Artificial Organs

EMG-Based Estimation of Limb Movement Using Deep Learning With Recurrent Convolutional Neural Networks

Peng Xia, Jie Hu, and Yinghong Peng

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

Abstract: A novel model based on deep learning is proposed to estimate kinematic information for myoelectric control from multi-channel electromyogram (EMG) signals. The neural information of limb movement is embedded in EMG signals that are influenced by all kinds of factors. In order to overcome the negative effects of variability in signals, the proposed model employs the deep architecture combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The EMG signals are transformed to time-frequency frames as the input to the model. The limb movement is estimated by the model that is trained with the gradient descent and backpropagation procedure. We tested the model for simultaneous and proportional estimation of limb movement in eight healthy subjects and compared it with support vector regression (SVR) and CNNs on the same data set. The experimental studies show that the proposed model has higher estimation accuracy and better robustness with respect to time. The combination of CNNs and RNNs can improve the model performance compared with using CNNs alone. The model of deep architecture is promising in EMG decoding and optimization of network structures can increase the accuracy and robustness. **Key Words:** Electromyogram—Myoelectric control—Deep learning—Convolutional neural network—Recurrent neural network.

Myoelectric control is an effective solution used to operate robots and treat limb amputation. Surface electromyography (EMG) is the technique for controlling currently available myoelectric prostheses, which is generated from the electrical manifestation of the muscles and contains information about the neural signals sent from the spinal cord to control the muscles. The signals are noninvasively detected on the surface of skin, which reflect the muscle activity and provide limb movement information. The surface EMG signals are used as the input to myoelectric control systems, which are applied in active prostheses (1), operation of robots (2), and EMG-controlled exoskeletons (3).

The extensively studied myoelectric control strategy is pattern recognition, in which it is assumed that there exists a set of distinguishable and repeatable patterns in EMG signals during different types of movements. The patterns are learned by pattern recognition algorithms that consist of two main steps: feature extraction and classification. Feature extraction and classification of the EMG have been performed in many ways (4–9). It is now accepted that among the large number of possible combinations of features and classifiers, the proper selections can provide a good balance between recognition performance and computational efficiency, and the classification accuracy of over 11 classes of movements can reach more than 95% (10).

In spite of the significant performance of the myoelectric control methods, few artificial limbs based on EMG pattern recognition are currently used in commercial and clinical situations, where the environments are not as conditioned and repeatable as in research laboratories (11,12). The pattern recognition-based myoelectric control systems employ a mutex strategy, in which only one class can be selected in one decision, and the number of patterns for the controller is limited. However, natural movements of the limb are usually

doi: 10.1111/aor.13004

Received April 2017; revised June 2017; accepted July 2017.
Address correspondence and reprint requests to Jie Hu, School of Mechanical Engineering, Shanghai Jiao Tong University, 800 Dongchuan Rd, Minhang District, Shanghai 200240, China. E-mail: hujie@sjtu.edu.cn

simultaneous and smooth, which requires the coordination of multiple physiological degrees of freedom (DOFs) across several joints. Therefore, simultaneous and continuous control of multiple DOFs is a necessary challenge for multifunction prostheses research. Jiang et al. (13–15) proposed an approach based on the muscle synergy concept (16), which can simultaneously estimate wrist kinematic information by extracting a linear mapping between neural control signals and muscle activation with nonnegative matrix factorization (NMF) algorithms. Nielsen et al. (17) applied a mirrored bilateral training strategy to the above method and used multi-layer perceptrons (MLPs) to mitigate the effect of nonlinear association between EMG features and hand motion. Artemiadis et al. (18,19) presented a learning scheme for an anthropomorphic robot arm with myoelectric control in the three dimensional (3D) space, which takes advantage of both classification and regression algorithms. The classifiers discriminate the subspace and task of movements, and the mapping between EMG signals and continuous motion is estimated for specific task. They improved the proportional control with the development of muscle synergies (20,21). Scheme et al. (22–24) not only explored the application of artificial neural networks (ANNs) and support vector machine (SVM) but also investigated force, position, and visual target based training paradigms for improving simultaneous and proportional myoelectric control. However, despite the improvements in the scientific research, the robust natural control of robotic prosthesis is still very challenging (12). Faced with EMG variability over the use of prosthesis, such as electrode repositioning, sweat at electrode-skin site, fatigue, and physiological factors (1), the classical machine learning methods have not shown significant power for myoelectric control. This article is aimed at developing an approach based on deep learning technology, which is good at finding complicated features and relations in high-dimensional data (25).

In recent years, the applications of deep learning technology have demonstrated impressive power in many domains of modern society and have presented significant advantages over other machine learning methods (26). Traditional myoelectric control schemes, including carefully constructed and chosen features, require domain expertise and are often experiential choices. Deep learning methods can extract features on multiple levels of representation and learn very complex functions with composition of enough representation layers. The key point is that the features of each layer are learned with a general learning procedure without human designing (25).

The convolutional neural network (CNN) is one particular type of feed-forward network and has the ability of time-frequency information mining, which is widely applied in data processing such as images, audio, and video (27–30). Atzori et al. (31) applied CNNs to EMG-based hand movement recognition. In their methods, the signal data is subsampled and low-pass filtered as the model input, and the model is composed of convolutional, pooling, and softmax layers. The results showed that the classification accuracy of CNNs was higher than the average results of classical methods but lower than the best reference methods. Their research has inspired further study on the application of deep neural networks to myoelectric control.

Recurrent neural networks (RNNs) are good at processing sequential data, such as speech and language (32,33), in that the history information of a sequence can be maintained in a multilayer network. EMG signals, as time sequential data, demand storage of history information to make up for the deficiency of using time window function. Recurrent convolutional neural networks (RCNNs), combining the advantages of CNNs and RNNs, brought the promising results for scene labeling and video classification (34). Based on the properties of deep neural networks, RCNNs can be utilized to process biomedical signals (35). In this article, we present a novel approach for extracting movement information of upper limb from surface EMG signals. In this approach, EMG signals are transformed into time-frequency frames and taken as the inputs of the RCNN model. The gradient descent and backpropagation procedure is used to train the RCNN model with EMG and motion data.

SUBJECTS AND METHODS

Subjects and data acquisition

Eight healthy subjects (three females, five males; age range: 22–40 years) participated in the experiment. For each subject, five pairs of surface bipolar electrodes were placed on shoulder and arm to record a group of five muscles: biceps brachii, triceps brachii, anterior deltoid, posterior deltoid, and middle deltoid (see Fig. 1). The signals were recorded by EMG system (NCC Medical Co., LTD, Shanghai, China) at the frequency of 2 kHz

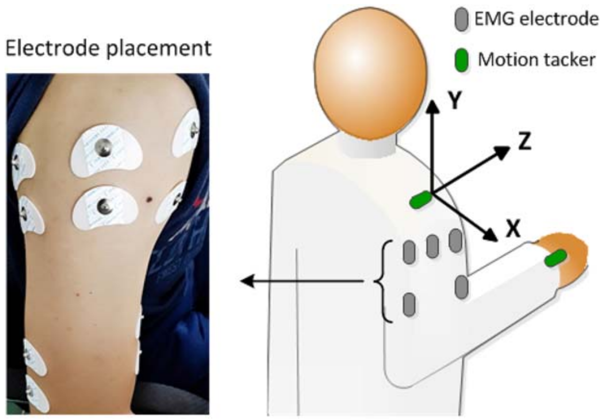

FIG. 1. The placement of EMG electrodes and motion trackers in the data acquisition system. The hand motion trajectories are recorded by position tracking system and the origin of 3D coordinates is the position of acromion. [Color figure can be viewed at wileyonlinelibrary.com]

through five channels that is from the five pairs of surface bipolar electrodes separately.

The motion of the upper limb was analyzed in 3D space, where approximately 30 muscles actuate 7 DOFs. In this article, we concentrate on the hand position in 3D space and wrist motion is not included in the analysis. The elbow and shoulder motion, as the intermediate variable, is not included in the model output. The kinematic vector $P=[x, y, z]$ denotes the position of hand in 3D space and the origin of 3D coordinates is the position of acromion (see Fig. 1). The movement of hand was tracked at the frequency of 50 Hz by the position tracking system (Xsens Technologies B.V., Enschede, The Netherlands). The EMG recording and the position tracking were synchronized with analysis windows, in which EMG recording was segmented by a 50 ms window with 20 ms overlap so that each analysis window was synchronized to one sample of position data. During an experimental session, a subject was instructed to move his arm in the possible ranges of motion. The data recorded was used to train the deep neural networks. The experiment was performed in accordance with the Declaration of Helsinki.

Data processing model

Model input

In order to organize the data into the input maps for neural networks, the EMG signals of each channel are represented in time-frequency scale. The Hamming window was used as the window function to segment the signals. In frequency domain, the raw signals were transformed into linear spectral coefficients based on Fourier transformation in each analysis window. As a consequence, the continuous EMG signals can be represented with time-frequency spectrum as the input of the model (see Fig. 1), which are consecutive frames of spectral coefficients. The data of input was normalized so that the data have zero mean. The vector $V_k=[v_1\ v_2 \cdots\ v_B]$ denotes the kth frame of a channel and B is the number of frequency bands. In this article, B=20, which means 20 coefficients were computed in a time-frequency frame. The time-frequency spectrum of ith channel can be represented by the matrix $I_i=[V_{i,1}\ \ V_{i,2} \cdots V_{i,K}]$ in K time steps. The five channels of signals were transformed into a group of time-frequency spectra as the whole input of model, in which five time-frequency spectra were simultaneous and the frames at kth time step can be represented by $\{V_{i,k}=[v_{i,k,1}\ v_{i,k,2}\ \cdots\ v_{i,k,20}] \mid i=1,\ 2, \cdots, 5\}$.

CNN approach

The CNN is an extension of standard neural networks for processing data in the form of multiple arrays such as signals, language, images, etc. The key components of CNN are convolutional and pooling layers. In a convolutional layer, each unit is in one of the feature maps and connected to a local patch in input feature maps from the previous layer. The neighboring units connect to the patches that move a few rows or columns. A set of connection weights constitutes a filter bank and the units in a feature map share the same connection weights. Different feature maps use the filter bank with different connection weights to detect the pattern of input data. The outcomes of the operation pass through a nonlinear activation function (such as a hyperbolic tangent). A pooling layer often follows a convolutional layer and pooling operation is used to reduce the dimensionality of data and merge similar features. The units in a typical pooling layer use maximization activation function to take maximum unit from the patch of input feature maps. The connection mode of pooling layer is the same as that in convolutional layer. Several convolutional and pooling layers are stacked and followed with fully connected layers.

As is shown in Fig. 2, the one dimensional (1D) CNN processes a frame of time-frequency input. At a time step, the input feature maps (time-frequency frames) to a convolutional layer are denoted as vectors $\{X_i=[x_{i,1}\ x_{i,2}\ \cdots\ x_{i,N}] \mid i=1,\ 2, \cdots, I\}$, where N is the number of features in a map and I is the number of input maps. In the input layer, feature maps are time-frequency frames from five channels

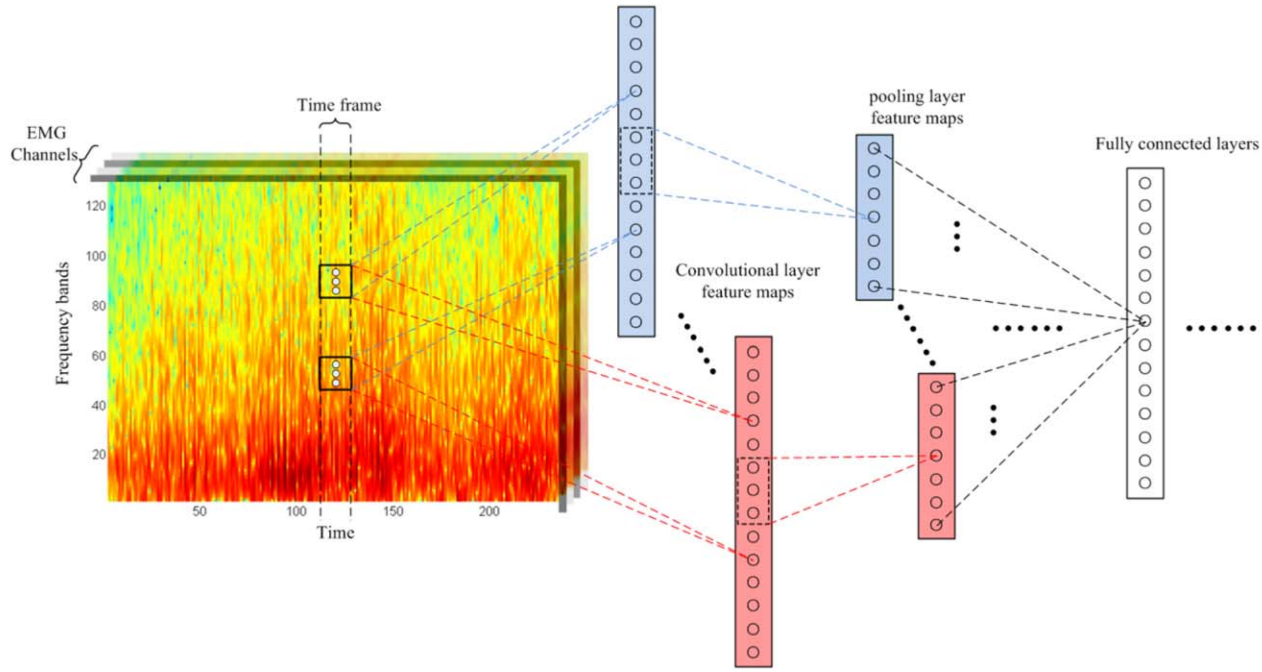

FIG. 2. An illustration of the CNN approach for processing a frame of input. The 1D convolution is executed along frequency bands with the same filter bank in a feature map. In the picture, the size of filter banks of the convolutional layer is three. [Color figure can be viewed at wileyonlinelibrary.com]

and $I=5$. The weight vectors of filter banks are denoted as $\{W_j=[w_{j,1}\ w_{j,2}\ \cdots\ w_{j,S}]\mid j=1, 2,\cdots,J\}$, where S is the size of a filter bank and J is the number of filter banks. The convolutional layer feature maps are computed as:

$$y_{j,k}=\sigma\left(\sum_{i=1}^{I}\sum_{l=1}^{S}w_{j,l}x_{i,k+l-1}+b_k\right),\quad k=1,\ 2,\cdots,K \tag{1}$$

Where $y_{j,k}$ is the kth unit of the jth feature map in a convolutional layer, b_k is the bias and σ is non-linear activation function. According to the equation, the number of features in a convolutional layer map K is decided by the number of features in the previous layer N and the size of a filter banks S, which is computed as:

$$K=N-S+1 \tag{2}$$

The specific size of filter banks is described in the whole architecture. The number of filter banks is the number of feature maps in a convolutional layer, which is related to the specific architecture of model.

The feature maps in a pooling layer, which follows a convolutional layer, are denoted as $\{P_h=[p_{h,1}\ p_{h,2}\ \cdots p_{h,M}]|h=1,\ 2,\cdots,H\}$, where M is the number of units in the feature maps and H equals to the number of maps in the previous layer. The units in a max-pooling layer are computed as:

$$p_{h,m}=\max_{l=1,2,\cdots,q}\left(y_{j,(m-1)\times r+l}\right),\quad m=1,\ 2,\cdots,M,\ \ h=j \tag{3}$$

Where $p_{j,m}$ is the mth unit of the hth output map, q is the pooling size and r is the number of moving rows. If q is larger than r, adjacent pooling patches overlap each other. According to the equation, the number of features in a pooling layer map M is decided by the number of features in the previous layer map K, the pooling size q and the number of moving rows r, which can be computed as:

$$M=\frac{K-q+r}{r} \tag{4}$$

The specific pooling size and number of moving rows are described in the whole architecture.

RNN approach

The dependency of sequential data is a significant issue. Quite a few models use sliding context windows to process partial data of interest, which

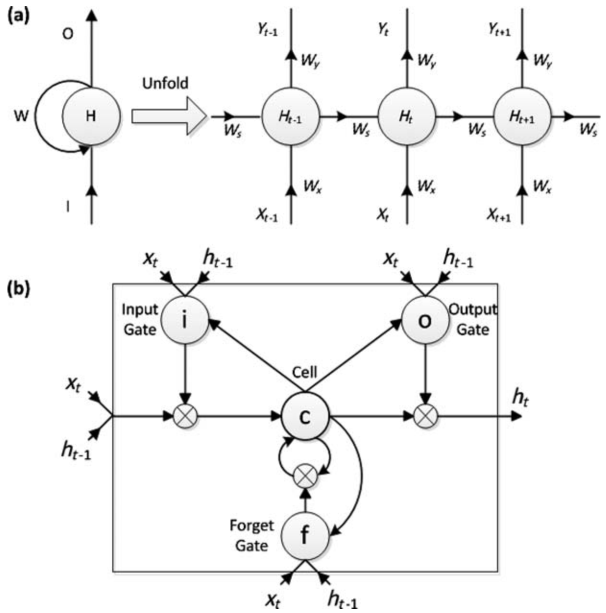

FIG. 3. (a) An illustration of the RNN approach. The output of a time step is connected with weights to the input of next time step so that RNNs can map the input sequence X_t to the output sequence Y_t and each output depends on all the previous inputs. (b) An illustration of LSTM unit.

preclude the dependent data outside the context window. RNNs have the ability to selectively contain and pass information across time steps with hidden states. The typical architecture of RNNs is shown in Fig. 3a. The output of a hidden unit at a time step is taken as the input of the same unit at next time step. When recurrence is unfolded and units at different time steps are connected, hidden states passing through time is shown in the complete sequence. The input vector sequence of a unit is denoted as $X=[x_0, x_1, \cdots, x_t, \cdots]$. The output vector sequence $Y=[y_0, y_1, \cdots, y_t, \cdots]$ and the hidden state vectors $H=[h_0, h_1, \cdots, h_t, \cdots]$ are computed as:

$$h_t=\mu(W_x x_t+W_h h_{t-1}+b_h) \tag{5}$$

$$y_t=\delta\left(W_y h_t+b_y\right) \tag{6}$$

Where W_x, W_h, and W_y are the weight matrices for input-to-hidden, hidden-to-hidden, and hidden-to-output connections. b_s and b_y are the biases vector for hidden layer and output. μ and δ are nonlinear activation functions. Because of the gradient exploding or vanishing problem in backpropagation procedure (36), the long short-term memory (LSTM) architecture is used in the hidden layers (37). Figure 3b illustrates the version of LSTM unit in the article (38). The LSTM is implemented as follows:

$$i_t=\alpha(W_{xi}x_t+W_{hi}h_{t-1}+W_{ci}c_{t-1}+b_i) \tag{7}$$

$$f_t=\alpha\left(W_{xf}x_t+W_{hf}h_{t-1}+W_{cf}c_{t-1}+b_f\right) \tag{8}$$

$$o_t=\alpha(W_{xo}x_t+W_{ho}h_{t-1}+W_{co}c_{t-1}+b_o) \tag{9}$$

$$c_t=f_t * c_{t-1}+i_t * \tanh(W_{xc}x_t+W_{hc}h_{t-1}+b_c) \tag{10}$$

$$h_t=o_t * \ \tanh(c_t) \tag{11}$$

Where α is the sigmoid function and $*$ is the product of corresponding elements in two vectors. W terms are the weight matrices of connections (e.g., W_{xi} is the weight matrix between input vector x_t and input gate) and b terms are the bias vectors of connections. i, f, o, and c are the input gate, forget gate, output gate, and internal cell vectors respectively. The input gate modulates the input to internal cell, the forget gate allows LSTM unit to forget the previous memory, and output gate decide the output of the hidden layer.

The architecture of RCNN model

Combining the properties of RNNs and CNNs, the RCNN model is implemented for EMG signals processing in this article. The architecture of RCNN model is shown in Fig. 4. At a time step, five time-frequency vectors are the maps in the input layer. There are three 1D CNN layers in the model and all the sizes of filter banks are 3. The number of filter banks in the first and second convolutional layer is 32, and the number of filter banks in the third convolutional layer is 64. The rectified linear unit (ReLU) is used as the activation function of all the convolutional layers. The pooling size is 2 and number of moving rows is also 2 in the two pooling layers. All the nodes in the feature maps of the second pooling layer are fully connected to the first LSTM layer. The output dimensionality of both LSTM layers is 20, which means the number of elements in each of the gate and cell vectors is 20. The internal cell and output of LSTM layers are saved and used at the next time step. The output layer is a fully connected layer with three nodes that corresponds to the position vector in 3D space. Because the model is used for the regression analysis, no activation function is used in the output layer. The output of RCNN model is a sequence of position vectors through time steps that denote the movements of upper limb.

Model training and testing

The proposed model was used for decoding the EMG signals from eight subjects. The EMG data was collected in six sessions for each subject and

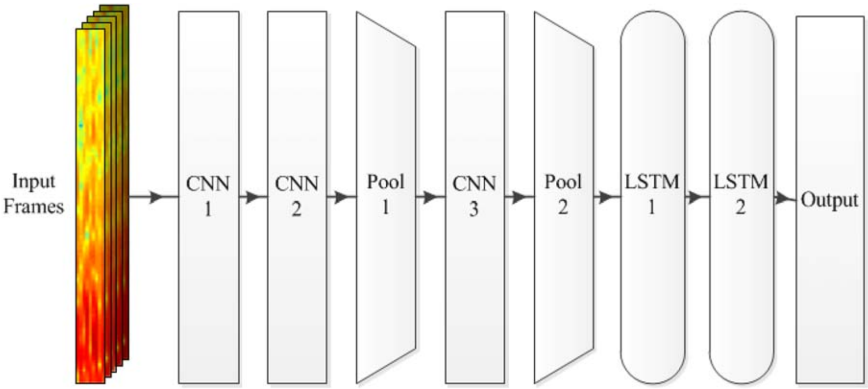

FIG. 4. The architecture of the RCNN model. [Color figure can be viewed at wileyonlinelibrary.com]

there was a time interval of 20 minutes for resting between two data collection sessions. The recurrent model was trained with sequential data, so the data of a session was divided into 15 seconds long segments as training samples. The data of a subject was used for training and testing a set of model parameters and different subjects corresponded to different model parameters. The cross validation was used for model comparison in the training and testing. In the time robustness experiment, models were trained with the first session of data and tested with the other five sessions. The decoding process was recorded and the results were analyzed.

The model training is optimizing the parameters of neural networks by gradient descent method (39). The gradients are computed using backpropagation procedure, which is taking the derivative with the chain rule. The output of model is continuous estimated value in 3D space, so the loss function of model is computed as mean square error of estimated position $\hat{P}_t=[\hat{x}_t, \hat{y}_t, \hat{z}_t]$:

$$L_t=\frac{1}{2}\left[(\hat{x}_t-x_t)^2+(\hat{y}_t-y_t)^2+(\hat{z}_t-z_t)^2\right] \tag{12}$$

where L is the loss function and minimized over training data. The gradient of loss function with respect to the connection weights in neural networks can be computed with backpropagation procedure to update weights matrix. There is no weight in max-pooling layers needing training and the error is propagated to the maximum unit in the pooling patch.

The weights of model were initialized with random values in a normal distribution. The range of random values was decided by the input data range. The multilayer networks were trained by stochastic gradient descent (SGD). The hyper-parameters were tuned by random search (40) on the original data set. The momentum was set at 0.95, the learning rate at 0.002 and the weight decay at 0.0005. The training batch size was 20 and epoch number was 30. The dropout regularization technique was used to prevent the model from overfitting (41) and the dropout rate was set at 30%. The larger amount of training data can improve the performance of deep model, so data augmentation was performed by adding white Gaussian noise to the original signal with a signal-to-noise ratio 20.

Because the output of model is a time series of 3D coordinates, the measurement of model can be defined as the goodness of estimating 3D trajectory. The criterion value R^2 proposed by (42) is used to measure the model performance:

$$R^2=1-\frac{\sum_{t=0}^{T}(\hat{x}_t-x_t)^2+(\hat{y}_t-y_t)^2+(\hat{z}_t-z_t)^2}{\sum_{t=0}^{T}(x_t-\bar{x}_t)^2+(y_t-\bar{y}_t)^2+(z_t-\bar{z}_t)^2} \tag{13}$$

where T is the length of time series and $[\bar{x}_t, \bar{y}_t, \bar{z}_t]$ is the mean of the position vector. The numerator is the sum of square error from estimated 3D trajectory and the denominator is the total square residual of actual trajectory. R^2 denotes the proportion of total variation in actual trajectory described by model output, so the model performance is better if the R^2 is closer to one.

RESULTS

Model performance comparison

The estimated movement for a segment of data from one subject is shown on three axes in Fig. 5. The RCNN model is compared with support vector regression (SVR) presented in Ref. 24 and CNNs presented in Ref. 31 for EMG signal decoding. In the SVR method, four time domain features (mean absolute value, waveform length, zero crossings, and slope signs changes) in 200 ms sliding window

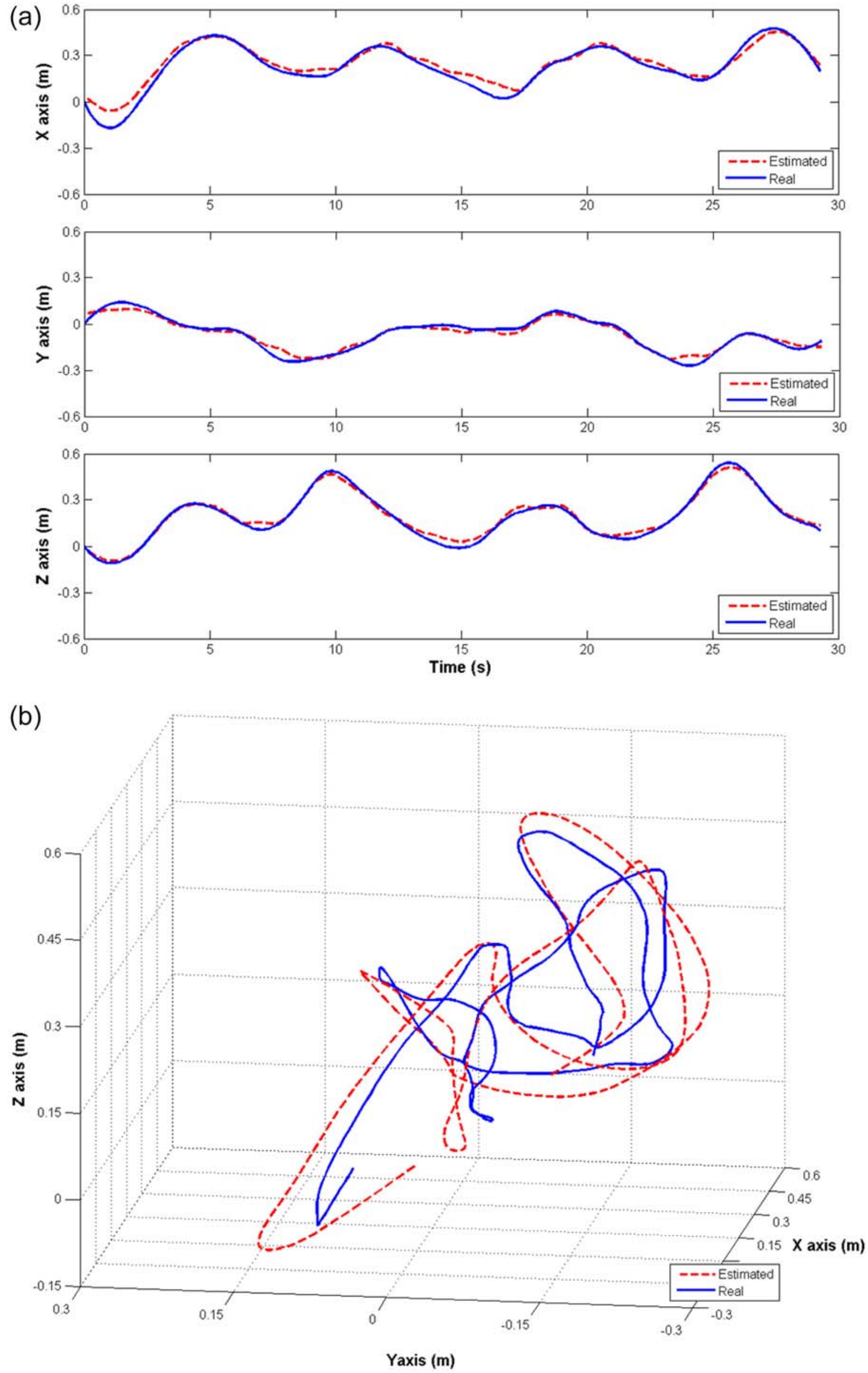

FIG. 5. (a) The estimated movement for a segment of data along three axes. The solid lines are the real trajectory and the dashed lines are the estimated trajectory. (b) The real and estimated trajectory in 3D space. The criterion value R^2 for the results shown is 91.7%. [Color figure can be viewed at wileyonlinelibrary.com]

were extracted as the model input and the radial basis function was used as the kernel type. In the CNNs method, the signal was subsampled at 200 Hz and low-pass filtered at 1 Hz and segmented with 150 ms windows as the model input. The architecture of CNNs and hyper-parameters used in training were based on the reference paper, whereas the output layer was replaced with the one in the RCNN model for regression analysis.

The performances of three methods are represented with the mean R^2 value (mean $\pm$ SD) for combined results across eight subjects from the first and second collection sessions of data. The fivefold cross validation was used for training and testing. Significance level is analyzed with t-test to compare the effects of decoding models and $P \leq 0.01$ is considered significant. As is presented in Fig. 6, the mean R^2 value for RCNNs is $90.3 \pm 4.5\%$ that is

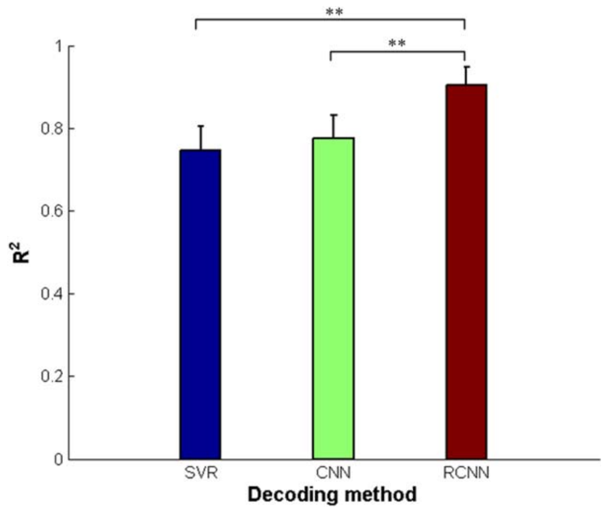

FIG. 6. The performance comparison of three models (SVR, CNN, RCNN), which are represented with the R^2 value (mean ± SD) for combined results across subjects. The R^2 value of RCNN is significantly higher than those of SVR and CNN (** $P \leq 0.01$). [Color figure can be viewed at wileyonlinelibrary.com]

significantly higher ($P \leq 0.01$) than SVR (74.5 ± 6.1%) and CNNs (77.6 ± 5.6%).

Robustness with respect to time

The performance of decoding models depends on the consistency of EMG characteristics, which is influenced by many factors over time, including environmental and physiological changes (1,43). Therefore, it is necessary to test the model robustness to time variation. The first collection session of EMG data was used to train the models and time robustness was tested with the data of the other five collection sessions. As is shown in Fig. 7, the performance of three models all becomes worse over time. The R^2 values for the first testing session are highest and the R^2 value decrease of RCNNs (from 90.6 ± 4.1% to 83.1 ± 5.6%) is less than CNNs (from 78.4 ± 5.5% to 61.9 ± 7.1%) and SVR (from 74.9 ± 5.4% to 50.6 ± 7.3%). In other words, the robustness of RCNNs to time variation is better than CNNs and SVR. Besides, the RCNN model can still keep a relatively good performance over a period of time.

DISCUSSION

For the EMG-based estimation of limb movement in myoelectric control, a decoding model based on deep learning is proposed in this article, which combines the properties of CNNs and RNNs. The results show that the RCNN model can estimate limb movement with high accuracy and has good robustness to time variation of EMG signals. Compared with other linear and nonlinear models, the deep learning method is good at representation obtained with multi-layer neural networks, each layer of which transforms the representation into a more abstract one. With enough layers, deep neural networks can learn very complex functions like the transformation from surface EMG signals to limb movement. The motion information contained in EMG signals depends on the motor neuron spike trains and muscle fiber electrical properties, which can be represented in the frequency domain so that the spectral coefficients of EMG signals is computed in each analysis window as the model input. However, the association between motion information and the frequency spectrum is influenced by so many factors that neuron spike trains are difficult to be completely separated and then transformed to motion information. With convolutional and pooling operation, CNNs can represent the local characteristics, detect the motif along frequency domain and combine lower-level features into higher-level ones, which is efficient in representing the structure of the frequency spectrum and robust against noises. Since motion information and quite a few factors of influence are changing over time, the decoding model should represent the features of time dependencies. RNNs can selectively pass and contain information through time steps, so the approach is used to map

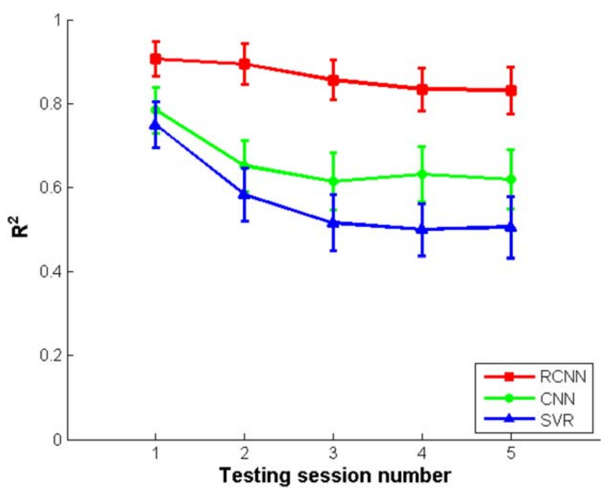

FIG. 7. The performance of three models with respect to time (R^2, mean ± SD). The EMG data was collected in six sessions for each subject and the time interval between two sessions is 20 minutes. The models are trained with data of the first session and used for estimation of the other five testing sessions. [Color figure can be viewed at wileyonlinelibrary.com]

the feature vectors obtained by the CNNs to output sequence. With the advantages of both CNNs and RNNs, the RCNN presents a better performance of decoding the EMG signals than SVR and CNNs.

Compared with traditional methods, such as SVR, the expressive power of deep learning is based on the depth of its architecture (25) defined as the longest path from an input to an output node, which is related to the layers of networks. On the basis of empirical results in acoustic modeling for speech recognition (28,29) whose input has similar characteristics to EMG signals, we proposed a neural network model with seven hidden layers (including pooling layers) in this article. The number of fully connected layers implemented with RNN approach is two because the complexity of surface EMG signals causes that the length of input associated to output is not fixed (32,33). With two recurrent layers, the varying length can be learned as nonlinear mapping between time steps. The CNN approach is applied to the other layers in that the computation efficiency and generalization performance of CNNs are better than those of fully connected networks (25). With the capability of processing raw data, the RCNN model is easy to be applied to other kinds of signals like electroencephalography.

Compared with using CNNs alone, the recurrent layers can represent time dependencies, which are crucial in time sequence analysis. The RCNN model is proposed for simultaneous and proportional estimation of limb movement that is regression analysis, whereas the CNN model is used for classification of hand gestures in Ref. 31. The regression analysis of movement trajectory depends on longer term inputs than classification tasks. Larger networks can achieve better performance on vision and audio tasks. In this article, the forward depth of RCNN model is more than that of the model presented in Ref. 31 and recurrent layers have the depth through time. Another difference is the input to the model: the signal was subsampled at 200 Hz and low-pass filtered at 1 Hz in Ref. 31, which may cause the loss of some high-frequency information in preprocessing. In this article, the spectral coefficients computed with Fourier transformation may lose some time domain information. Nevertheless, the shorter time window, window function and recurrent approach can compensate for the loss. For the sake of model performance, the input should remain the information of signals as much as possible. There is a tradeoff between computation time and model performance in the further research of deep neural networks for EMG decoding.

At the training and testing stage, the model parameters were trained and tested specifically for each person. Because a myoelectric control system is set for one person, different subjects would share the same model architecture but not the model parameters. The deviation across subjects in results indicates the good generalization of model architecture. If a myoelectric control system is intended to be used for different individuals and conditions, it is a model adaptation problem that needs unsupervised and transfer learning research. Despite the better performance of RCNNs over SVR and CNNs, the results showed the deteriorating decoding accuracy of RCNNs over time. An approach to improving the robustness of model to time variation is employing larger neural networks with more layers and parameters, which requires more training data to prevent overfitting. The extra computation resulting from larger scale of model will affect the training time and system delay. In some studies (19,44), the task learning scheme was used to overcome the time variation problem. Although their model performance was robust for specific grasp tasks, it lacked generalization ability especially for simultaneous and proportional myoelectric control. The EMG-based myoelectric control model is also supposed to be generalized to other organs like hands and legs. A prospective way to the robust performance may be online learning with the unlabeled data collected during everyday practice, which needs further development of unsupervised learning methods for myoelectric control. More types and structures of neural networks are to be studied in order that the training stage would be simplified for individual users and the model would have better adaptive capability.

CONCLUSIONS

In this article, we proposed the recurrent convolutional neural networks model to estimate limb movement with surface EMG signals. The method is based on the deep learning technology combining convolutional neural networks and recurrent neural networks, which is good at representing the features of EMG signals in time and frequency domain. The deep architecture of model is the origin of expressive power. The CNN approach is efficient in extracting the features in frequency domain and robust against noises. The RNN approach can represent the features of time dependencies. The results showed that the RCNN model had higher estimation accuracy and better robustness to time variation than SVR and CNNs. The recurrent approach improves the model performance for regression tasks better than using CNNs alone. Further study of deep architecture is suggested for EMG decoding models.

Acknowledgments: The authors thank the volunteer participants. This work was supported by the National Basic Research Program of China (973 Program) under grant 2011CB707503, the National Natural Science Foundation of China under grant 51475288, 51605302, and the Shanghai Committee of Science and Technology under grant 14142200700.

Conflict of Interest: None.

REFERENCES

1. Farina D, Jiang N, Rehbaum H, et al. The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. *IEEE Trans Neural Syst Rehabil Eng* 2014;22:797–809.
2. Artemiadis PK, Kyriakopoulos KJ. EMG-based control of a robot arm using low-dimensional embeddings. *IEEE Trans Robot* 2010;26:393–8.
3. Leonardis D, Barsotti M, Loconsole C, et al. An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation. *IEEE Trans Haptics* 2015;8:140–51.
4. Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans Biomed Eng* 2003;50:848–54.
5. Ajiboye AB, Weir RFF. A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control. *IEEE Trans Neural Syst Rehabil Eng* 2005;13: 280–91.
6. Wang G, Yan Z, Hu X, Xie H, Wang Z. Classification of surface EMG signals using harmonic wavelet packet transform. *Physiol Meas* 2006;27:1255–67.
7. Hargrove LJ, Li G, Englehart KB, Hudgins BS. Principal components analysis preprocessing for improved classification accuracies in pattern-recognition-based myoelectric control. *IEEE Trans Biomed Eng* 2009;56:1407–14.
8. Fougner A, Scheme E, Chan ADC, Englehart K, Stavdahl Ø. Resolving the limb position effect in myoelectric pattern recognition. *IEEE Trans Neural Syst Rehabil Eng* 2011;19:644–51.
9. He J, Zhang D, Jiang N, Sheng X, Farina D, Zhu X. User adaptation in long-term, open-loop myoelectric training: implications for EMG pattern recognition in prosthesis control. *J Neural Eng* 2015;12:46005.
10. Scheme E, Englehart K. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *J Rehabil Res Dev* 2011;48:643–60.
11. Atzori M, Müller H. Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview. *Front Syst Neurosci* 2015;9:162.
12. Vujaklija I, Farina D, Aszmann OC. New developments in prosthetic arm systems. *Orthop Res Rev* 2016;8:31–9.
13. Jiang N, Englehart KB, Parker PA. Extracting simultaneous and proportional neural control information for multiple-DOF prostheses from the surface electromyographic signal. *IEEE Trans Biomed Eng* 2009;56:1070–80.
14. Jiang N, Vest-Nielsen JL, Muceli S, Farina D. EMG-based simultaneous and proportional estimation of wrist/hand dynamics in uni-Lateral trans-radial amputees. *J Neuroeng Rehabil* 2012;9:42.
15. Muceli S, Jiang N, Farina D. Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms. *IEEE Trans Neural Syst Rehabil Eng* 2014;22:623–33.
16. Tresch MC, Saltiel P, Bizzi E. The construction of movement by the spinal cord. *Nat Neurosci* 1999;2:162–7.
17. Nielsen JLG, Holmgaard S, Jiang N, et al. Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training. *IEEE Trans Biomed Eng* 2011;58:681–8.
18. Artemiadis PK, Kyriakopoulos KJ. A switching regime model for the EMG-based control of a robot arm. *IEEE Trans Syst Man Cybern B Cybern* 2011;41:53–63.
19. Liarokapis MV, Artemiadis PK, Kyriakopoulos KJ, Manolakos ES. A learning scheme for reach to grasp movements: on EMG-based interfaces using task specific motion decoding models. *IEEE J Biomed Health Inform* 2013;17:915–21.
20. Ison M, Artemiadis P. Proportional myoelectric control of robots: muscle synergy development drives performance enhancement, retainment, and generalization. *IEEE Trans Robot* 2015;31:259–68.
21. Ison M, Vujaklija I, Whitsell B, Farina D, Artemiadis P. High-density electromyography and motor skill learning for robust long-term control of a 7-dof robot arm. *IEEE Trans Neural Syst Rehabil Eng* 2016;24:424–33.
22. Scheme E, Englehart K. Training strategies for mitigating the effect of proportional control on classification in pattern recognition based myoelectric control. *J Prosthet Orthot* 2013;25:76–83.
23. Ameri A, Kamavuako EN, Scheme EJ, Englehart KB, Parker PA. Real-time, simultaneous myoelectric control using visual target-based training paradigm. *Biomed Signal Process Contr* 2014;13:8–14.
24. Ameri A, Kamavuako EN, Scheme EJ, Englehart KB, Parker PA. Support vector regression for improved real-time, simultaneous myoelectric control. *IEEE Trans Neural Syst Rehabil Eng* 2014;22:1198–209.
25. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436–44.
26. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Networks* 2015;61:85–117.
27. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012;25:1097–105.
28. Abdel-Hamid O, Mohamed A, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. *IEEE/ACM Trans Audio Speech Lang Process* 2014; 22:1533–45.
29. Sainath TN, Kingsbury B, Saon G, et al. Deep convolutional neural networks for large-scale speech tasks. *Neural Netw* 2015;64:39–48.
30. Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 2014;1:1725–32.
31. Atzori M, Cognolato M, Müller H. Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands. *Front Neurorobot* 2016;10:9.
32. Sutskever I, Vinyals O, Le Q. Sequence to sequence learning with neural networks. *Adv Neural Inf Process Syst* 2014; 4:3104–12.
33. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013;38:6645–9.
34. Donahue J, Hendricks LA, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit* 2015;39:677.

35. Kriegeskorte N. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu Rev Vis Sci* 2015;1:417–46.
36. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw* 1994;5:157–66.
37. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9:1735–80.
38. Gers FA, Schraudolph NN, Schmidhuber J. Learning precise timing with LSTM recurrent networks. *J Mach Learn Res* 2002;3:115–43.
39. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature* 1986; 323:533–6.
40. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res* 2012;13:281–305.
41. Srivastava N, Hinton GE, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 2014;15:1929–58.
42. d'Avella A, Portone A, Fernandez L, Lacquaniti F. Control of fast-reaching movements by muscle synergy combinations. *J Neurosci* 2006;26:7791–810.
43. Dimitrova NA, Dimitrov GV. Interpretation of EMG changes with fatigue: facts, pitfalls, and fallacies. *J Electromyogr Kinesiol* 2003;13:13–36.
44. Corbett EA, Perreault EJ, Körding KP. Decoding with limited neural data: a mixture of time-warped trajectory models for directional reaches. *J Neural Eng* 2012;9:36002.