Hence, our subsequent experiments use the non-redundant data group.

Table 4
Results of the redundant group

	RCMNN	CMNN	BP
RMSE	0.0254	0.0358	0.0254
NMAE	0.2188	0.3267	0.2157
NRMSE	0.2248	0.3168	0.2252
CC	0.9640	0.8826	0.9495

Table 5
Results of the non-redundant group

	RCMNN	CMNN	BP
RMSE	0.0213	0.0335	0.0233
NMAE	0.1810	0.2878	0.2056
NRMSE	0.1881	0.2957	0.2068
CC	0.9639	0.9035	0.9488

In the previous section, equation (30) is employed to smooth the output of RCMNN, and make an excellent result with a not suitable enough coefficient. In this part, an algorithm is summarized in Algorithm 1 to get a more appropriate coefficient.

Algorithm 1 adaptive coefficient algorithm

1 : Set: $X(t)=[y(t-2),y(t-1),y(t)]$; $Y(t)=y_0(t)$; a weight vector with same number to the number of formula (30) coefficient and a threshold m .

2 : Train: Divide $X(t)$ and $Y(t)$ into training data and testing data; train the network until the correlation coefficient between the output of BP and is bigger than m .

3 : Finish: Assign the weight to the coefficient of formula (30), and use testing data to verify whether the coefficient is suitable.

With the coefficient gotten by Algorithm 1, the non-redundant group shows its experimental results as Table 6.

Table 6
The results with the coefficient gotten by algorithm 1

Subjects	RMSE	NMAE	NRMSE	CC
1	0.0023	0.0266	0.0291	0.9864
2	0.0026	0.0194	0.0235	0.9866
3	0.0025	0.0364	0.0427	0.9843
4	0.0028	0.0216	0.0226	0.9867

In Table 6, after the weights obtained after training by the BP neural network are used as the coefficients of Formula (30), the experimental results obtained by the predictor are greatly improved. The correlation coefficient between the experimental output results and the torque data has been increased by more than two percentage points, and the improved performance of other parameters are also obvious. One of the curves on training data and testing data is drawn as follows.

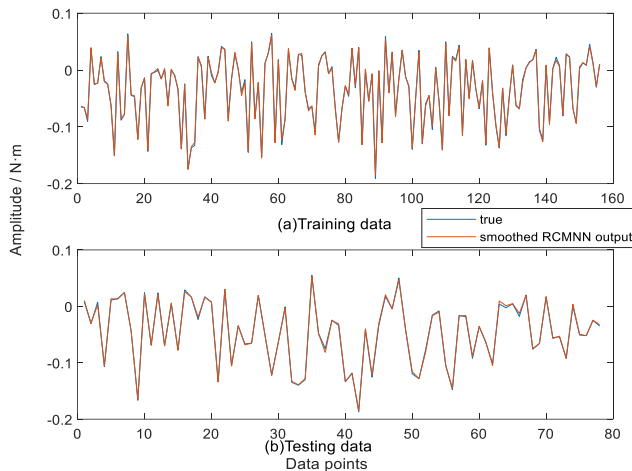


Figure 8

Performance of RCMNN after smoothing

Conclusions

In this study, a RCMNN is employed as a predictor to estimate torque of ankle joint from velocity, position and sEMG signals. RCMNN shows excellent fitting ability in torque prediction, that is the correlation coefficient between torque and estimation result reaches 98.43%, and other performance parameters like RMSE, NMAE, and NRMSE are also satisfactory. Moreover, RCMNN accelerates the torque prediction, making the predictor a control source of the bionic prosthesis and rehabilitation robot. Hence, the outstandingly fitting ability of RCMNN has great application prospects on torque prediction.

References

- [1] T. Kikuchi, C. Sato, K. Yamabe, et al. "Upper limb training/assessment program using passive force controllable rehabilitation system," IEEE International Conference on Rehabilitation Robotics, pp. 505-510, 2017
- [2] Q. B. Lin, S. C. Chen and C. M. Lin, "Parametric fault diagnosis based on fuzzy cerebellar model neural networks," IEEE Transactions on Industrial Electronics, Vol. 66, No. 10, pp. 8104-8115, 2019
- [3] R. Merletti and P. Parker, "Electromyography: physiology, engineering, and noninvasive applications." New York, USA: IEEE Press, pp. 381-425, 2004
- [4] M. Atzori, A. Gijsberts, C. Castellini, et al. "Clinical parameter effect on the capability to control myoelectric robotic prosthetic hands," Journal of Rehabilitation Research & Development, Vol. 53, No. 3, pp. 345-58, 2016

-
- [5] C. S. Li , Y. Q. Zhou and Y. Li, "The signal processing and identification of upper limb motion based on sEMG," *Wireless Personal Communications*, Vol. 103, No. 1, pp. 887-896, 2018
- [6] B. P. Xiong, N. Y. Zeng, H. Li, et al. "Intelligent prediction of human lower extremity joint moment: an artificial neural network approach". *IEEE Access*, Vol. 7, pp. 29973-29980, 2019
- [7] J. Chu, I. Moon, Y. Lee, S. Kim and M. Mun, "A supervised feature-projection-based real-time EMG pattern recognition for multifunction myoelectric hand control," *IEEE/ASME Transactions on Mechatronics*, Vol. 12, No. 3, pp. 282-290, 2007
- [8] W. Y. Cheng, C. F. Juang. "A fuzzy model with online incremental SVM and margin-selective gradient descent learning for classification problems," *IEEE Transactions on Fuzzy Systems*, Vol. 22, No. 2, pp. 324-337, 2014
- [9] R. E. Precup, T. A. Teban, A. Albu, et al. "Evolving fuzzy models for prosthetic hand myoelectric-based control," *IEEE Transactions on Instrumentation and Measurement*, Vol. 69, No. 7, pp. 4625-4636, 2020
- [10] C. F. Juang, Y. Y. Lin, R. B. Huang, "Dynamic system modeling using a recurrent interval-valued fuzzy neural network and its hardware implementation," *Fuzzy Sets & Systems*, Vol. 179, No. 1, pp. 83-99, 2011
- [11] R. Zall and M. R. Kangavari, "On the construction of multi-relational classifier based on canonical correlation analysis," *International Journal of Artificial Intelligence*, Vol. 17, No. 2, pp. 23-43, 2019
- [12] E. Hedrea, R. Precup, R. Roman, et al. "Tensor product- based model transformation approach to tower crane systems modelling," *Asian Journal of Control*, DOI: 10.1002/asjc.2494, 2021
- [13] C. Pozna, R. E. Precup, "Applications of signatures to expert systems modelling," *Acta Polytechnica Hungarica*, Vol. 11, No. 2, pp. 21-39, 2014
- [14] H. Kim, H. Park, S. Lee, et al. "Joint torque estimation using sEMG and deep neural network," *J. Electr. Eng. Technol.* Vol. 15, pp. 2287-2298, 2020
- [15] L. Lu, Q. Wu, X. Chen et al. "Development of a sEMG-based torque estimation control strategy for a soft elbow exoskeleton," *Robotics and Autonomous Systems*, Vol. 111, pp. 88-98, 2019
- [16] D. Xu, Q. Wu, and Y. Zhu, "Development of a sEMG-based joint torque estimation strategy using hill-type muscle model and neural network," *Journal of Medical and Biological Engineering*, pp. 1-11, 2020
- [17] M. H. Jali, T. A. Izzuddin, Z. H. Bohari, et al. "Predicting EMG based elbow joint torque model using multiple input ANN neurons for arm rehabilitation," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 188-193, 2014

-
- [18] Y. Liu, S. M. Shih, S. L. Tian, et al. "Lower extremity joint torque predicted by using artificial neural network during vertical jump". *Journal of Biomechanics*, Vol. 42, No. 7, pp. 906-911, 2009
- [19] J. Han, Q. Ding, A. Xiong and X. Zhao, "A state-space EMG model for the estimation of continuous joint movements," *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 7, pp. 4267-4275, 2015
- [20] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation control," *Journal of Dynamic System, Measurement and Control*, Vol. 97, No. 3, pp. 220-227, 1975
- [21] J. S. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control*, Vol. 97, No. 3, pp. 228-233, 1975
- [22] J. S. Guan, C. M. Lin, G. L. Ji, L. W. Qian and Y. M. Zheng, "Robust adaptive tracking control for manipulators based on a TSK fuzzy cerebellar model articulation controller," *IEEE Access*, Vol. 6, pp. 1670-1679, 2018
- [23] T. L. Le, T. T. Huynh, and C. M. Lin, "Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using PSO" *International Journal of Fuzzy Systems*, Vol. 21, No. 8, pp. 2524-2541, 2019
- [24] J. Zhao, Z. Zhong, C. M. Lin and H. K. Lam, "H ∞ tracking control for nonlinear multivariable systems using wavelet-type TSK fuzzy brain emotional learning with particle swarm optimization" *Journal of the Franklin*, Vol. 358, No. 1, pp. 650-673, 2021
- [25] C. M. Lin, V. H. La and T. L. Le, "DC - DC converters design using a type-2 wavelet fuzzy cerebellar model articulation controller", *Neural Computing and Applications*, Vol. 32, No. 7, pp. 2217-2229, 2020
- [26] T. T. Huynh, C. M. Lin, T. L. Le, H. Y. Cho, T. T. T. Pham, N. Q. K. Le and F. Chao, "A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped Gaussian membership Functions," *IEEE Transactions on Industrial Electronics*, Vol. 67, No. 11, pp. 9671-9682, 2020
- [27] M. Sartori, M. Reggiani, D. Farina and D. G. Lloyd, "EMG-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity," *PLoS ONE*, Vol. 7, No. 12, pp. e52618, 2012
- [28] C. M. Lin, L. Chen and D. S. Yeung, "Adaptive filter design using recurrent cerebellar model articulation controller," *IEEE Transactions on Neural Networks*, Vol. 21, No. 7, pp. 1149-1157, 2010
- [29] T. L. Le, T. T. Huynh and C. M. Lin, "A K-means interval type-2 fuzzy neural network for medical diagnosis", *International Journal of Fuzzy Systems*, Vol. 21, No. 7, pp. 2258-2269, 2019