J. Neural Eng. **16** (2019) 036015 (11pp) https://doi.org/10.1088/1741-2552/ab0e2e

Regression convolutional neural network for improved simultaneous EMG control

Ali Ameri[1], **Mohammad Ali Akhaee**[2], **Erik Scheme**[3] **and Kevin Englehart**[3]

[1] Department of Biomedical Engineering, Shahid Beheshti University of Medical Sciences, Tehran, Iran
[2] Department of Electrical Engineering, University of Tehran, Tehran, Iran
[3] Institute of Biomedical Engineering, University of New Brunswick, Fredericton, Canada

E-mail: ali.ameri@unb.ca

Received 9 December 2018, revised 5 March 2019
Accepted for publication 8 March 2019
Published 16 April 2019

Abstract

Objective. Deep learning models can learn representations of data that extract useful information in order to perform prediction without feature engineering. In this paper, an electromyography (EMG) control scheme with a regression convolutional neural network (CNN) is proposed as a substitute of conventional regression models that use purposefully designed features. *Approach.* The usability of the regression CNN model is validated for the first time, using an online Fitts' law style test with both individual and simultaneous wrist motions. Results were compared to that of a support vector regression-based scheme with a group of widely used extracted features. *Main results.* In spite of the proven efficiency of these well-known features, the CNN-based system outperformed the support vector machine (SVM) based scheme in throughput, due to higher regression accuracies especially with high EMG amplitudes. *Significance.* These results indicate that the CNN model can extract underlying motor control information from EMG signals during single and multiple degree-of-freedom (DoF) tasks. The advantage of regression CNN over classification CNN (studied previously) is that it allows independent and simultaneous control of motions.

Keywords: emg, myoelectric control, machine learning, prostheses, deep learning

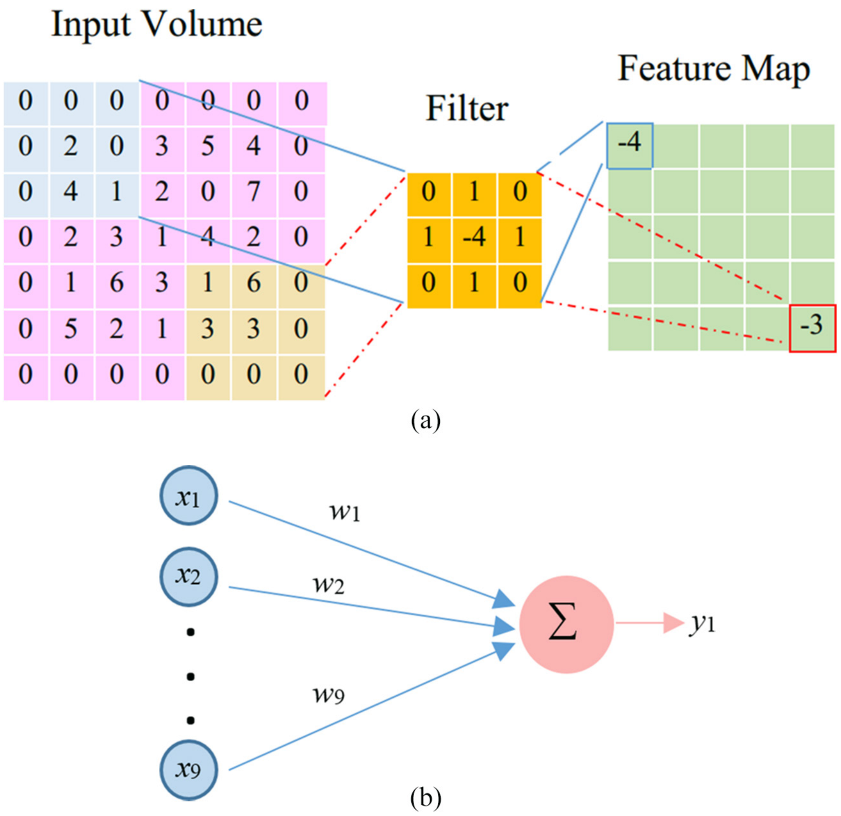
(Some figures may appear in colour only in the online journal)

1. Introduction

Myoelectric control extracts motor intent from electromyography (EMG) signals and translates it into command signals to control prostheses, human–computer interfaces, and robotic exoskeletons. The demand for multi-function dexterous control has motivated the exploration of more advanced EMG control schemes [1]. To this end, pattern recognition techniques have been extensively studied in the past decades [2–11].

Pattern recognition is the process of using machine learning algorithms to discover regularities in data to make data-driven predictions or decisions (e.g. classification) [12]. Due to the stochastic nature of EMG signals, windows of EMG data (typically 100–200 ms), rather than a single instantaneous value, are exploited for pattern recognition [13–15]. However, the raw EMG in these windows present high-dimensional data that have insufficient information density for most machine learning predictors to process. It is therefore necessary to transform the data into reduced representations or features. The selected features should contain useful and discriminatory information about the data for use in prediction. The performance of pattern recognition methods heavily depends on the choice of features [16]. For this reason, much of the effort in pattern recognition-based methods has been devoted to feature design [16]. The process of constructing sets of application-dependent features by a human expert is called feature engineering. The goal of feature engineering is to extract discriminative information from the data. Designing EMG features has been the subject of extensive research and various EMG features containing temporal or spectral information of EMG have been proposed [17].

In order to improve the performance and ease of application, it would be highly desirable to have learning algorithms not dependent on feature engineering. Such algorithms would



0	0	0	0	0	0	0
0	2	0	3	5	4	0
0	4	1	2	0	7	0
0	2	3	1	4	2	0
0	1	6	3	1	6	0
0	5	2	1	3	3	0
0	0	0	0	0	0	0

0	1	0
1	-4	1
0	1	0

Figure 1. Convolution layer: (a) this example shows convolution of an input volume with a 3 × 3 filter. (b) With a 3 × 3 filter, each neuron receives input from only nine neurons of the previous layer.

be expected to learn representations of data that make it easier to extract useful information for conducting a prediction task [16]. The goal is to achieve a more abstract and more useful representation as input to a predictor. This approach is called representation learning or feature learning [16]. One way of learning representations is deep learning methods [18]. This solution conducts learning in terms of a hierarchy of representations, with each of them defined through its relation to simpler representations. The hierarchy of representations allows computers to learn complex features by constructing them from simpler ones [19]. Layer-wise stacking of feature extraction can result in better representations, e.g. in terms of classification accuracy [16]. Different deep learning architectures including deep belief networks, recurrent neural networks, and convolutional neural networks, have been employed with tremendous results in various fields including computer vision, audio recognition, and bioinformatics [20–22].

The convolutional neural network (CNN) [23] is one of the most popular deep learning architectures and is often applied in image analysis. Due to their feature learning capability, CNNs can perform classification or regression from high-dimensional raw data, with no prior feature engineering [16]. Recently, CNNs have been used in a handful of studies [24–30] for EMG-based classification of hand/wrist gestures. In [31], a recurrent CNN was deployed for offline regression-based estimation of upper limb motions from EMG. The advantage of regression-based models over classification-based schemes is that they allow independent simultaneous control of motions or degrees of freedom (DoFs). Simply, this means that multiple DoFs can be active at the same time, and with different magnitudes. Conventional regression-based myoelectric schemes that rely on purposely extracted features have been investigated in previous online studies [32–36]. However, online testing and real-time usability considerations of emerging CNN architectures remain unexplored. In this work, a regression-based CNN is developed for online EMG-based estimation of simultaneous wrist motions. It's usability is validated in an online Fitts' law test [37] in comparison with that of a support vector regression-based scheme with engineered features. Support vector regression has previously been shown to outperform MLP-based regression in myoelectric control [32].

2. Background—convolutional neural network

The convolution layer is the main building block of CNNs and computes the convolution of a set of learnable filters and the 3D (width, height, depth) input data. Every filter is small spatially (along width and height) but extends through the full depth of the input volume. For example, if the input data depth is 16, a typical filter is 3 × 3 × 16. During convolution, the filter slides across the width and height of the input data and performs the dot product at each position. This results in a 2D activation map that yields the response of the filter at every position (figure 1). For a layer with n filters, there will therefore be n separate 2D activation maps, which will be stacked along the depth dimension to produce the output volume. Due to convolution, the resulting output at every position depends only on a local (along width and height) region of the input volume (this region extends through the full depth of the input volume), called the receptive field, and it's size equals the filter size (e.g. 3). In other words, in a convolution layer, each neuron receives input only from its receptive field in the previous layer (figure 1). This substantially reduces the number of free parameters and allows convolution layers to be applied to large data such as images. Conversely, in a fully connected layer, each neuron receives input from all neurons of the previous layer. Hence, it is impractical to apply a fully connected layer to images, due to a very high number of free parameters. For this reason, regular neural networks, known as multilayer perceptron (MLP), cannot be used for processing large data.

After the convolution operation, a downsampling is typically conducted in a pooling layer to reduce the data size to decrease the computation, and to avoid overfitting. For example, a 2×2 average-pooling outputs the average of each spatial block of 2×2 at each position. The pooling is performed independently on every depth slice of the input volume. The extracted feature maps are then sent to activation functions that conduct nonlinear transformations such as rectified linear unit $(relu)\ (x) = \max(x, 0)$, sigmoid or hyperbolic tangent functions. Nonlinear activation functions are important because, otherwise, the network would be a linear predictor, without the ability to learn nonlinear features. *relu* is often used because it does not saturate, hence the network can train much faster, with similar accuracy. After several convolution and pooling layers, the high-level reasoning is performed by a fully connected layer. The final layer is normally the loss layer which determines how training penalizes the error between the predicted and true values, for classification or regression problems.

3. Methods

Ten able-bodied individuals (ages: 31.4 ± 4.1 years, one left-handed, nine right-handed) took part in this study in August, September 2018. All subjects signed written informed consent. The ethics board of Shahid Beheshti Medical University approved the experimental protocol. To familiarize themselves with the experiment, the subjects conducted practice experiments on a different day before the main experiment. Eight bipolar EMG channels were recorded using eight pairs of surface electrodes (g.HiAmp, g-tec Inc.) placed around the dominant forearm proximal to the elbow. A reference electrode was attached on the ulnar styloid of the opposite wrist. The EMG signals were recorded at 1.2 KHz, and filtered between 5–500 Hz and notch filtered at 50 Hz with eighth and fourth-order Butterworth filters, respectively. Data collection and machine learning were performed using Matlab 2017Rb, on a 2.8 GHz quad-core computer with 16 GB RAM.

Eight wrist motions of extension, flexion, supination, pronation, and their simultaneous combinations: extension and supination, extension and pronation, flexion and supination, flexion and pronation were investigated. During the experiment, the subjects conducted a training session followed by online Fitts' law tests.

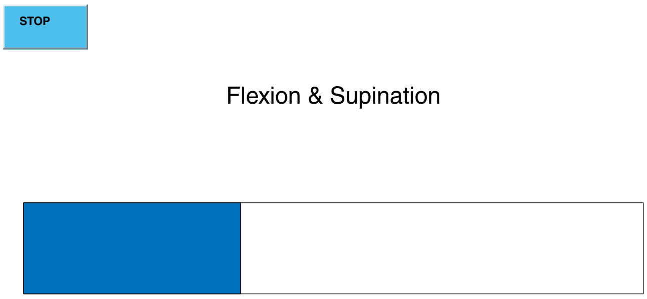


Figure 2. A screenshot of the training interface during the combined flexion and supination training trial. At this moment, the moving progress bar is at one third of the full-range, and the user is expected to perform combined flexion and supination, with the intensity of one third of full-range.

Because this study involved real-time control tests, the system configuration had to be fully determined before conducting the main experiment. Therefore, pilot experiments were performed using three subjects to find the optimum values of various parameters. The configuration of the CNN, SVM and features were determined using offline pilot tests. However, to optimize the online test parameters such as speed gain and threshold, several pilot online tests were performed. These subjects also participated in the main study.

3.1. Training session

The training session comprised a *no motion* trial and eight dynamic trials, corresponding to the motions mentioned above. During the *no motion* trial, subjects were asked to maintain a relaxed posture for 30 s. The dynamic trials lasted 48 s each, while the users were prompted to elicit the trial-specific wrist dynamic motion with an intensity proportional to a progress bar on a computer display (figure 2). The following cycle repeated four times in each dynamic trial: *no motion* (3 s), increasing the contraction intensity to the full-range (a subjective comfortable medium intensity) contraction (3 s), keeping the full-range contraction (3 s), and returning to *no motion* (3 s). During combined motions trials, the subjects were asked to increase or decrease both DoFs simultaneously, with an intensity proportional to the same progress bar.

3.2. Machine learning

Machine learning was performed offline after the training session. First, EMG data segmentation was performed using 167 ms (200 samples) windows with increments of 40 ms (48 samples). To decode the motor intent from the EMG data, two regression approaches based on CNN and SVM were investigated.

For these models, a visual target training strategy [38] was employed so that the normalized position of the progress bar (visual target) was used as the regression target. The regression target was two-dimensional (2D) corresponding to

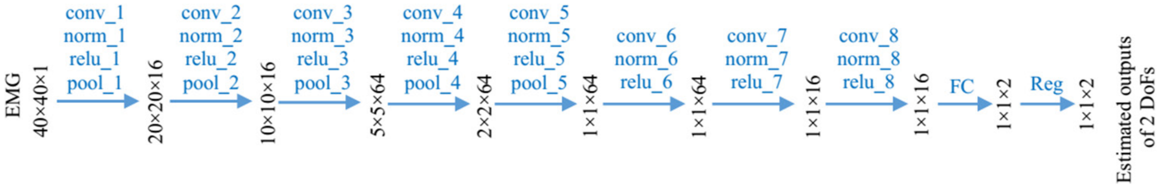


Figure 3. The learned feature maps are presented in the CNN model with convolution (conv), batch normalization (norm), rectified linear units (*relu*), average pooling (pool), fully connected (FC), and regression (Reg) layers. The input is the EMG matrix and the outputs are the estimated values of the two DoFs.

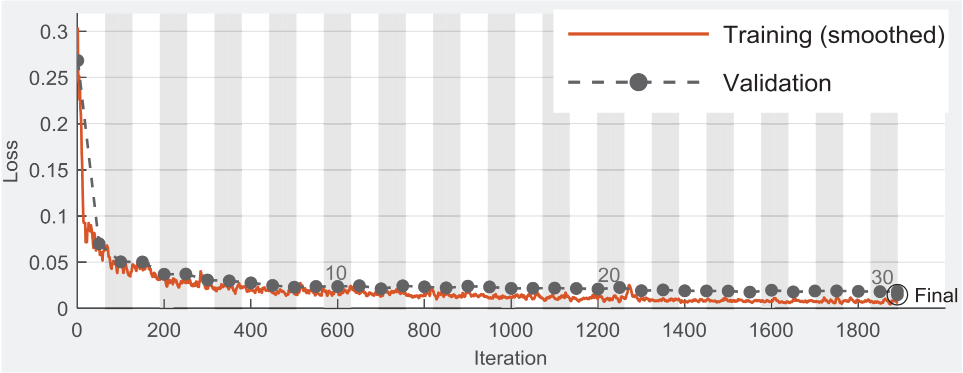


Figure 4. The training and validation error rates during the CNN training epochs are plotted for a representative subject.

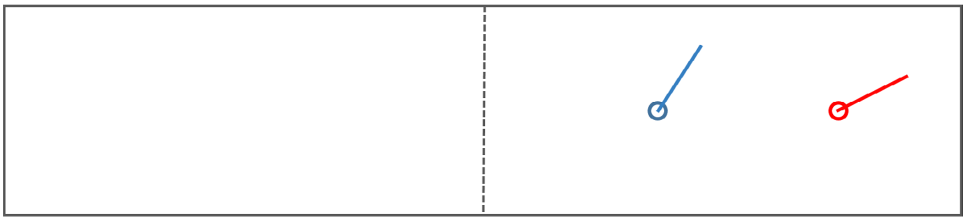


Figure 5. A screenshot of the Fitts' law test interface. The user controls the cursor (blue) to reach the fixed target (red). The cursor can rotate (by rotation of the blue line) and move horizontally (by moving the circle and line together to left and right). The cursor orientation is defined as the angle of the blue line. When a new target appears, the cursor is located at the center of the screen (dotted line) with the blue line aligned vertically upward (zero angle). Flexion-extension moves the cursor horizontally and pronation-supination rotates the cursor. At the moment of this screenshot, the user (right-handed) performs simultaneous extension and supination to reach the target.

the two DoFs: flexion-extension and pronation-supination. Flexion and pronation were denoted by negative values, while extension and supination were considered as positive values. Therefore, the regression target for each DoF was in the range of $[-1, 1]$, where the target magnitude and sign represented the DoF contraction intensity and direction, respectively. For the combined motions training trials, the position of the progress bar was used as the target for both DoFs.

3.2.1. CNN-based model. The CNN-based model was prepared as follows. The model was applied on raw EMG data. First, each EMG segment ($8 \times 200 = 1600$ samples) was arranged to form a 40×40 matrix. A regression CNN was trained from scratch for every subject, using the 40×40 EMG input matrices and the visual target. A single CNN with two outputs (one per DOF) was employed, as pilot studies showed that using a separate CNN for each DOF did not improve the performance (unlike in previously reported MLP-based regression models [32]).

In the pilot studies, the CNN configuration was found empirically as follows. The proposed CNN had 32 layers consisting of an input layer, eight convolution layers, each followed by a normalization and *relu* layers, five average pooling layers, a fully connected layer, and a regression layer (figure 3). This network architecture resembles that of widely used models such as AlexNet [20].

The eight convolution layers had 16, 16, 64, 64, 64, 64, 16, 16 filters, respectively, where all filters were 3×3. The input matrices were zero-padded before convolution. Average pooling was conducted on 2×2 areas with a stride of 2. The model was trained with a *Stochastic Gradient Descent with Momentum (SGDM)* method. An initial learning rate of 0.01 was used, and the learning rate was dropped every 20 periods by a factor of 10.

A minibatch size of 128 was used, as the pilot studies showed that minibatch sizes below 128 resulted in longer training time, while not changing the performance. Also, the minibatch sizes larger than 128 decreased the regression accuracy. Thirty epochs were used as higher values did not enhance the results but, similarly, resulted in longer training time. Fewer epochs were found to reduce the performance.

To avoid overfitting, *L*2 regularization was performed using a regularization factor of 10^{-4}. Also, a validation set

Table 1. For each motion, two different distances and three different widths were used for targets which resulted in six levels of difficulty.

Distance	Width	I.D.
1	0.10	1.80
1	0.15	1.57
1	0.25	1.32
0.5	0.10	0.92
0.5	0.15	0.75
0.5	0.25	0.59

Table 2. The offline and online performance metrics averaged across all subjects are shown using mean ± standard deviation ($n = 10$), along with the paired samples t-tests results including p-values, t-values, degrees of freedom (df), and standard deviation of the difference between the two groups (sd).

		CNN	SVM	p-value	t-value	df	sd
Regression error (%)	Fle/ext	5.8 ± 0.6	7.5 ± 0.8	3×10^{-7}	-13.6	9	0.4
	Pro/sup	8.6 ± 1.1	11.3 ± 1.5	8×10^{-7}	-11.9	9	0.7
Regression error (%) (low amp)	Fle/ext	3.9 ± 0.6	4.4 ± 0.7	8×10^{-5}	-6.7	9	0.2
	Pro/sup	5.6 ± 0.4	6.0 ± 0.5	0.01	-3.1	9	0.4
Regression error (%) (high amp)	Fle/ext	8.9 ± 1.4	12.9 ± 1.8	8×10^{-7}	-12.0	9	1.1
	Pro/sup	13.7 ± 3.1	20.4 ± 3.9	1×10^{-6}	-11.4	9	1.8
Output magnitude (%) (low amp)	Fle/ext	5.4 ± 0.5	5.3 ± 0.5	0.11	1.8	9	0.2
	Pro/sup	6.6 ± 0.5	6.4 ± 0.6	0.17	1.5	9	0.4
Output magnitude (%) (high amp)	Fle/ext	79.3 ± 1.2	74.2 ± 1.6	2×10^{-8}	18.8	9	0.9
	Pro/sup	73.9 ± 2.6	66.2 ± 3.4	4×10^{-7}	13.2	9	1.9
Throughput		0.48 ± 0.02	0.41 ± 0.01	2×10^{-6}	11.0	9	1.9
Overshoot		0.09 ± 0.06	0.06 ± 0.04	0.12	1.7	9	0.05
Path eff (%)		96.6 ± 1.9	97.6 ± 1.2	0.12	-1.7	9	1.8
Average speed (%)		84.3 ± 2.3	71.2 ± 2.2	2×10^{-9}	23.2	9	1.8

comprising randomly sampled 10% of the data was used to validate the network, once per epoch, during the network training. Both the training and validation data were shuffled before each epoch. An NvidiaGTX 1050 graphics processing unit (GPU), was used for training the CNN. The training and validation error rates during the CNN training epochs are plotted for a representative subject (figure 4).

3.2.2. SVM-based model. The support vector regression-based model was implemented as follows. For this model, prior feature extraction was necessary as SVMs cannot be applied directly to raw EMG data. Five features comprising the well-known TD set [39] and mean frequency were computed for each EMG window. The mean frequency was included because it improved the performance in pilot studies. Autoregressive coefficients [17], wavelet marginals [40], and Willison amplitude [17] were also investigated in pilot studies, but they did not result in improvements, hence they were not included.

Since SVMs do not allow multi-output regression, one SVM for each DoF was used. For each DoF, the extracted features were used as the inputs to train an SVM using ν-support vector regression algorithm. The visual target provided during training was again used as the regression target. The Kernel function of radial basis, $\nu = 0.5$, $\gamma = \text{number of features}^{-1} = \frac{1}{40}$, and *cost* parameter of 0.5, were used. These parameters were determined empirically. The libSVM [41] Matlab package was used for simulation.

3.3. Fitts' law test

The real-time control performance of the proposed models was assessed using a 2D Fitts' law style target acquisition test. A fixed target appeared on a computer display and the subjects were asked to move a cursor to acquire the target as quickly as possible. In order to reach a target, the cursor orientation and horizontal position had to be matched to those of the target.

Velocity Control was used, hence the regressor outputs in the flexion-extension and pronation-supination DOFs, were mapped to the cursor horizontal and orientational velocities, respectively. For right-handed users, moving the cursor to left and right was performed using flexion and extension, respectively. Rotation of the cursor orientation counter-clockwise and clockwise was achieved by pronation and supination, respectively (figure 5). For left-handed users, reverse directions were used. The horizontal and orientational speeds of the cursor were calculated using equation (1).

$$|v| = g \cdot \frac{|x| - h}{1 - h} \qquad (1)$$

where x is the regressor output in the corresponding DoF, and h is a threshold used to prevent unintentional cursor movements

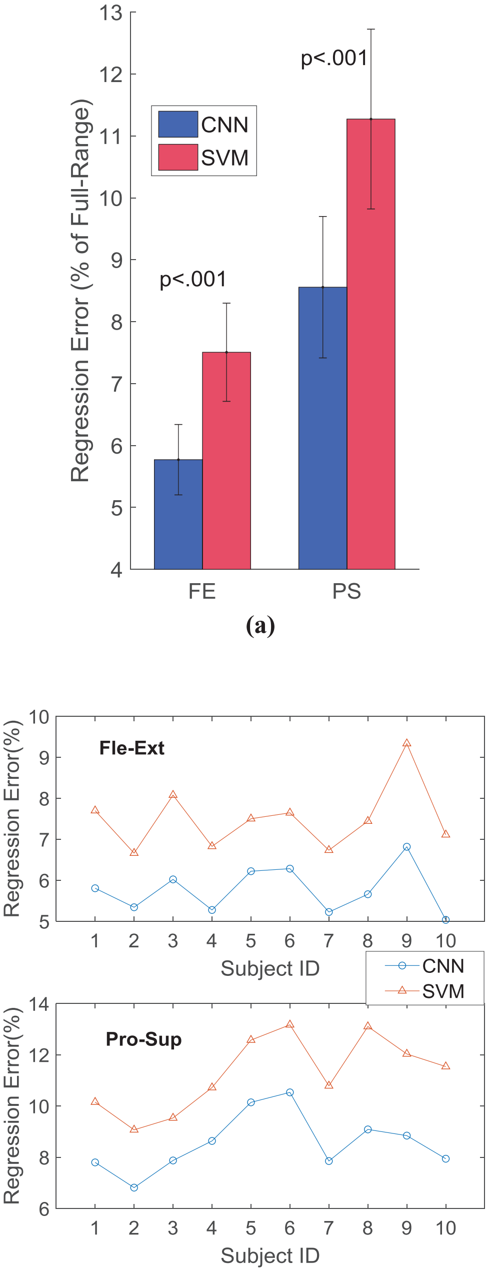


Figure 6. The offline regression error (percentage of full-range) for each DoF representing (a) mean and standard deviation across all subjects ($n = 10$), along with the paired samples t-tests p-values for each DoF (b) individual subjects.

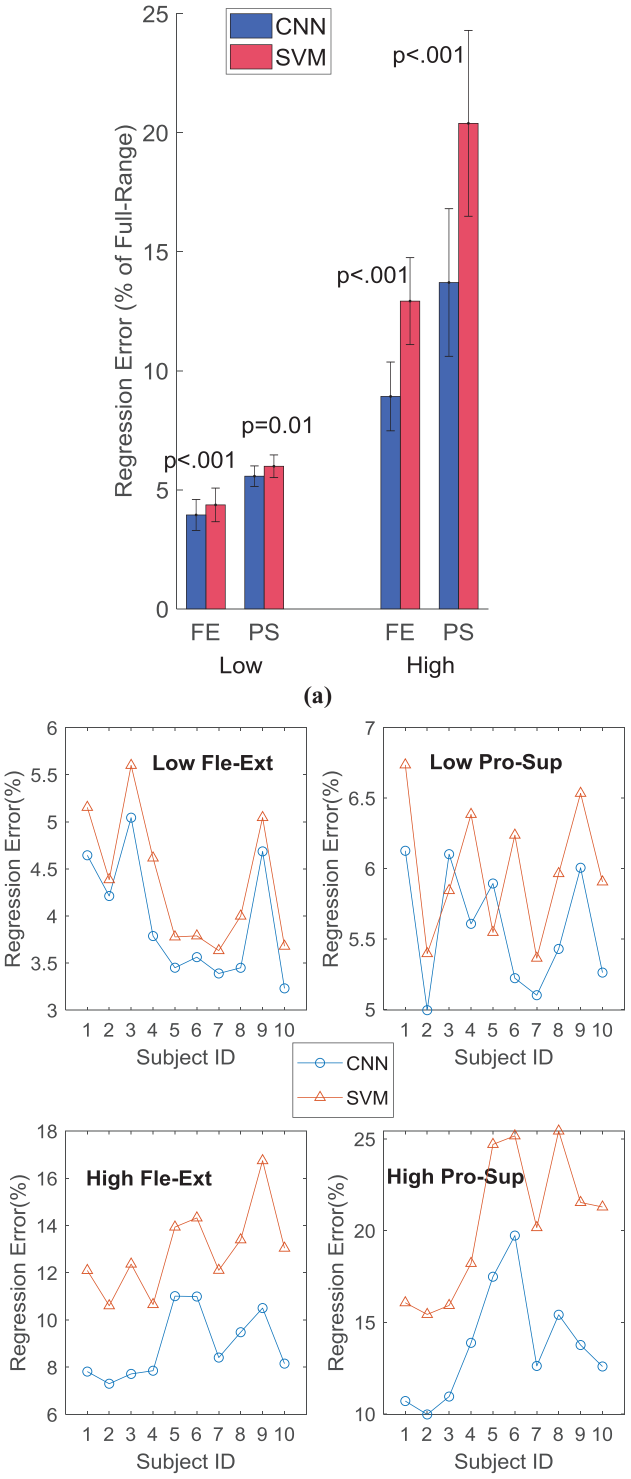


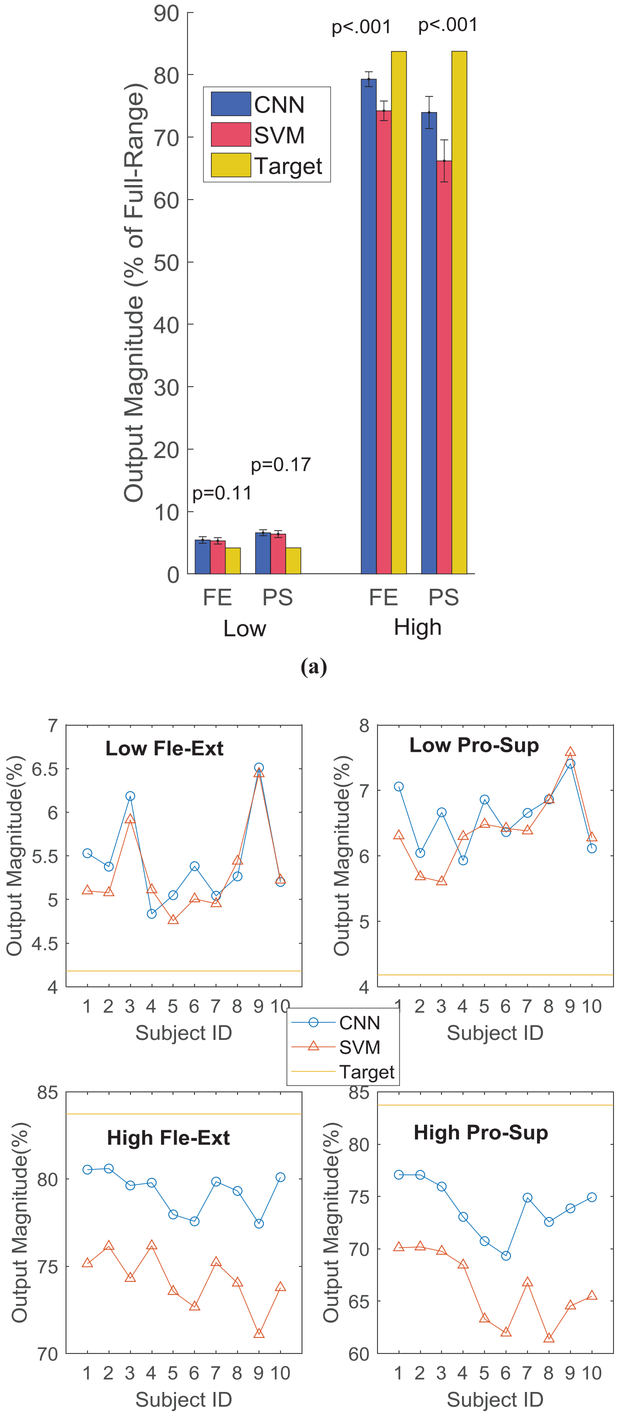
Figure 7. The offline mean regression error (percentage of full-range) for each DoF is shown for low and high amplitude contractions representing (a) mean and standard deviation across all subjects ($n = 10$), along with the paired samples t-tests p-values in each case (b) individual subjects.

and was set to 0.25 (determined empirically). This method results in better control in low speeds [42]. The speed gain (g) of 0.36 units per second was used, as this value resulted in the optimum speed-accuracy tradeoff. Speeds greater than g were clipped to g.

During the Fitts' law test, targets corresponding to the eight motions were presented, with six levels of difficulty (table 1), producing a total of $8 \times 6 = 48$ targets, for each control scheme.

A Shannon–Welford [43, 44] formulation of the *index of difficulty* (y) was employed which separates the effects of target width (w) and distance (d) to improve prediction accuracy (equation (2)). For $k = 1$, this method turns into Shannon form.

$$y = \log_2(\frac{w+d}{w^k}) \tag{2}$$



(a)

(b)

Figure 8. The offline output mean magnitude (percentage of full-range) for each DoF is shown for low and high amplitude contractions, along with the target mean magnitude representing (a) mean and standard deviation across all subjects ($n = 10$), along with the paired samples t-tests p-values in each case (b) individual subjects.

where k was set to 0.5 as the pilot studies showed that this value resulted in a maximum linearity of movement time versus index of difficulty.

Considering the cursor horizontal position and orientation as the two Cartesian dimensions, if the cursor was kept in the 2D Euclidean distance of $w/2$ of the target center for a dwell time of 1 s, the target was successfully achieved. If the cursor was within the target width, the cursor color changed as feedback. The maximum time allowed for each target acquisition was 15 s. The task was reported failure when the target was not achieved within 15 s.

The targets were presented in a random order, in four blocks of 12 trials (targets). Each test block was run with one control scheme, selected randomly, and then was repeated with the other scheme. Subsequently, it was proceeded to the next test block. The control scheme was switched after each block (not each trial), because it allows the users to learn and compensate for that method during the block. The control performance was evaluated using four metrics [45] of completion rate, throughput, overshoot, and path efficiency.

The online computation time of the CNN, SVM-based models were 6 ms, and 13 ms, respectively. In order to compare the control performance, all control schemes must have the same real-time delay. For this purpose, a single real-time code was implemented for both methods. Using this code, the screen was updated every 42 ms (sum of the processing times of the two methods as well as other processing delays such as plotting) to display the cursor new position and orientation.

3.4. Statistics

To test if the results have normal distributions, a Shapiro–Wilk test was conducted for every performance metric of the CNN and SVM-based methods, separately. In all cases $p > 0.05$ was obtained, hence normal distributions can be assumed. Subsequently, for every metric, a *paired samples t-test* was performed to compare the performance between the CNN and SVM-based methods. A significance level of 0.05 was used.

4. Results

The full results of the paired samples t-tests conducted on the offline and online performance metrics to compare the CNN and SVM-based methods, are listed in table 2. The p-values are also presented along with the bar plots.

4.1. Offline results

The offline regression error for each DoF, using a four-fold cross-validation is shown in figure 6, along with the paired samples t-tests p-values (the offline outputs were lowpass filtered using a third order Butterworth filter with a cutoff frequency of 1 Hz). The results show that the CNN-based model outperformed the SVM-based model in both DoFs.

For further investigation, the offline regression error for low (*target amplitude* < 0.4) and high (*target amplitude* $\geqslant 0.4$) amplitudes were computed, separately (figure 7). The target amplitude refers to the length of the progress bar (figure 2) in the training trials. For example, high amplitude contractions for a given DoF represents contractions that activates

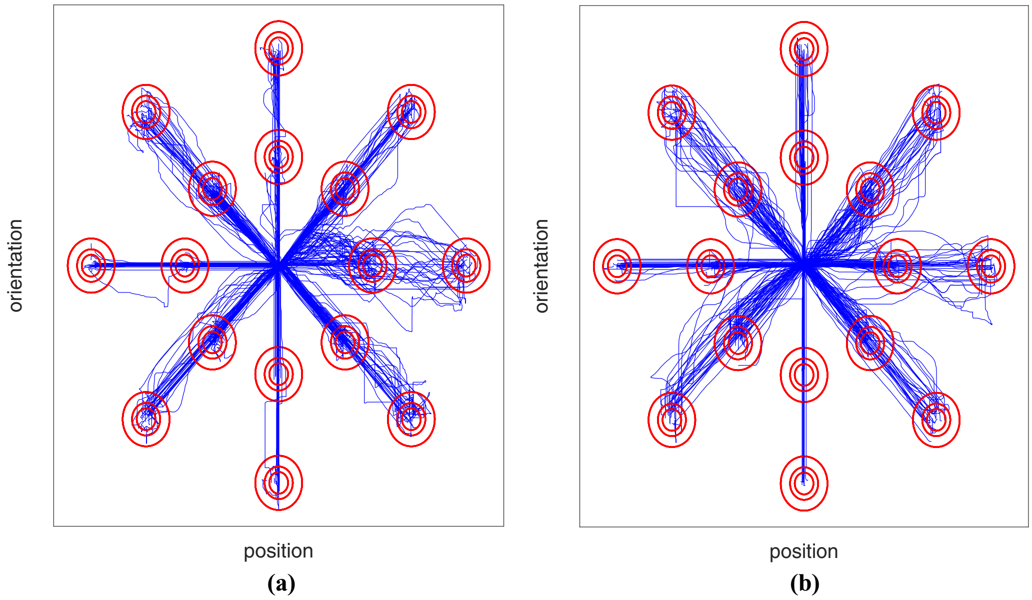


Figure 9. An overlay of all users' path traces during the Fitts' law test for the (a) CNN and (b) SVM-based control systems. The cursor horizontal position (flexion-extension) is mapped to the x-axis and the cursor orientation (pronation-supination) is mapped to the y-axis. Each of the eight motions is represented with six targets with three different widths and two different distances.

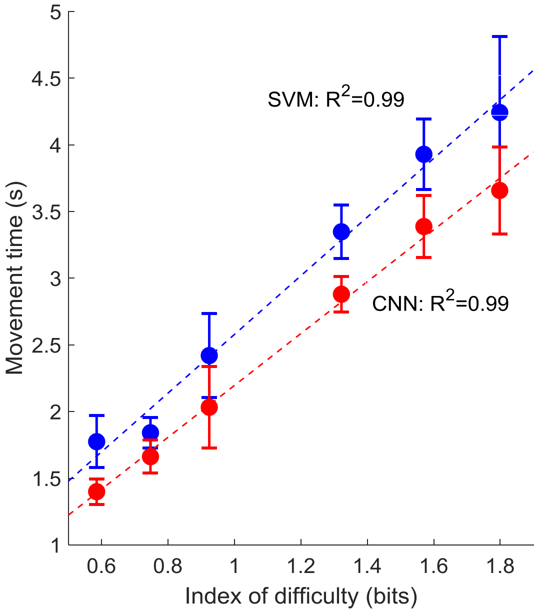


Figure 10. For each level of difficulty, the mean target acquisition time, averaged across all subjects, with standard deviations ($n = 10$), are plotted for each control scheme along with the best linear fit. High R^2 values for each method shows high linearity of movement time versus index of difficulty.

this DoF, while the length of the progress bar is above 40% of full-range. The results show that the CNN-based scheme outperformed the SVM-based scheme during both low and high amplitude contractions.

The offline average magnitude output of each model was also computed for low and high amplitudes, separately (figure 8). The results show that no difference was found between the two models during low amplitudes, but again, the output magnitude of the CNN-based scheme was higher than that of the SVM-based scheme during high amplitude contractions. For comparison, the target average magnitudes are also shown in figure 8.

4.2. Fitts' law test results

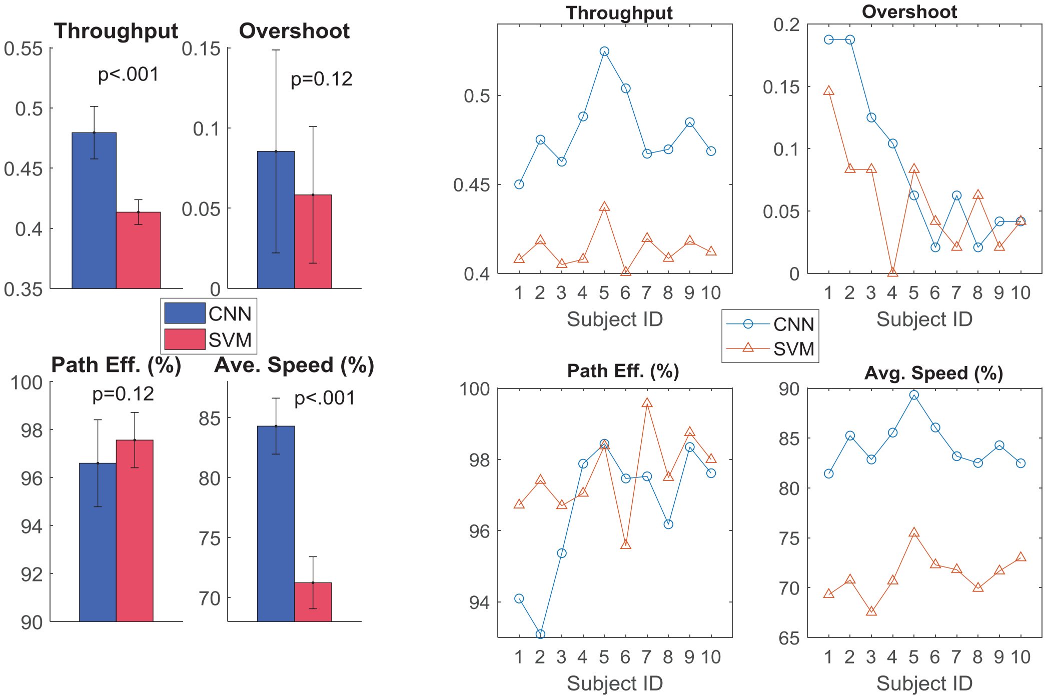
A 2D illustration of the overlay of all users' path traces during the Fitts' law test for each control scheme is shown in figure 9. Figure 10 displays a plot of the average target acquisition time versus ID, along with the best linear fit. The linearity was determined using the coefficient of determination (R^2) (equation (3)).

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_{est}(i) - y(i)\right)^2}{\sum_{i=1}^{N} \left(y(i) - \bar{y}\right)^2} \tag{3}$$

where $N = 6$, and y and y_{est} are the measured and estimated (by the linear fit) average movement time for each level of difficulty, respectively. Also, $\bar{y}$ is the average movement time across all levels of difficulty.

The results complied with Fitts' law, as the acquisition time versus index of difficulty for both control schemes were strongly linear ($R^2 > 0.98$).

Figure 11 presents the Fitts' law test performance metrics as well as the cursor average speed (percentage of full-speed) for each method. The completion rate for both methods were 100%, hence it is not included in the graphs. The results show that the throughput of the CNN-based regression model was significantly higher than that of the SVM-based regression model. However, no difference was found between the two control schemes in the other metrics i.e. completion rate, overshoot, and path efficiency. Also, the average cursor speed of the CNN-based control was higher than that of the SVM-based control.



(a) (b)

Figure 11. The Fitts' law test metrics representing (a) mean and standard deviation across all subjects ($n = 10$), along with the paired samples t-tests p-values in each case (b) individual subjects.

5. Discussion

In this study, the usability of a regression CNN-based EMG control scheme was validated in an online control environment and the usability was compared to that of an SVM-based regression scheme. Based on the offline results (figure 7), during high amplitude contractions, the CNN model had higher regression accuracy, hence the CNN mean output magnitude was closer to the mean magnitude of the target (figure 8). This resulted in more accurate control during high amplitude contractions in the online CNN-based control test, and consequently higher average cursor speeds, and higher throughput (figure 11). Independently, users subjectively noted that the cursor speed of the CNN-based control appeared to be slightly faster.

Both models showed high control performance as indicated in the Fitts' law test results and the path traces in figure 9. However, occasionally inadvertent cursor movements occurred caused by false simultaneous activation of one DoF due to the activation of the other DoF. This issue degrades controllability especially when the intended magnitude of a DoF is low or zero, but the estimated magnitude is higher. For example, during pure extension by the user, the cursor sometimes exhibited unwanted rotation which causes frustration, lower efficiency, and degrades all control performance metrics. This type of error can be tied to the regression accuracies for each DoF during low amplitude contractions. Although the offline regression error results (figure 7) show that the CNN outperformed during low amplitudes, but the difference between the two methods was small ($\approx 0.5\%$). This is the main reason for the similar performance of the two models in overshoot, path efficiency, and completion rate, as shown in the Fitts' law test results.

The literature on EMG control schemes primarily relies on feature engineering. Although, many features have been identified with successful characterization of EMG in a discriminative fashion to perform accurate predictions, they have yet to satisfactorily address many of the challenges with the robustness of myoelectric control such as limb position, and electrode shift. Future work could investigate application of CNNs to improve the robustness to these dynamic factors.

The major advantage of the CNN model is its feature learning ability which eliminates the need for feature engineering. This allows the CNN to operate on raw EMG data, resulting in a significant ease of application. However, designing CNNs is more complex than conventional models such as SVMs, due to a larger number of parameters needed to be manually determined, such as the choice of layers, their numbers, and the number of filters in convolution layers.

The results indicate that the CNN was capable of extracting underlying motor information in EMG data. An EMG signal is a superposition of motor unit action potentials (MUAPs). As a muscle contraction level increases, the force is regulated by a combination of an increasing firing rate of MUAPs and a larger number of motor units recruited. This will impact magnitude, frequency distribution and the structural complexity of the EMG signal. The complex interaction between firing rate and motor unit recruitment has made feature engineering a challenging task for EMG pattern recognition, requiring both temporal and spectral content of EMG data to capture motor information.

The engineered features used for the SVM-based model incorporated both temporal (mean-absolute value, waveform length) and spectral (waveform length, zero-crossings, slop sign changes, mean frequency) information of the EMG signals. While these cover many of the desirable known signal characteristics [46], it is possible that additional features could improve the performance of the SVM system. Indeed, it is not yet known how the accumulation of all known EMG features compares to those learned by automated deep learning systems. Nevertheless, the high performance of the CNN model indicates that it was able to learn the desired temporal and spectral representations of the EMG data needed to accurately estimate the motor intent. These representations were learned through successive convolution layers.

The training session duration was 7 min, resulting in approximately 10 000 training samples (for offline analysis, 75% of these data were used for training). The superior performance of the CNN model over the SVM model demonstrates the CNN efficacy even with a modest amount of training data. In general, accurate predictions, especially with deep learning algorithms (due to high number of parameters), require sufficiently large amount of training samples. However, recording more data by extending the training session is not realistic. To address this challenge, transfer learning [29] can be employed to leverage inter-subject data by pre-training a model on multiple subjects before training it on a new user. Transfer learning [29] has shown to improve the performance of CNNs while requiring less amount of training data.

Both CNN and SVM-based models are computationally efficient, hence the real-time control was not deteriorated by the processing delay. The prediction processing time of the CNN was 6 ms, where the SVM-based scheme 13 ms (6 ms for each SVM and 1 ms for feature extraction). This rapid processing of the CNN model suggest that it could be suitable for implementation in myoelectric prostheses.

6. Conclusion

A regression CNN-based EMG control system was proposed and its usability was validated in an online Fitts' law style test for the first time. The control quality metrics were compared to those of a support vector regression-based scheme with the extraction of engineered features (combined TD set and mean frequency). The CNN-based system outperformed the SVM-based scheme in throughput, due to higher estimation accuracies especially during higher amplitude contractions. The high performance of the CNN model shows its ability to learn underlying motor information from EMG data.

Future work could study the use of CNN-based architectures to EMG data in combination with additional sensors including accelerometer data and force myography for further performance improvement. Moreover, the robustness of CNN's to dynamic factors such as electrode shift and limb position could be investigated.

Acknowledgment

We would like to thank the national brain mapping laboratory for their support.

ORCID iDs

Ali Ameri https://orcid.org/0000-0001-6518-3184
Kevin Englehart https://orcid.org/0000-0003-4525-1121

References

[1] Farina D, Jiang N, Rehbaum H, Holobar A, Graimann B, Dietl H and Aszmann O C 2014 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.* **22** 797–809

[2] Castellini C and van der Smagt P 2009 Surface EMG in advanced hand prosthetics *Biol. Cybern.* **100** 35–47

[3] Simon A M, Hargrove L J, Lock B A and Kuiken T A 2011 The target achievement control test: evaluating real-time myoelectric pattern recognition control of a multifunctional upper-limb prosthesis *J. Rehabil. Res. Dev.* **48** 619

[4] Clancy E A, Liu L, Liu P and Moyer D V Z 2012 Identification of constant-posture EMG–torque relationship about the elbow using nonlinear dynamic models *IEEE Trans. Biomed. Eng.* **59** 205–12

[5] Jiang N, Vest-Nielsen J L, Muceli S and Farina D 2012 EMG-based simultaneous and proportional estimation of wrist/hand kinematics in uni-lateral trans-radial amputees *J. Neuroeng. Rehabil.* **9** 42

[6] Muceli S and Farina D 2012 Simultaneous and proportional estimation of hand kinematics from EMG during mirrored movements at multiple degrees-of-freedom *IEEE Trans. Neural Syst. Rehabil. Eng.* **20** 371–8

[7] Hargrove L J, Lock B A and Simon A M 2013 Pattern recognition control outperforms conventional myoelectric control in upper limb patients with targeted muscle reinnervation *2013 35th Annual Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)* (IEEE) pp 1599–602

[8] Scheme E and Englehart K 2013 Training strategies for mitigating the effect of proportional control on classification in pattern recognition based myoelectric control *J. Prosthet. Orthot.* **25** 76

[9] Young A J, Smith L H, Rouse E J and Hargrove L J 2013 Classification of simultaneous movements using surface EMG pattern recognition *IEEE Trans. Biomed. Eng.* **60** 1250–8

[10] Ameri A, Scheme E J, Englehart K B and Parker P A 2014 Bagged regression trees for simultaneous myoelectric force estimation *2014 22nd Iranian Conf. on Electrical Engineering* (IEEE) pp 2000–3

[11] Ameri A, Scheme E J, Kamavuako E N, Englehart K B and Parker P A 2014 Real-time, simultaneous myoelectric control using force and position-based training paradigms *IEEE Trans. Biomed. Eng.* **61** 279–87

[12] Bishop C M 2016 *Pattern Recognition and Machine Learning* (New York: Springer)

[13] Englehart K and Hudgins B 2003 A robust, real-time control scheme for multifunction myoelectric control *IEEE Trans. Biomed. Eng.* **50** 848–54

[14] Farrell T R and Weir R F 2007 The optimal controller delay for myoelectric prostheses *IEEE Trans. Neural Syst. Rehabil. Eng.* **15** 111–8

[15] Oskoei M A and Hu H 2007 Myoelectric control systems—a survey *Biomed. Signal Process. Control* **2** 275–94

[16] Bengio Y, Courville A and Vincent P 2013 Representation learning: a review and new perspectives *IEEE Trans Pattern Anal. Mach. Intell.* **35** 1798–828

[17] Phinyomark A, Phukpattaranont P and Limsakul C 2012 Feature reduction and selection for EMG signal classification *Expert Syst. Appl.* **39** 7420–31

[18] Phinyomark A and Scheme E 2018 EMG pattern recognition in the era of big data and deep learning *Big Data Cogn. Comput.* **2** 21

[19] Goodfellow I, Bengio Y, Courville A and Bengio Y 2016 *Deep Learning* (Cambridge, MA: MIT Press)

[20] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Adv. Neural Inf. Process. Syst.* **25** 1097–105

[21] Abdel-Hamid O, Mohamed A-R, Jiang H, Deng L, Penn G and Yu D 2014 Convolutional neural networks for speech recognition *IEEE/ACM Trans. Audio Speech Lang. Process.* **22** 1533–45

[22] Choi E, Schuetz A, Stewart W F and Sun J 2016 Using recurrent neural network models for early detection of heart failure onset *J. Am. Med. Inform. Assoc.* **24** 361–70

[23] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324

[24] Allard U C, Nougarou F, Fall C L, Giguère P, Gosselin C, Laviolette F and Gosselin B 2016 A convolutional neural network for robotic arm guidance using semg based frequency-features *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (IEEE) pp 2464–70

[25] Atzori M, Cognolato M and Müller H 2016 Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands *Frontiers Neurorobot.* **10** 9

[26] Geng W, Du Y, Jin W, Wei W, Hu Y and Li J 2016 Gesture recognition by instantaneous surface EMG images *Sci. Rep.* **6** 36571

[27] Park K-H and Lee S-W 2016 Movement intention decoding based on deep learning for multiuser myoelectric interfaces *2016 4th Int. Winter Conf. on Brain–Computer Interface (BCI)* (IEEE) pp 1–2

[28] Ameri A, Akhaee M A, Scheme E and Englehart K 2018 Real-time, simultaneous myoelectric control using a convolutional neural network *PLoS One* **13** e0203835

[29] Côté-Allard U, Fall C L, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, Laviolette F and Gosselin B 2018 Deep learning for electromyographic hand gesture signal classification using transfer learning *IEEE Trans. Neural Syst. Rehabil. Eng.* (https://doi.org/10.1109/TNSRE.2019.2896269)

[30] Zia ur Rehman M, Waris A, Gilani S, Jochumsen M, Niazi I, Jamil M, Farina D and Kamavuako E 2018 Multiday EMG-based classification of hand motions with deep learning techniques *Sensors* **18** 2497

[31] Xia P, Hu J and Peng Y 2018 EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks *Artif. Organs* **42** E67–77

[32] Ameri A, Kamavuako E N, Scheme E J, Englehart K B and Parker P A 2014 Support vector regression for improved real-time, simultaneous myoelectric control *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 1198–209

[33] Hahne J M, Biessmann F, Jiang N, Rehbaum H, Farina D, Meinecke F, Müller K-R and Parra L 2014 Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 269–79

[34] Jiang N, Rehbaum H, Vujaklija I, Graimann B and Farina D 2014 Intuitive, online, simultaneous, and proportional myoelectric control over two degrees-of-freedom in upper limb amputees *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 501–10

[35] Kamavuako E N, Scheme E J and Englehart K B 2014 Combined surface and intramuscular EMG for improved real-time myoelectric control performance *Biomed. Signal Process. Control* **10** 102–7

[36] Amsuess S, Vujaklija I, Goebel P, Roche A D, Graimann B, Aszmann O C and Farina D 2016 Context-dependent upper limb prosthesis control for natural and robust use *IEEE Trans. Neural Syst. Rehabil. Eng.* **24** 744–53

[37] Fitts P M 1954 The information capacity of the human motor system in controlling the amplitude of movement *J. Exp. Psychol.* **47** 381

[38] Ameri A, Kamavuako E N, Scheme E J, Englehart K B and Parker P A 2014 Real-time, simultaneous myoelectric control using visual target-based training paradigm *Biomed. Signal Process. Control* **13** 8–14

[39] Hudgins B, Parker P and Scott R N 1993 A new strategy for multifunction myoelectric control *IEEE Trans. Biomed. Eng.* **40** 82–94

[40] Nielsen J L, Holmgaard S, Jiang N, Englehart K B, Farina D and Parker P A 2011 Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training *IEEE Trans. Biomed. Eng.* **58** 681–8

[41] Chang C-C and Lin C-J 2011 LIBSVM: a library for support vector machines *ACM Trans. Intell. Syst. Technol.* **2** 27

[42] Scheme E, Lock B, Hargrove L, Hill W, Kuruganti U and Englehart K 2014 Motion normalized proportional control for improved pattern recognition-based myoelectric control *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 149–57

[43] Welford A T 1968 *Fundamentals of Skill* (London: Methuen)

[44] Shoemaker G, Tsukitani T, Kitamura Y and Booth K S 2012 Two-part models capture the impact of gain on pointing performance *ACM Trans. Comput-Hum. Interact.* **19** 28

[45] Williams M R and Kirsch R F 2008 Evaluation of head orientation and neck muscle EMG signals as command inputs to a human–computer interface for individuals with high tetraplegia *IEEE Trans. Neural Syst. Rehabil. Eng.* **16** 485–96

[46] Phinyomark A, Khushaba R N, Ibánez-Marcelo E, Patania A, Scheme E and Petri G 2017 Navigating features: a topologically informed chart of electromyographic features space *J. R. Soc. Interface* **14** 20170734