DEEP LEARNING-BASED ROBUST NEURAL-MACHINE INTERFACE FOR DEXTEROUS CONTROL OF ROBOTIC HAND

Feng Xu

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Biomedical Engineering

Chapel Hill
2021

Approved by:

Derek Kamper

Nitin Sharma

Katherine Saul

ABSTRACT

Feng Xu: Deep Learning-Based Robust Neural-Machine Interface for
Dexterous Control of Robotic Hand
(Under the direction of Xiaogang Hu)

Neuromuscular injuries can impair hand function and impact the quality of life. To restore hand dexterity, numerous assistive devices have been developed. However, the lack of a robust neural-machine interface may limit functionality of these devices. Accordingly, a robust neural decoding approach was developed that can continuously decode the intended finger motor output. High-density electromyogram (HD-EMG) signals were obtained from the extrinsic finger flexor and extensor muscles. Convolutional neural networks were implemented to learn the mapping from HD-EMG features to finger-specific population neuron firing frequency, which was then used to control a prosthetic hand in real-time. In comparison with the HD-EMG amplitude approach, the network-based decoder predicted finger forces and angles with lower prediction errors. The network-based decoder also demonstrated better isolation with minimal predicted output in the unintended fingers. The outcomes offer a novel neural-machine interface technique that allows intuitive control of assistive robotic hands in a dexterous manner.

To my mentor and friend, I couldn't have done this without you. Thank you for all of your support along the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Our ability to control individual fingers independently allows us to perform a wide variety of complex tasks, such as cooking, dressing, and typing. The human digits are capable of performing precise, coordinated movements with little conscious effort. The hand is crucial to overall upper limb function, as evidenced by the equating of loss of the hand (e.g., hand amputation) with a 90% loss of total upper limb function and 54% loss of the whole body function [1]. The importance of hand function to daily life is further underscored by results of a survey taken by individuals with tetraplegia, affecting both upper and lower limbs. Responders listed regaining hand function as their top priority, well above walking [2]. To restore impaired or lost hand function, the design of assistive devices, such as prosthetic hands or exoskeleton gloves, have advanced to a degree that can imitate movements of the human biological hand [3, 4, 5, 6]. However, clinical translation of these robotic devices has been limited. One major hurdle limiting widespread use of robotic hands is the lack of a robust neural-machine interface that can reliably translate the user's intent into executable control commands for the devices. Although different neural recording modalities have been developed to acquire brain signals for the detection of motor intent [7, 8], an accurate decoding of individual finger movements remains challenging. For individuals with voluntary muscle activation capability, surface electromyography (EMG), a method of recording the electrical signals arising from muscle activation, affords an attractive means of capturing motor intent, as evidenced by its widespread usage in the control of wearable robots [9, 10]. Typically, EMG features are extracted to identify the desired movement from a finite set of movements [11, 12, 13, 14, 15] or the movement of a specific joint is made proportional to the EMG feature (with EMG amplitude as a commonly used feature) [16, 17]. These global EMG-based control strategies are prone to interference. For example, the macro EMG absolute amplitude can be biased by intrinsic physiological factors, such as the varying distances and tissue types located between the muscle source and the recording electrode on the surface of the skin. EMG signals can also be biased by crosstalk of neighboring muscles (especially in the case of small hand muscles lying in close proximity), inconsistent electrode-skin contact, and motion artifacts [18, 19]. All of these factors can limit the robustness and efficacy of the human-robot interaction. Alternatively, it is possible to use spinal motoneuron firing activity (binary events) as a neural interface. Although individual motoneurons have distinct firing properties [20, 21] that may not directly reflect the descending command, the frequency/probability of the spinal motoneuron firing at the population level can directly reflect high-level neural drive from the brain to the muscles. Essentially, the spinal cord output signal (motoneuronal firing events) is used to decode the spinal input signal (descending command from the brain). This decoded neural drive has the potential to be much more robust for decoding user intent than global EMG patterns as the binary discrimination is less susceptible to noise than measuring analog features.

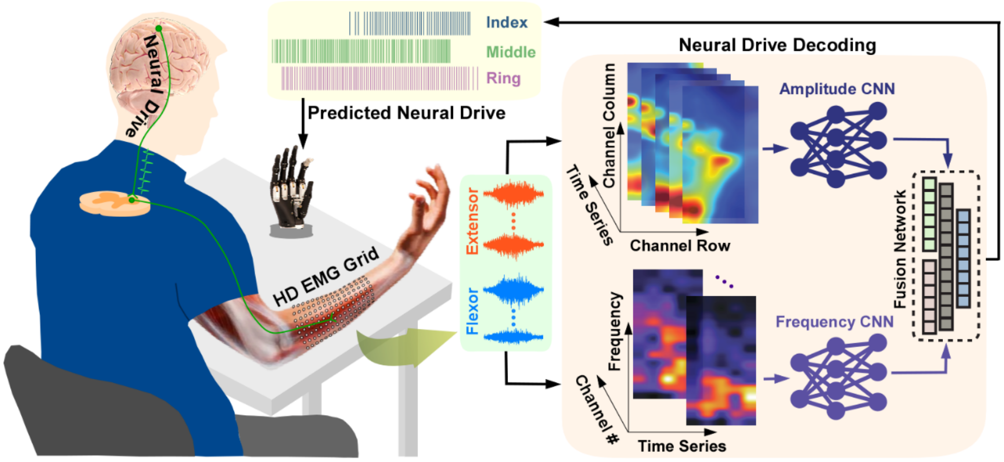

Figure 1.1: Overview of the research approach. High-density electromyographic (HD-EMG) signals were obtained from the extrinsic finger flexor and extensor muscles. Convolutional neural network (CNN)-based models were used to learn the mapping from HD-EMG amplitude and frequency features to finger-specific neural drive signals (i.e., populational motoneuron firing frequency). The predicted neural drive signals were then used to continuously control the fingertip forces or joint angles of the index, middle, and ring fingers of a prosthetic hand in a dexterous manner.

Currently, motoneuron firing events are extracted through motor unit (MU) decomposition of EMG signals. MU decomposition is the process of separating the spatiotemporally superimposed action potentials into individual motoneuron firing activities [22, 23, 24]. Different real-time MU decomposition algorithms have shown great promise for neural decoding purposes [25, 26]. Nevertheless, one drawback of this approach is the inefficiency associated with the calculation of MU firing activity. During decomposition, spike trains corresponding to individual MUs are extracted; these are then additively merged to represent the firing behavior of the neural population. Ultimately, knowledge of firing rates of individual MUs, the most time-consuming step of the decomposition algorithm, is not utilized or needed. Instead, a potentially more efficient computational approach could be used to estimate the population neural firing frequency directly from the EMG without performing MU decomposition. With the remarkable advancement in deep learning, deep neural network models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been developed for neural signal processing. For example, gesture recognition can be achieved using multi-stream CNNs [27, 28]. Moreover, EMG signals have been used for inter-person recognition of a set of hand gestures after fine-tuning the model parameters [29]. However, these deep learning-based decoders are still finite-state classifiers, which can only specify a fixed number of movement patterns, rather than providing a continuous prediction of the neural drive signals. Accordingly, the purpose of this study was to develop a robust neural decoding method that can continuously decode the motor command to individual fingers, feasible for real-time control of robotic hands (Figure 1.1). A CNN-based neural network model was implemented to establish continuous mapping from high-density EMG (HD-EMG) signals to population motoneuronal firing frequency for individual finger muscles. The model mapped the HD-EMG amplitude and frequency features directly to populational neuronal firing frequency (i.e., neural drive) without having to perform the intermediate process of MU decomposition. Given that EMG features have a more direct relation with MU firing activity than with fingertip force or joint kinematics, the model was trained on populational firing frequency. This allowed us to establish a generic model that is generalizable to fingertip force or joint kinematic control tasks. The calculation of amplitude and frequency features for the network model input, rather than directly using raw EMG signals, further improved the model performance and learning rate. The network model was first trained using a large dataset of different subjects to learn the general mapping relation, and then the model parameters was calibrated using subject-specific data such that the model can accurately predict the desired finger output (fingertip force and joint angle) of individual subjects. The network-based neural drive decoding performance was evaluated (prediction error and correlation) by comparing with an EMG amplitude approach. The developed neural decoding approach could accurately predict desired fingertip forces and joint angles with minimal predicted output in the undesired fingers. The substantially improved independence of finger control, both in terms of isometric finger force and dynamic movement, provides a foundation for continuous control of assistive devices in a dexterous manner.

CHAPTER 2
METHODS

2.1 Participants

Ten neurologically intact participants (three females and seven males, age: 23-36) were recruited for this study. Prior to the experiment, all participants gave informed consent via protocols approved by the Institutional Review Board of the University of North Carolina at Chapel Hill.

2.2 Experimental Design

The experiment was composed of two types of tasks: isometric finger force tasks and dynamic finger movement tasks. During the experiment, the participants were seated in front of a desk with their right forearm in the neutral position supported by a large foam pad, with a fixed wrist angle. After skin preparation, two 8×16 HD-EMG electrode grids (a 3-mm recording diameter for each electrode and an inter-electrode distance of 10 mm) (OT Bioelettronica) covered the anterior and posterior sides of the forearm to record EMG signals of extrinsic finger flexor and extensor muscles. The position of the electrode grid was based on palpation of the targeted muscles. The biosignal amplifier EMG-USB2+ (OT Bioelettronica) sampled the EMG signals at 2048 Hz with a gain of 1000 and band-pass filtered at 10-900 Hz. In the finger force tasks, the index, middle, and ring fingertip forces were recorded using three miniature load cells (SM-200N, interface) sampled at 1000 Hz. The load cell position was adjusted such that the three fingers spread out comfortably. The participants produced either single- or three-finger isometric flexion for model calibration and validation, by following a target trajectory comprised of pseudorandom force targets (Figure 2.1B), ranging from 0% to 50% maximum voluntary contraction (MVC). The participants completed 22 trials in a random finger order, involving index, middle, ring, and three-finger flexion forces for 4, 4, 4, and 10 target trials, respectively. The participants then produced isometric flexion forces using either a single finger or three fingers for the construction of force regression functions, and they followed a target trajectory composed of a trapezoidal shape ranging from 0% to 50% MVC (Figure 2.1C). Lastly, to test the decoder performance, the participants produced different forces by following a pseudorandom target trajectory (Figure 2.1D), and they repeated the task using either one of the three fingers or three fingers concurrently. The different target trajectories were used to evaluate the generalizability of the decoders in different conditions. In the dynamic finger movement tasks, joint kinematics of the metacarpophalangeal (MCP) joints of the index, middle, and ring fingers were recorded using a custom-made sensor glove, and the MCP angle data were sampled at 100 Hz. The participants produced either single- or three-finger flexion and extension movements for model refinement for 4 trials. They followed a target trajectory comprised of half-sinewaves (Figure 2.1E), ranging from 0% (full extension) to 100% (full flexion) range of motion (ROM) for a total of 10 seconds. The participants then produced either single- or three-finger movement for the construction of angle regression functions, by following a target trajectory from 0% to 100% ROM (Figure 2.1F). Lastly, to test the decoder performance, the participants produced different finger movements from 0% to 100% ROM, and they repeated the task using either one of the three fingers or three fingers concurrently (Figure 2.1G).

2.3 Neural Network Model

A neural network model (parameters detailed in Table B.1 in Appendix B) was implemented with two parallel convolutional neural networks (CNNs) before a fusion network and multi-output layers to predict the population motoneuronal firing frequency associated with the index, middle, and ring fingers. Briefly, The CNN learned the mapping from HD-EMG amplitude and frequency features with embedded spatiotemporal information to populational neuronal firing frequency (i.e., neural drive to individual fingers). The computational latency was 37 ms (16 ms delay from retrieving data packet, up to 3 ms for signal preprocessing, up to 11 ms for neural decoding, and 7 ms for regression and filtering). This delay is well below the acceptable loop delay (100-150 ms) in human-robot interactions [30, 31].

2.3.1 Feature calculation

Although CNN can directly learn features from raw EMG signals, pre-calculated features can reduce the training and calibration time, essential for real-time control. Two types of features were calculated as the input to the neural network model, separately for the flexor and extensor muscles. The 128-channel HD-EMG recordings of the flexor or extensor muscles were segmented into a sequence of 96-sample (46.88 ms) windows with a step size of 64 samples (31.25 ms), and five consecutive windows were used to include temporal information. The first feature map (termed

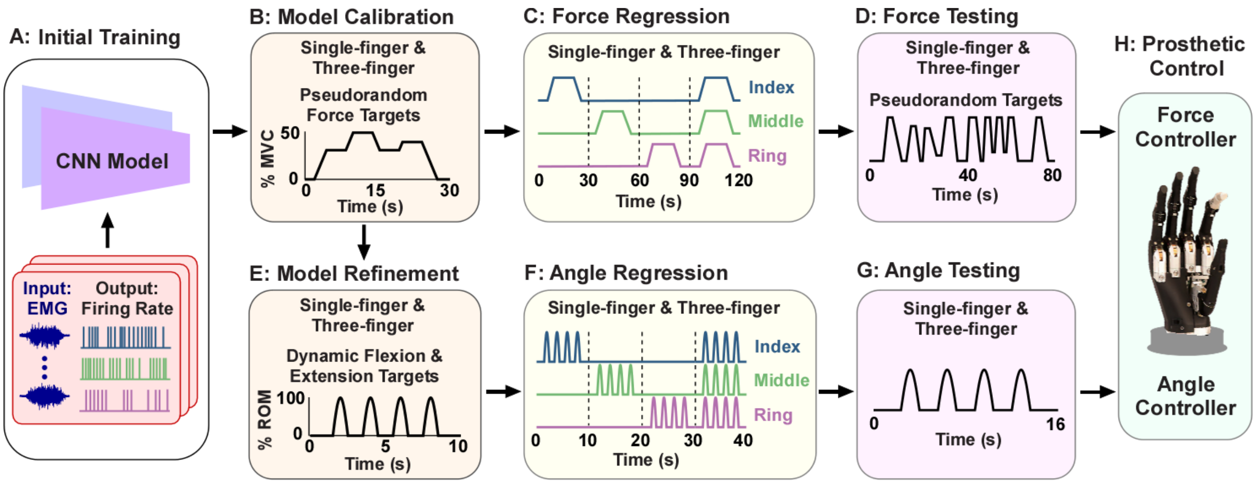

Figure 2.1: Neural network model training, refinement, and testing for real-time control of prosthesis. A: Initial training of the neural network using a large dataset to establish a generic mapping from EMG features to population firing frequency. B: Subject-specific model calibration using data from single- and multi-finger force tasks. The intended fingers produced isometric forces following a pseudorandom target ranging from 0% to 50% MVC. C: Perform a linear regression to convert populational firing frequency to fingertip force. Data from the single- and multi-finger force tasks following a series of trapezoidal targets were used for the regression. D: Testing of the force prediction performance using data from the single- and multi-finger force tasks following a pseudorandom target (0-50% MVC). E: The calibrated model was further refined using subject specific data from dynamic single- and multi-finger movement tasks. The MCP joints started from full extension (0% ROM), and flexed to 100% ROM, and extended back to 0% ROM. F: Perform a quadratic regression to convert populational firing frequency to joint angle. Data from the single- and multi-finger movement tasks were used for the regression. G: Testing of the angle prediction performance using data from the single- and multi-finger movement tasks (0-100% MVC). H: Real-time control of a prosthesis using a force controller or an angle controller.

amplitude map) was calculated based on the root-mean-square (RMS) value of all the samples within a window in a channel-wise manner. The 128 RMS values were then organized into a 16×8 (height×width) amplitude map corresponding to the electrode grid spatial organization in Figure 1.1. Five amplitude maps from five consecutive windows were stacked vertically to form a three-dimensional (3D) array. The 3D array was the amplitude feature $F_{amp} \in \mathbb{R}^{T \times H \times W}$ (Time×Spatial×Spatial), where T=5, H=16, and W=8. The second feature map (termed frequency map) was calculated using the short-time Fourier transform. 128 frequency spectrum vectors corresponding to the 128 channels were calculated within each 96-sample window. Five 1D spectrum vectors from five consecutive windows of the same channel were concatenated into a 2D array. 128 2D arrays from the 128 channels were stacked vertically to form the third dimension, resulting the frequency feature $F_{spec} \in \mathbb{R}^{N \times M \times T}$ (Spatial×Frequency×Time), where N=128, M=49, and T=5.

2.3.2 Neural network structure

Within a convolution operation, a matrix multiplication was performed between the kernel and an area covered by the kernel of either the amplitude feature or the frequency feature, and the results were summed onto the representation map for the next layer. The frequency CNN enabled the exploration of the changes in frequencies at different time scales by sliding the kernel ($\omega \in \mathbb{R}^{N\times p\times q}$) pixel by pixel across the frequency feature maps (2.1), where Y is the output representation map, and i, j are the coordinates indicating the location of the kernels on the input feature. Because action potentials in HD-EMG signals could arrive at different channels at different times, the third dimension of the kernels explored the spatial-frequency relation among channels.

$$Y_{ij} = \sum_{a=0}^{p-1} \sum_{b=0}^{q-1} \omega_{ab} F_{spec_{(i+a)(j+b)}} \tag{2.1}$$

The amplitude CNN was also used to extract the information from the amplitude feature. The kernels ($\nu \in \mathbb{R}^{T\times p\times q}$) were able to detect the shape of the activated area and the change in the EMG amplitude of each channel (2.2), where Z is the output representation map, and i, j are the coordinates indicating the location of the kernels on the input feature.

$$Z_{ij} = \sum_{a=0}^{p-1} \sum_{b=0}^{q-1} \nu_{ab} F_{amp_{(i+a)(j+b)}} \tag{2.2}$$

Since the active area differed when each of the three fingers moved, the retained electrode topographic information in the amplitude feature allowed an intuitive feature extraction for high-level representations, and could provide an accurate finger isolation in terms of the populational firing frequency of each finger. The third dimension of the kernels of the amplitude CNN explored the temporal-spatial relation of the changes in five amplitude maps that varied over time in the same area. By passing through all the convolutional layers, the high-level representation was extracted, and was used to predict the population firing frequency of the three individual fingers. The high-level representations were flattened into feature vectors, and were then passed through fully connected layers, which improved the ability to learn nonlinear combinations from the representational information extracted by the convolutional layers. A fusion network with two fully connected layers mapped the extracted feature vectors to the multi-output branches. Each output branch had one hidden layer with 256 neurons and an output layer with 16 neurons. The probability of each target class (0 to 15), corresponding to the normalized populational firing frequency, was given by the softmax function [32]. Three output branches predicted the firing frequencies for the index, middle, and ring fingers in a concurrent manner. Given the ground truth population neuron firing frequency $\boldsymbol{S} \in \mathbb{R}^{3\times 1}$ and the two input feature maps F_{spec} and F_{amp}, the decoding process can be modeled as finding the prediction that maximizes the following log probability (2.3), where d is the finger number from one to three.

$$log\ p\,(\boldsymbol{S}|F_{spec}, F_{amp}) = \sum_{d=1}^{3} log\ p(\boldsymbol{S}_d|F_{spec}, F_{amp}; \boldsymbol{S}_0,\ \boldsymbol{S}_1,\ \ldots, \boldsymbol{S}_{d-1}) \tag{2.3}$$

In the training process, in order to maximize the log probability of a correct prediction, the training objective was set as (2.4), where K denotes the number of training samples.

$$Objective = \ -\sum_{i=1}^{K} log\ p(\boldsymbol{S}^i|{F_{spec}}^i, {F_{amp}}^i) \tag{2.4}$$

Once the training process is complete, the population firing frequency was estimated by finding the $\hat{\boldsymbol{S}}$ that maximize the probability above: $\hat{\boldsymbol{S}} = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, p(\boldsymbol{S}|F_{spec}, F_{amp})$. The overall loss on cross-entropy is described in (2.5), where $\{\boldsymbol{x}, \boldsymbol{label}\} \in \mathbb{R}^{B\times 3}$, and B is the batch size. $\boldsymbol{x}$ is the network output sent to the softmax function. $\boldsymbol{label}_d \in [0,\ C-1]$ is a scalar, and C=16 is the number of total target classes. $w \in \mathbb{R}^C$ is the weight on each class.

$$loss\,(\boldsymbol{x},\ \boldsymbol{label}) = \sum_{d=1}^{3} -w_{label} \cdot log \frac{e^{\boldsymbol{x}_{d\boldsymbol{label}_d}}}{\sum_{i=0}^{C-1} e^{\boldsymbol{x}_{d_i}}} = -w_{label} \cdot \left[-\boldsymbol{x}_{d\boldsymbol{label}_d} + log \sum_{i=0}^{C-1} e^{\boldsymbol{x}_{d_i}} \right] \tag{2.5}$$

2.3.3 Network output target

To use the population firing frequency as the model output target for training, it was needed to obtain the firing frequency information from the EMG signals. A fast independent component analysis (FastICA) method was used [33, 26, 34] to extract the firing times of individual motor units (MUs). The pseudocode of the algorithm is shown in Algorithm 1 below. Briefly, the raw EMG signals were first extended by an extension factor (f_e=10) and whitened. The separation vectors and associated signal sources representing individual MU information were obtained via a fixed-point iteration algorithm [35, 36]. The binary firing events of individual MU were separated from the background signal sources through a binary classification using the Kmeans++ algorithm [35, 36]. A modified ‘silhouette distance’ (SIL) measurement [37] was used to quantify the separation quality of the source signals, where source signals with low SIL values (<0.6) were excluded from the MU pool.

```
Algorithm 1: MOTOR UNIT DECOMPOSITION ALGORITHM
  Data: Extended and Whitened multi-channel EMG signals
  Result: Firing event trains of individual MUs
  I. Initialize matrix $\underline{\mathbf{B}}$ to an empty matrix
  II. for $j = 1, 2, \ldots, M$ do
      1. Initialize the vector $\mathbf{w}_j(0)$ and $\mathbf{w}_j(-1)$
      2. while $\left|\mathbf{w}_j(n)^T\mathbf{w}_j(n-1) - 1\right| > Tolx$ do
          a. Fixed Point Algorithm:
              $\mathbf{w}_j(n) = E\left\{\underline{\mathbf{z}}g\left[\mathbf{w}_j(n-1)^T\underline{\mathbf{z}}\right]\right\} - A\mathbf{w}_j(n-1)$ with $A = E\left\{g'\left[\mathbf{w}_j(n-1)^T\underline{\mathbf{z}}\right]\right\}$ and
            $g(y) = y^3/3$
          b. Orthogonalization
              $\mathbf{w}_j(n) = \mathbf{w}_j(n) - \underline{\mathbf{B}\mathbf{B}}^T\mathbf{w}_j(n)$
          c. Normalization
              $\mathbf{w}_j(n) = \frac{\mathbf{w}_j(n)}{\|\mathbf{w}_j(n)\|}$
          d. set $n = n + 1$
      end
      3. Estimate the source signal $\hat{s}_j(k) = \mathbf{w}_j(n)^T\underline{\mathbf{z}}(k)$
      4. Estimate the firing event train $\mathbf{T}_j$ using peak detection and then the binary
        classification of the detected peaks using the Kmeans++ algorithm
      5. Calculate SIL according to $\mathbf{T}_j\text{and}\hat{S}_j$
      6. Add $\mathbf{w}_j(n)$ to matrix $\underline{\mathbf{B}}$
  end
```

The retained binary firing activities were further summated to a single composite train, from which the populational firing frequency of the MU pool can be obtained. The population firing frequency was normalized and categorized into 16 target classes ranging from 0 to 15 for the neural network training. The normalization process divided the population firing frequency by the maximum firing frequency, and then scaled up to 15. Inherently, 0 corresponded to the lowest firing frequency and 15 corresponded to the highest firing frequency.

2.4 Neural Network Training and Testing

2.4.1 Initial training

The initial training first established a generic mapping from EMG features to population firing frequency of individual fingers. The neural network training was optimized by the Adam optimizer

[38] with 50% dropout [39]. HD-EMG recordings with decomposed MU firing frequency training targets from a large dataset of isometric finger force tasks [40, 41, 42] were first used to train the neural network (Figure 2.1A). To optimize the neural network with multi-finger targets, the firing frequency of each finger was calculated via the summation of the firing events of individual MUs grouped by fingers, and was then normalized and scaled to the target classes ranging from 0 to 15. All the training data in the dataset, except a trial retained for validation, were used to train the neural network for 1500 epochs. The validation was performed after each epoch by evaluating the performance of the neural network on the validation trial. A validation score was calculated as the arithmetic mean of the three correlation coefficients between the predicted firing frequencies and the actual measured forces of the index, middle, and ring fingers of the validation trial. After the training session has reached the maximum number of epochs, the weight coefficients of the neural network right after the epoch with the highest validation score were adopted for the initially trained model.

2.4.2 Model calibration

To adapt to the EMG characteristics of individual subjects, the CNNs were fine-tuned using calibration data of each subject (Figure 2.1B). The calibration procedure was guided by a validation trial, including one trial in each of the four force tasks (index, middle, ring, and three-finger). Based on the initially trained weight coefficients, the neural network was fine-tuned for 80 epochs, and the validation was performed at the end of each epoch. In each epoch, all the training data from the calibration trials were used once. The same validation score as in the initial training was used here. The weight coefficients of the neural network right after the epoch with the highest validation score were adopted.

2.4.3 Model refinement for joint angle control

Given the neural network model was never trained or refined on data involving dynamic movement, a second model refinement was performed using dynamic task based on the calibrated model using data from finger force tasks (Figure 2.1E). The joint angle decoding performance without this second refinement is shown in Figure A.4 in Appendix A. The second refinement was guided by a validation trial comprised of 4 dynamic movement segments, each including one index, middle, ring, and three-finger tasks. Similarly, the neural network was fine-tuned for 80 epochs with validation performed at the end of each epoch. The weights of the neural network after the epoch with the highest validation score were adopted.

2.4.4 Fingertip force and joint angle regression

To derive the motor output (fingertip force or joint angle) for continuous control of the prosthesis, regression analyses were performed. The predicted firing frequency was first smoothed by a moving average filter (500-ms window with a moving step of 31.25 ms). The smoothed firing frequency was further smoothed by a Kalman filter [43]. As a result, the smoothed firing frequencies were updated every 31.25 ms (32 Hz) for the three individual fingers concurrently. The 32 Hz was chosen because the data packet can be obtained from the acquisition system at 32 Hz. Three bivariate linear regression models were constructed using data from the force regression trials (Figure 2.1C) to estimate three finger forces based on the smoothed firing frequency. The data of the intended finger were used for each regression. In addition, three bivariate quadratic regression models were constructed using data from the dynamic angle regression trials (Figure 2.1F), in order to predict MCP angles of the three fingers based on the smoothed firing frequency. An earlier study has shown that the quadratic regression function outperformed the linear function for dynamic kinematic estimation [43].

2.4.5 Evaluation of decoding performance

To evaluate the decoder performance, real-time decoding of intended fingertip forces and joint angles were performed on HD-EMG signals when subjects performed fingertip force tasks (Figure 2.1D) or dynamic MCP flexion and extension tasks (Figure 2.1G). As a comparison with the neural network decoder, a EMG amplitude method [40] (termed EMG method) was implemented to estimated force and angle prediction. First, the top 60 channels with the highest RMS value out of the 128 channels were selected separately for the flexor and extensor muscles. The EMG data during the activation of individual intended fingers were used for the channel selection. Second, because of EMG activities associated with inevitable activation of unintended fingers, some of the selected top 60 channels could represent motor output of unintended fingers. This could lead to false positive errors in motor output predictions of intended fingers. To address this issue, a channel refinement procedure was performed to further remove channels with potential EMG recordings (i.e., cross-talk) of unintended fingers. Specifically, the EMG amplitude (RMS) was calculated using the moving average filter (500-ms window with a moving step of 31.25 ms) for the individual 60 channels. A regression analysis was then performed between the EMG amplitude and different finger forces during individual finger force tasks. If the coefficient of determination (R^2) value with the intended fingers were higher than with the other two fingers, the given EMG channel was retained. Otherwise, the channel was removed. The assumption was that the EMG channel with muscle activities associated with a given finger should have a high correlation with the motor output of that finger. The same refinement procedure was also performed among the 60 EMG channels for the dynamic movement tasks, except that the regression analysis was performed between the EMG amplitude and the measured joint angles of the three fingers. Finally, the RMS values of the retained EMG channels were calculated using the same moving average filter method and averaged across channels for each finger. The same Kalman filter was applied to the average RMS values. Three bivariate linear regression functions for the EMG method were also constructed using data from the force regression trials, and three bivariate quadratic functions were constructed using data from the angle regression trials.

2.4.6 Prosthesis control

The predicted force or angle data of the CNN method were also used to control the MCP joint of the index, middle, and ring fingers of the prosthetic hand (i-Limb, Ossur) using a custom MATLAB (MathWorks Inc) interface. Three force-sensitive resistors (FSR) and three angle sensors were fixed on the prosthetic fingers to record the fingertip forces and MCP joint angles, respectively. The force and angle information was sent to two custom-made proportional derivative (PD) controllers as feedback signals to control the fingertip force or the MCP angle of each of the three fingers. For the isometric force task, a force controller was employed, while an angular position controller was utilized in the dynamic task. With the force controller, the amount of force generated by the prosthetic hand was modulated by the predicted forces from the CNN method. With the angular position controller, motor commands altered each finger's angular position in order to match the predicted joint angle of the CNN method. Both controllers updated the reference force or angle at a rate of 8 Hz, while motor commands were updated at a rate of 32 Hz. The force recordings of the FSR were monitored, if the force was > 1 N, the force regression functions and the force controller were activated, otherwise, the angle regression function and angle controller were used.

2.5 Statistical Analysis

To quantify the performance of force and joint angle predictions of the CNN and EMG methods, the root mean squared error (RMSE) and coefficient of determination (R^2) values were calculated between the actual measured forces (or angles) and the predicted forces (or angles). To quantify finger isolation of the decoding, both intended and unintended fingers were evaluated in the single-finger tasks. Repeated measures analysis of variance (ANOVA) was performed on the dependent variables. The significance level α was set as 0.05. A pairwise comparison was conducted using the Bonferroni method when necessary. To further quantify finger isolation, we also categorized the fingers into active or rest states based on the predicted finger output (force or angle) in the single-finger tasks. Predicted finger output time series were categorized into active or rest states based on an output threshold (10% MVC for the force tasks and 10% ROM for the dynamic movement tasks). At a given time, if the output of a finger was above the threshold, this finger was categorized as active at that time. The percentages of output data samples in different finger combination categories were calculated. A high percentage only in the single intended finger category was considered a better finger isolation than a high percentage in the multi-finger categories

CHAPTER 3
RESULTS

3.1 Fingertip force decoding

To quantify the force decoding performance, the RMSE was calculated between the actual measured fingertip force via load cells and the predicted force (either from CNN method or EMG method). The decoding performance in the single-finger force task (one finger produced flexion forces following a pseudorandom target while other fingers remained relaxed) is shown in Figure 3.1. A representative trial of the index finger flexion task is shown in Figure 3.1A. The force prediction of the CNN method accurately followed the actual measured force in the intended index finger throughout the trial. In contrast, the force prediction of the EMG method showed large deviation from the actual force, especially in the trial period > 40 s. In addition, the EMG method also demonstrated large false positive forces in the unintended fingers (middle and ring). Figure 3.1B summarizes the overall average force prediction errors of the intended fingers. Filled circles represented individual subjects. The two-way (method (CNN vs. EMG) × finger (index vs. middle vs. ring)) ANOVA showed that the RMSE of the CNN method was significantly smaller than that of the EMG method across fingers (F(1,9)=10.396, p=0.010) with no interaction effect. Figure 3.1C shows the force error of the unintended fingers. Since the unintended fingers were supposed to produce no force output, a zero-newton force was considered the ground-truth of the unintended fingers. Accordingly, the RMSE of the unintended fingers was calculated between zero-newton force and the actual or predicted forces. Essentially, a larger RMSE indicated a higher force in the unintended fingers. The two-way (method (CNN vs. EMG vs. actual force) × finger (index vs. middle vs. ring)) repeated measures ANOVA revealed a significant effect of the method (F(2,18)=49.934, $p<0.001$) with no interaction effect.

Further pairwise comparison showed that the RMSE of the EMG method was significantly higher than that of the CNN method ($p<0.001$), and that of the actual force ($p<0.001$). The RMSE of the CNN method was also higher than that of the actual measured force ($p<0.01$).

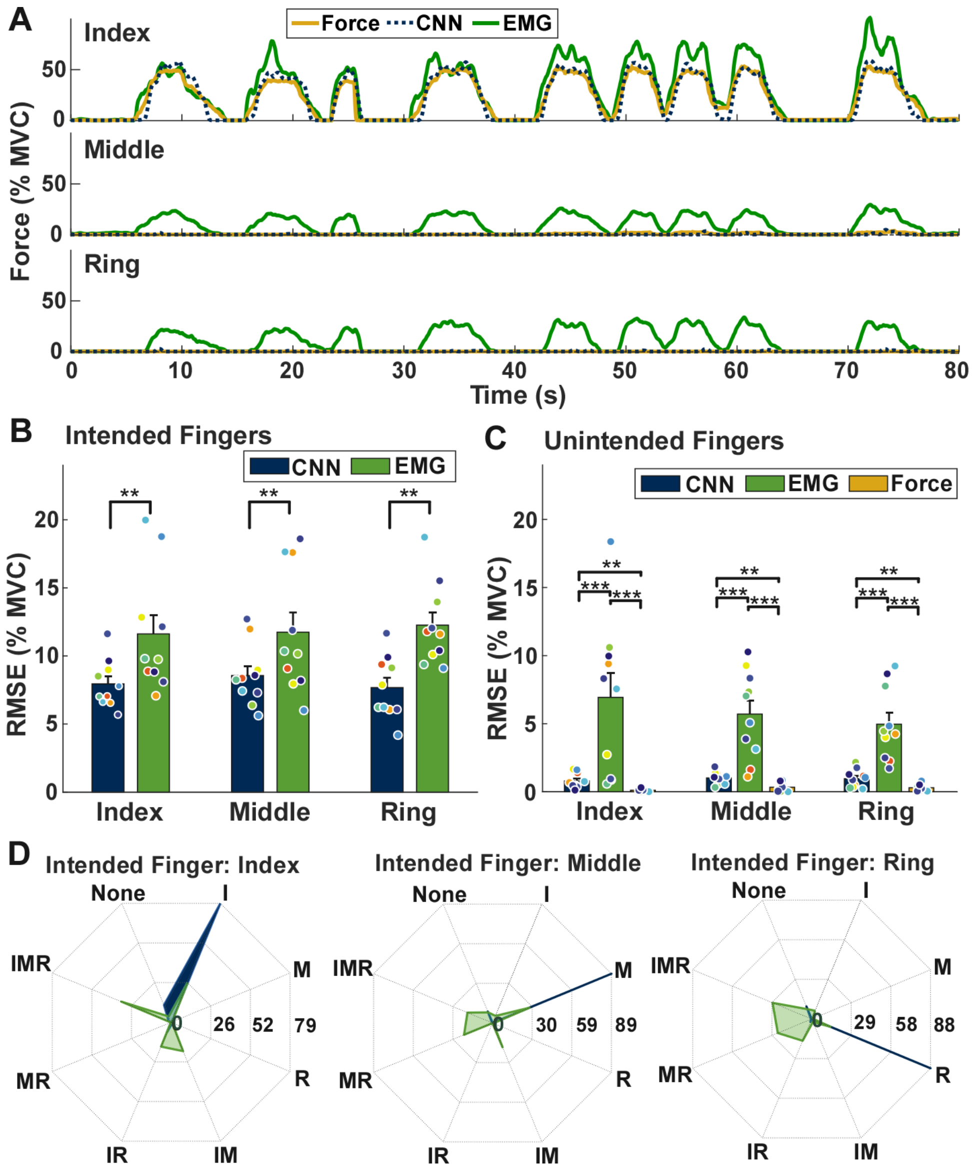

Figure 3.1: Prediction of individual fingertip forces using the CNN and EMG methods. A: An example trial of individual fingertip force prediction of the CNN and EMG methods, when the index finger was instructed to produce the required forces while other fingers relaxed. The summary of force prediction performance calculated as the root-mean-square error (RMSE) is shown separately for the intended (B) and unintended (C) fingers. Filled circles represent individual subjects with the same color denoting the same subject. Error bars represent standard error. D: Active finger categorization of the CNN method (blue) and the EMG method (green). I, M, and R represent index, middle, and ring fingers, respectively. Predicted force time series of the three fingers were categorized into active or rest states based on a 10% maximum voluntary contraction (MVC) threshold, with > 10% MVC categorized as active. If the predicted forces of all the three fingers were smaller than 10% MVC at a given time, it was counted as the None category. The radius of the plot represented the percentage of force data samples in each category, with different contour lines represent different percentage values. *, p<0.05. **, p<0.01. ***, p<0.001.

To further quantify finger isolation of the decoders, we also categorized the fingers into active or rest states based on the predicted finger forces (Figure 3.1). Predicted force time series of the three fingers were categorized into active or rest states based on a 10% maximum voluntary contraction (MVC) threshold. Similar categorization results based on the 5% and 15% MVC thresholds are shown in Figure A.1 in Appendix A. At a given time, if the force of a finger was above the 10% MVC, this finger was categorized as active. Most of the predicted force samples of the CNN method corresponded to the active intended fingers, and most of the forces of the unintended fingers were categorized into a rest state. In comparison, the force of the EMG method predicted co-activation of the intended finger coupled with one or multiple unintended fingers. For example, in the index finger trial the EMG method predicted I, IM, IR, and IMR active fingers, in which the IM, IR, and IMR were false positive states.

The decoding performance in the multi-finger force task (index, middle, and ring fingers produced flexion forces concurrently) is shown in Figure 3.2. A representative trial is shown in Figure 3.2A. The force predictions of the CNN method accurately followed the actual forces of the three fingers. In contrast, the force prediction of the EMG method demonstrated large deviation from the actual forces throughout the trial. Figure 3.2B illustrates the average force prediction errors of the intended fingers. The one-way repeated measures ANOVA showed that the RMSE of the CNN method (8.8% ± 0.7%) was significantly smaller than that of the EMG method (17.8% ± 1.9%) (F(1,9)=59.349, $p<0.001$). Figure 3.2C illustrates the average coefficient of determination (R2) between the actual and predicted forces of the two methods. The ANOVA results showed that the R2 value of the CNN method (0.836 ± 0.014) was significantly larger than that of the EMG method (0.637 ± 0.055) (F(1,9)=11.948, $p=0.007$).

In addition, the CNN method also showed more stable performance across tasks and over time. Specifically, the force prediction error of the CNN method was comparable between the single-finger and multi-finger tasks, while the force prediction error of the EMG method increased substantially from the single-finger task to the multi-finger task (Figure 3.2D). As shown in Figure A.2 in Appendix A, the force prediction error during initial force regression was similar between the CNN and EMG methods. However, a substantially larger prediction error was observed during subsequent testing in the EMG method, compared with the CNN method, indicating a degrading performance over time of the EMG method.

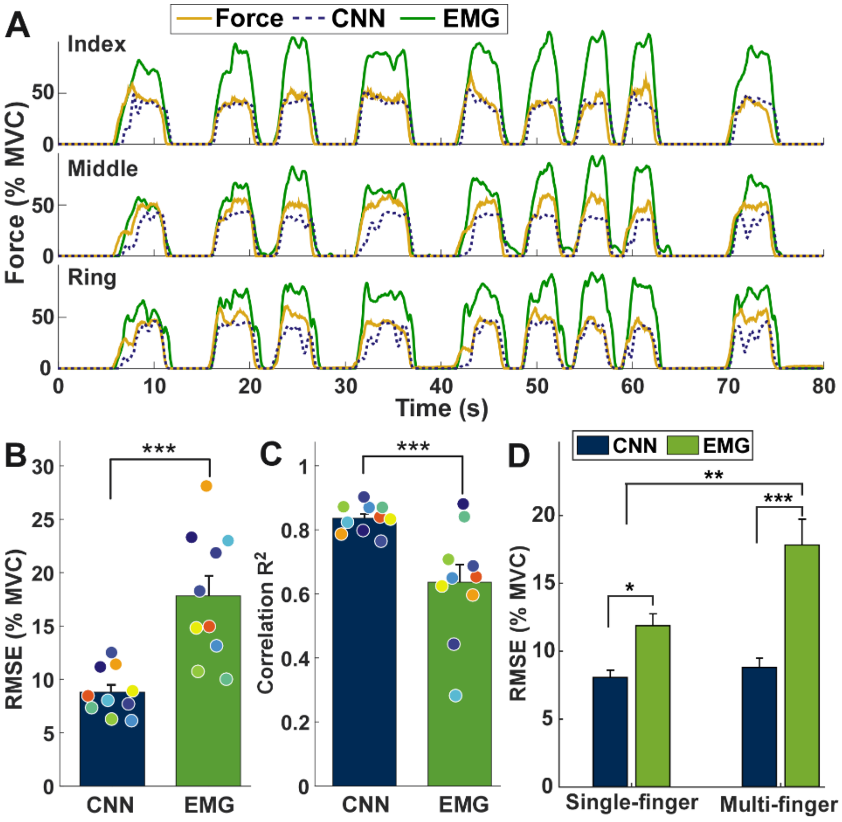

Figure 3.2: Prediction of multi-finger forces concurrently using the CNN and EMG methods. A: An example trial of multi-finger force prediction of the CNN and EMG methods. The averaged RMSE (B) and averaged coefficient of determination (C) of the two methods. D: A comparison of the RMSE of the two methods across the finger- and multi-finger tasks. Filled circles of the same color represent the same subjects. Error bars represent standard error. *, $p<0.05$. **, $p<0.01$. ***, $p<0.001$.

3.2 Finger joint angle decoding

We then evaluated the decoding performance of joint kinematics of individual fingers when subjects produced either single-finger or multi-finger flexion and extension movements around the metacarpophalangeal (MCP) joints. The decoding performance in the single-finger joint movement task is shown in Figure 3.3. An exemplar trial of the ring finger flexion-extension task is illustrated in Figure 3.3A. The CNN method showed a better angle prediction performance in comparison with the EMG method. The CNN method also revealed an underestimation of the joint angle in the unintended fingers, whereas the EMG method demonstrated an overestimation of the joint angle. Figure 3.3B illustrates the average prediction errors of the intended fingers across subjects. The two-way (method (CNN vs. EMG) × finger (index vs. middle vs. ring)) ANOVA revealed that the RMSE of the CNN method was significantly smaller than that of the EMG method across fingers (F(1,9)=5.703, $p=0.041$) with no interaction effect, and no significant finger effect.

We also quantified movement errors of the unintended fingers. Since the unintended fingers were supposed to produce no joint movement, a zero-degree angle movement was considered the ground-truth of the unintended fingers. Accordingly, the RMSE of the unintended fingers was calculated between zero-degree angle and the actual or predicted angles. The RMSE between zero-degree angle and the actual angle (or the predicted angles) were calculated (Figure 3.3C). The ANOVA results indicated that the method (F(2,18)=136.951, $p<0.001$) had a significant effect on the RMSE with no interaction effect. The pairwise comparison showed that the RMSE of the EMG method was significantly higher than that of the CNN method ($p<0.001$) and the actual joint angle ($p<0.01$). The RMSE of CNN was significantly smaller than that of the actual joint angle ($p<0.001$), indicating that the CNN method can better predict intended movement of the subjects than the actual movement. We also quantified the active vs. rest states of the fingers based on the predicted joint angles (Figure 3.3D). At a given time, if the angle of a finger was above the 10% range of motion (ROM) threshold, the finger was considered active. Most of the predicted joint angles of the CNN method were categorized into the active intended fingers, with most of the angle of the unintended fingers categorized into a rest state. In contrast, angles of the EMG method predicted coactivation of the intended finger and one or multiple unintended fingers, with the highest percentage observed in the three-finger IMR active state regardless of the task; this indicates poor finger isolation during decoding. Similar categorization results based on the 5% and 15% ROM thresholds are shown in

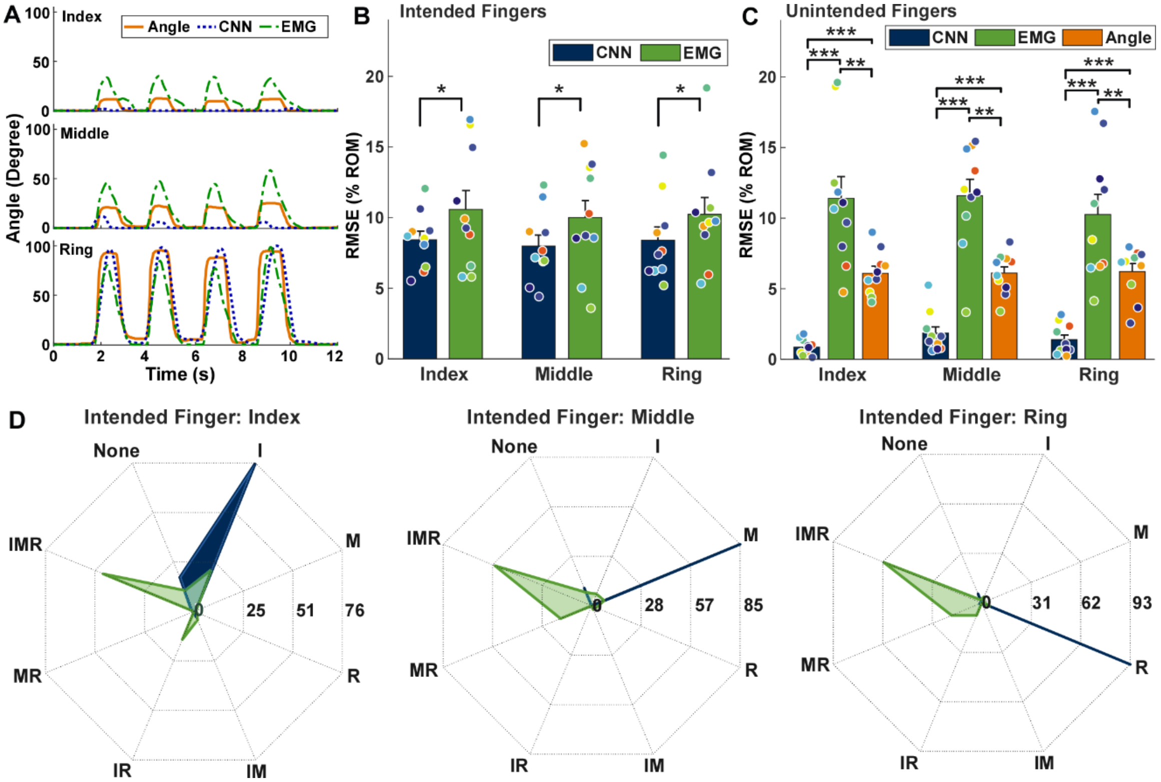

Figure 3.3: Prediction of individual finger joint angles using the CNN and EMG methods. A: An example trial of the joint angle prediction, when the ring finger was instructed to flex and extend while other fingers relaxed. Zero-degree means full extension. The RMSE of individual fingers was shown for the intended (B) and unintended (C) fingers. The range of motion (ROM) was normalized by the maximum flexion angle of individual fingers. Filled circles of the same color represent the same subject. Error bars represent standard error. D: Active finger categorization of the CNN method (Blue) and the EMG method (Green). I, M, and R represent index, middle, and ring fingers, respectively. Predicted angle time series of the three fingers were categorized into active or rest states based on a 10% ROM threshold. The radius of the plot represented the percentage of force data samples in each category, with different contour lines represent different percentage values. *, $p<0.05$. **, $p<0.01$. ***, $p<0.001$.

Figure A.3 in Appendix A.

The decoding performance in the multi-finger movement task is shown in Figure 3.4. An exemplar trial comparing the actual and predicted joint angles is shown in Figure 3.4A. The predicted angle of the CNN method followed the actual measured joint angle accurately. In contrast, a relatively larger prediction error of the EMG method was observed across the three fingers, especially at the peak flexion angles. The one-way repeated measures ANOVA showed that the RMSE (Figure 3.4B) of the CNN method was significantly smaller than that of the EMG method (F(1,9)=13.071, p=0.006). Similarly, the R2 (Figure 3.4C) of the CNN method was significantly higher than that of the EMG method (F(1,9)=6.892, p=0.028).

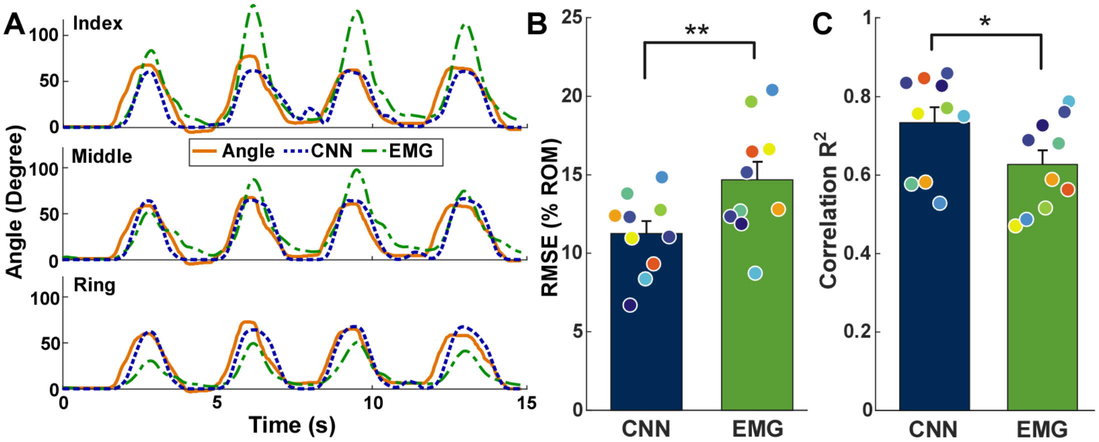

Figure 3.4: Prediction of multi-finger joint angles concurrently using the CNN and EMG methods. A: An example trial of multi-finger joint angle prediction of the CNN and EMG methods. The averaged RMSE (B) and coefficient of determination (C) of the two methods. Filled circles represent individual subjects. Error bars represent standard error. *, $p<0.05$. **, $p<0.01$

CHAPTER 4
DISCUSSION

This study developed a robust neural decoding method that can predict individual finger output (fingertip force and joint angle) for real-time control of assistive devices. The CNN model mapped the HD-EMG features to population neuronal firing frequency (i.e., neural drive) specific to individual fingers, which allows accurate and robust prediction of single- and multi-finger tasks. The pre-calculation of HD-EMG features and the two-step model training (inter-subject initial training and subject-specific refinement) enable an efficient learning of EMG feature mapping to neural drive signals. The decoding of neural drive based on population neuronal firing activity demonstrated robust decoding performance, with consistent decoding accuracy across fingers, across tasks, and over time. In contrast, the performance of the EMG decoder varied across tasks with higher error in the multi-finger task than the single-finger task. The EMG decoder also degraded over time, with a low decoding error during the initial regression, but the error increased substantially during subsequent testing. The outcomes suggest that population neuronal decoding offers a robust yet efficient method for continuous motor intent detection. The continuous decoding of motor output at the individual finger level provides a basis for dexterous control of assistive robotic devices.

The CNN decoder also demonstrated great finger isolation with minimal force or angle prediction in unintended fingers. Motor intent detection of isolated fingers has been a long-standing issue, because of our limited ability to activate individual finger muscle compartments [44, 45] and inevitable cross-talk of surface EMG recordings arising from different muscle compartments in close proximity [18, 46]. Indeed, our EMG decoding method revealed substantial false positive error in the unintended fingers, despite a channel selection procedure based on motor output of the desired fingers. Although channel selection can reduce the impact of cross-talk, by selecting the best channels that correspond to the desired finger output [46], the EMG signals can still contain substantial activity from neighboring compartments. In contrast, the decoding of neural drive signals using the CNN model can accurately detect intended and unintended finger output in both single-finger and multi-finger tasks. In fact, the predicted motor output, especially the predicted joint angle, was smaller than the actual measured joint angle in the unintended fingers. This is beneficial from a motor intent detection perspective, in that the decoded motor outputs reflect the desired motor tasks rather than the actually produced motor output. The underestimation of joint angle in the unintended fingers can arise from several factors. First, the surface HD-EMG grid on the forearm only captured activities of the extrinsic muscles near the skin surface. Activities of the intrinsic finger muscles in the hand and the flexor digitorum profundus were not recorded, which could contribute to the actual movement of the MCP joint in the undesired fingers. Second, it is well known that there is coordinated movement among fingers that limit finger independent. Mechanical coupling from the tendinous structure and skin connections across fingers [44, 47] is one factor that contributes to coordinated finger movements. However, this mechanical coupling effect was not considered in the CNN model.

The current study focused on the decoding of individual finger motor output, and the decoder performance was demonstrated on the real-time control of individual prosthetic fingers. Our decoding algorithm can also be used for the control of exoskeleton hands to assist individuals with hand impairment. In future work, we will perform the decoding of all five digits, especially the thumb movement, which will allow us to evaluate the decoder performance in functional tasks involving object manipulations. In addition, the current study only evaluated intact individuals, future work will investigate the decoder performance of individuals with an arm amputation. There will likely be a limit on the length of residual arm needed for the HD-EMG array placement. Although a subject-specific model refinement was performed in the current study, further refinement may be needed to cope with the different muscle activation patterns of different amputated individuals.

CHAPTER 5
CONCLUSION

In this study, a robust neural decoding method was developed to predict the populational firing frequency specific to individual fingers. The predicted population firing frequency was then used to control multiple digits of a robotic hand in a real-time manner. The method was evaluated in the isometric condition and the dynamic condition, where single-finger and multi-finger tasks were evaluated in both conditions. Our results showed that the extracted MU firing frequency information with regression models can obtain a better force estimation performance and a better joint angle estimation performance, compared with the conventional EMG amplitude-based method. Further development of this method can potentially provide a more reliable prediction of all five digits to achieve intuitive control of robotic hands with higher dexterity.

APPENDIX A

SUPPLEMENTAL FIGURES

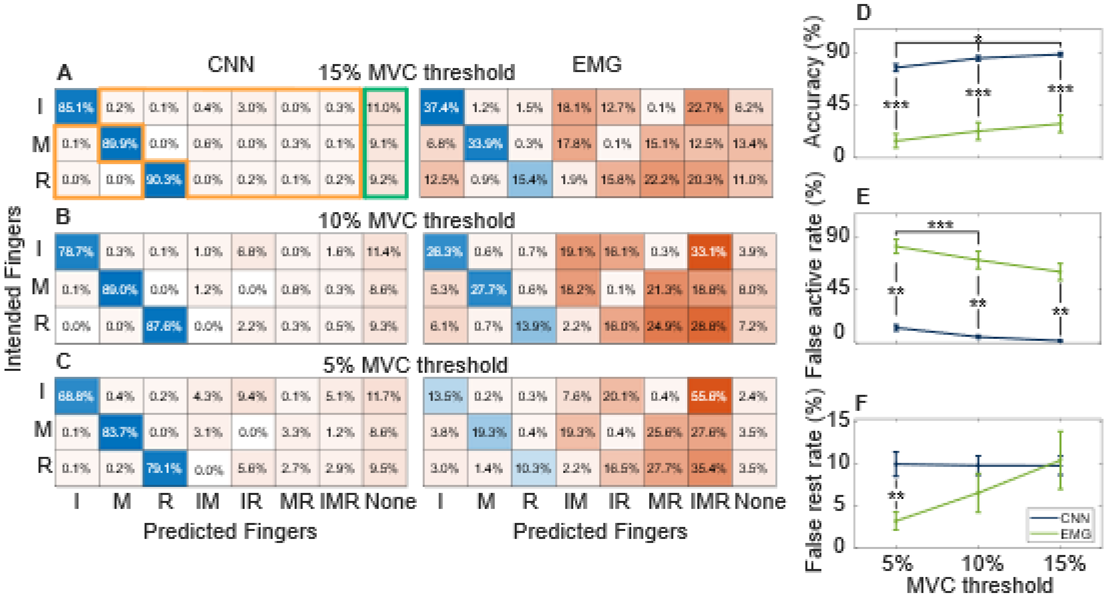

Figure A.1: Categorization of the intended fingers based on the predicted forces of the CNN and EMG methods in the single-finger tasks. The predicted forces of the three fingers were first normalized by the MVC of each finger, and were then categorized into active or rest states based on different force thresholds (5%, 10%, and 15% MVC). Specifically at a given time, if the predicted force of a finger was above the threshold, this finger was categorized as active at that time. The percentages of force samples in different finger combination categories were calculated and averaged across all subjects. The confusion matrics using the three thresholds are constructed in A, B, and C. Each row represents the prediction probability distribution in a single-finger force task when the intended finger was the index (I), middle (M), and ring (R) fingers, respectively, using the CNN method (left) or the EMG method (right). As an example in the confusion matrix of CNN prediction in A, the orange contour enframes the false active rate (probability of an unintended finger being identified as active), the green contour encircles the false rest rate (probability of an intended finger being identified in a resting state), and the blue cells are the true positive rate (probability of accurate categorization). Substantial false active errors were observed across the three thresholds in the EMG method, indicating poor finger isolation during force prediction. D: The average accuracy across all subjects across the three thresholds. E: The average false active rate across all subjects. F: The average false rest rate across all subjects. Error bars represent standard error. *, p<0.05; **, p<0.01; ***, p<0.001

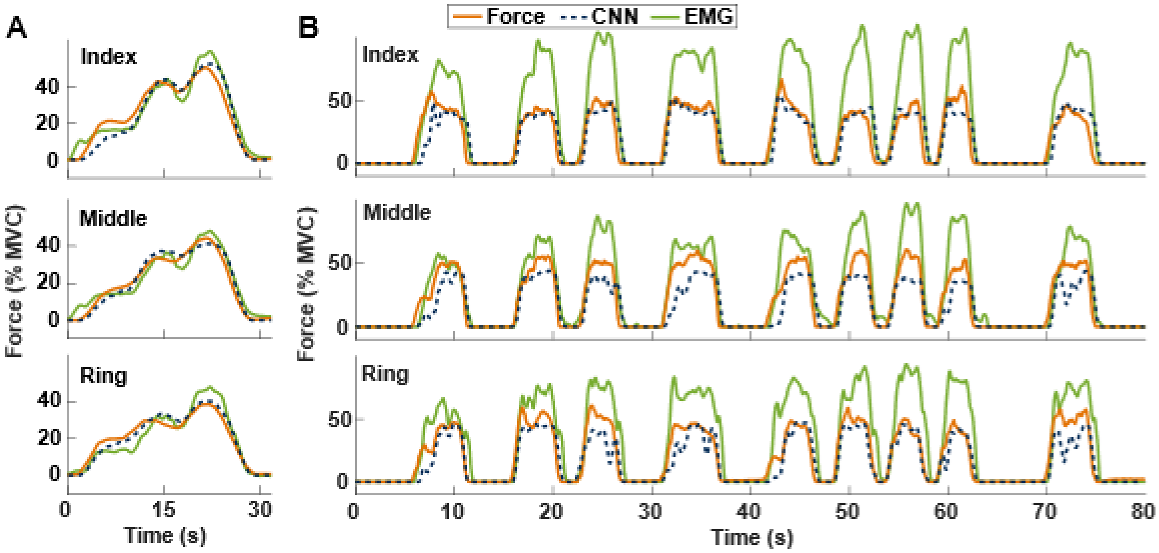

Figure A.2: Force prediction during regression and testing. A: An examplar trial during the initial force regression for the CNN and the EMG methods. B: An examplar trial during the subsequent testing of force prediction using the regression function for the CNN and the EMG methods. A substantially larger force prediction error was observed during testing in the EMG method, compared with the CNN method.

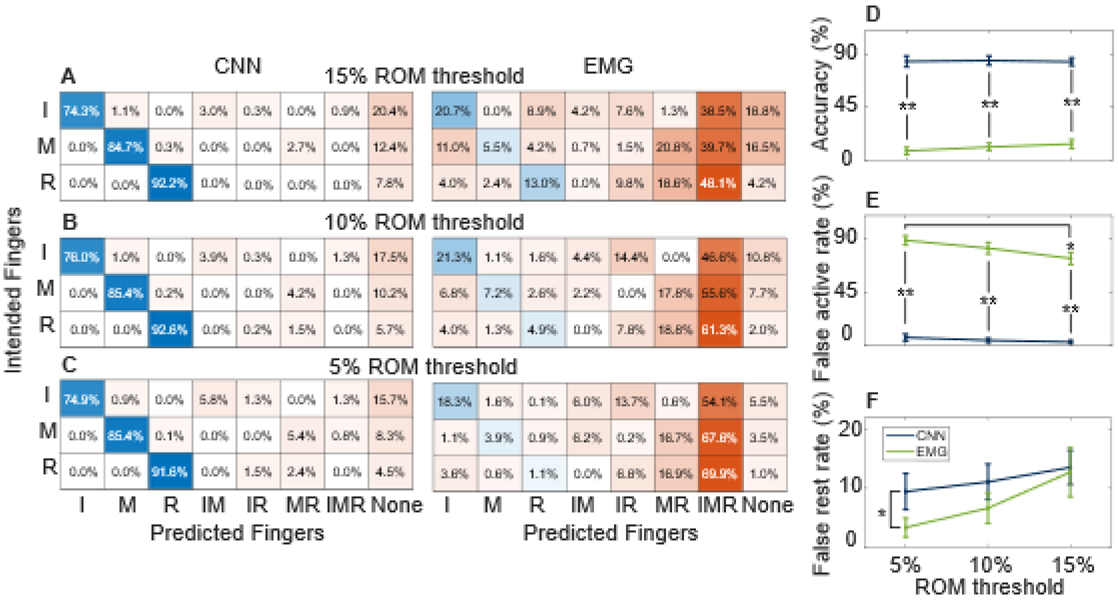

Figure A.3: Categorization of the intended fingers based on the predicted joint angles of the CNN and EMG methods in the single-finger tasks. The predicted joint angles of the three fingers were first normalized by the maximum ROM of MCP joint of each finger, and were then categorized into active or rest states based on different angle thresholds (5%, 10%, and 15% ROM). Specifically at a given time, if the predicted angle of a finger was above the threshold, this finger was categorized as active at that time. The percentages of angle samples in different finger combination categories were calculated and averaged across all subjects. The confusion matrics using the three thresholds are constructed in A, B, and C. Each row represents the prediction probability distribution when the intended finger was the index (I), middle (M), and ring (R) fingers, respectively. As an example in the confusion matrix of CNN prediction in A, the orange contour enframes the false active rate, the green contour encircles the false rest rate, and the blue cells are the true positive rate (accuracy). Substantial false active errors were observed across the three thresholds in the EMG method, indicating poor finger isolation during joint angle prediction. D: The average accuracy across all subjects across the three thresholds. E: The average false active rate across all subjects. F: The average false rest rate across all subjects. Error bars represent standard error. *, p<0.05; **, p<0.01.

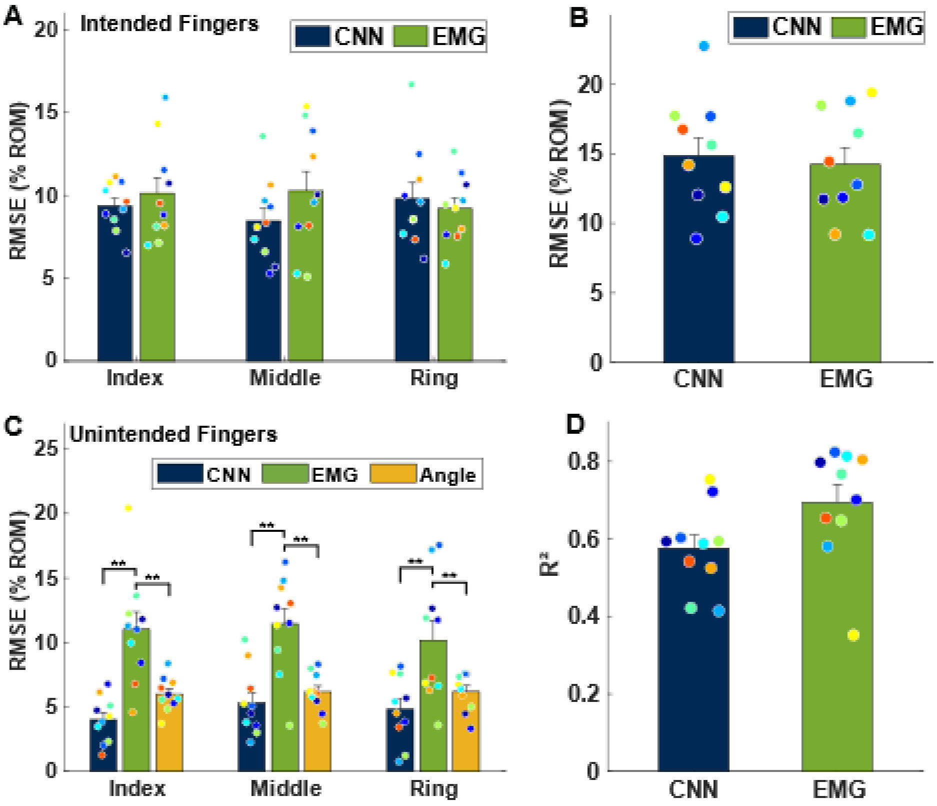

Figure A.4: Joint angle estimation results using the CNN method without refinement and the EMG method. In the single-finger joint movement task, the RMSE of individual fingers for the intended (A) and unintended fingers (C). In the multi-finger joint movement task, the average RMSE (B) and coefficient of determination (D) of the two methods. Filled circles represent individual subjects with the same color denoting the same subject. Without refinement using dynamic movement data, the CNN method showed similar performance as the EMG method in the intended fingers. The angle prediction errors of the CNN method were still smaller than that of the EMG method in the unintended fingers. Error bars represent standard error. **, $p<0.01$.

APPENDIX B

TABLE OF CONVOLUTIONAL NEURAL NETWORK STRUCTURE

Table B.1: Convolutional Neural Network Structure.

<table>
<tr><td>Input feature</td><td colspan="2">F_{freq}</td><td colspan="2">F_{amp}</td></tr>
<tr><td>CNN module</td><td colspan="2">Frequency CNN</td><td colspan="2">Amplitude CNN</td></tr>
<tr><td rowspan="14">Parallel CNN structure</td><td colspan="2">[3×3, 128]</td><td colspan="2">[3×3, 64]</td></tr>
<tr><td colspan="2">[6×2, 128]</td><td colspan="2">[6×2, 64]</td></tr>
<tr><td colspan="2">maxpool-3×1</td><td colspan="2">maxpool-1×3</td></tr>
<tr><td colspan="2">[3×3, 128]</td><td colspan="2">[3×3, 64]</td></tr>
<tr><td colspan="2">[6×2, 256]</td><td colspan="2">[2×4, 64]</td></tr>
<tr><td colspan="2">maxpool-3×1</td><td colspan="2">maxpool-1×2</td></tr>
<tr><td colspan="2">[3×3, 256]</td><td colspan="2">[3×3, 64]</td></tr>
<tr><td colspan="2">[3×3, 512]</td><td colspan="2">[2×2, 64]</td></tr>
<tr><td colspan="2">maxpool-2×2</td><td colspan="2">maxpool-2×2</td></tr>
<tr><td colspan="2">[3×3, 512]</td><td colspan="2">[3×3, 64]</td></tr>
<tr><td colspan="2">[2×2, 1024]</td><td colspan="2">[2×2, 128]</td></tr>
<tr><td colspan="2">maxpool-2×2</td><td colspan="2">maxpool-2×2</td></tr>
<tr><td colspan="2">FC-1024</td><td colspan="2">FC-256</td></tr>
<tr><td colspan="2">FC-512</td><td colspan="2">-</td></tr>
<tr><td rowspan="2">Fusion network</td><td colspan="4">FC-512</td></tr>
<tr><td colspan="4">FC-512</td></tr>
<tr><td rowspan="3">Multi-output branches</td><td>FC-256</td><td colspan="2">FC-256</td><td>FC-256</td></tr>
<tr><td>FC-16</td><td colspan="2">FC-16</td><td>FC-16</td></tr>
<tr><td>softmax</td><td colspan="2">softmax</td><td>softmax</td></tr>
</table>

REFERENCES

[1] A. M. Association, *Guides to the Evaluation of Permanent Impairment.* American Medical Association, 1990.

[2] K. D. Anderson, "Targeting recovery: priorities of the spinal cord-injured population," *Journal of neurotrauma*, vol. 21, no. 10, pp. 1371–1383, 2004.

[3] C. Castellini, "Upper limb active prosthetic systems-overview," in *Wearable Robotics*, pp. 365–376, Elsevier, 2020.

[4] M. S. Johannes, E. L. Faulring, K. D. Katyal, M. P. Para, J. B. Helder, A. Makhlin, T. Moyer, D. Wahl, J. Solberg, S. Clark, *et al.*, "The modular prosthetic limb," in *Wearable Robotics*, pp. 393–444, Elsevier, 2020.

[5] T. Worsnopp, M. Peshkin, J. Colgate, and D. Kamper, "An actuated finger exoskeleton for hand rehabilitation following stroke," in *2007 IEEE 10th international conference on rehabilitation robotics*, pp. 896–901, IEEE, 2007.

[6] K. Tong, S. Ho, P. Pang, X. Hu, W. Tam, K. Fung, X. Wei, P. Chen, and M. Chen, "An intention driven hand functions task training robotic system," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 3406–3409, IEEE, 2010.

[7] M. Alam, W. Rodrigues, B. N. Pham, and N. V. Thakor, "Brain-machine interface facilitated neurorehabilitation via spinal stimulation after spinal cord injury: Recent progress and future perspectives," *Brain Research*, vol. 1646, pp. 25–33, 2016.

[8] R. A. Andersen, T. Aflalo, and S. Kellis, "From thought to action: The brain–machine interface in posterior parietal cortex," *Proceedings of the National Academy of Sciences*, vol. 116, no. 52, pp. 26274–26279, 2019.

[9] C. Castellini, P. Artemiadis, M. Wininger, A. Ajoudani, M. Alimusaj, A. Bicchi, B. Caputo, W. Craelius, S. Dosen, K. Englehart, *et al.*, "Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography," *Frontiers in neurorobotics*, vol. 8, p. 22, 2014.

[10] J. Stein, K. Narendran, J. McBean, K. Krebs, and R. Hughes, "Electromyography-controlled exoskeletal upper-limb–powered orthosis for exercise training after stroke," *American journal of physical medicine & rehabilitation*, vol. 86, no. 4, pp. 255–261, 2007.

[11] G. Li, A. E. Schultz, and T. A. Kuiken, "Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 185–192, 2010.

[12] D. Yatsenko, D. McDonnall, and K. S. Guillory, "Simultaneous, proportional, multi-axis prosthesis control using multichannel surface emg," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6133–6136, IEEE, 2007.

[13] S. Lee and G. Saridis, "The control of a prosthetic arm by emg pattern recognition," *IEEE Transactions on automatic control*, vol. 29, no. 4, pp. 290–302, 1984.

[14] P. Zhou, M. M. Lowery, K. B. Englehart, H. Huang, G. Li, L. Hargrove, J. P. Dewald, and T. A. Kuiken, "Decoding a new neural–machine interface for control of artificial limbs," *Journal of neurophysiology*, vol. 98, no. 5, pp. 2974–2982, 2007.

[15] J. M. Hahne, F. Biessmann, N. Jiang, H. Rehbaum, D. Farina, F. C. Meinecke, K.-R. Müller, and L. C. Parra, "Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 2, pp. 269–279, 2014.

[16] R. Scott, "Myoelectric control of prostheses.," *Archives of physical medicine and rehabilitation*, vol. 47, no. 3, pp. 174–181, 1966.

[17] T. W. Williams III, "Practical methods for controlling powered upper-extremity prostheses," *Assistive Technology*, vol. 2, no. 1, pp. 3–18, 1990.

[18] J. N. Leijnse, N. H. Campbell-Kyureghyan, D. Spektor, and P. M. Quesada, "Assessment of individual finger muscle activity in the extensor digitorum communis by surface emg," *Journal of neurophysiology*, vol. 100, no. 6, pp. 3225–3235, 2008.

[19] L. Hargrove, K. Englehart, and B. Hudgins, "The effect of electrode displacements on pattern recognition based myoelectric control," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2203–2206, IEEE, 2006.

[20] J. Duchateau and R. M. Enoka, "Human motor unit recordings: origins and insight into the integrated motor system," *Brain research*, vol. 1409, pp. 42–61, 2011.

[21] C. J. Heckman and W. Z. Rymer, *Spinal Mechanisms for Control of Muscle Length and Force*, pp. 419–455. Totowa, NJ: Humana Press, 2003.

[22] C. J. De Luca, A. Adam, R. Wotiz, L. D. Gilmore, and S. H. Nawab, "Decomposition of surface emg signals," *Journal of neurophysiology*, vol. 96, no. 3, pp. 1646–1657, 2006.

[23] R. Merletti, A. Botter, A. Troiano, E. Merlo, and M. A. Minetto, "Technology and instrumentation for detection and conditioning of the surface electromyographic signal: state of the art," *Clinical biomechanics*, vol. 24, no. 2, pp. 122–134, 2009.

[24] A. Holobar and D. Zazula, "Multichannel blind source separation using convolution kernel compensation," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4487–4496, 2007.

[25] F. Negro, S. Muceli, A. M. Castronovo, A. Holobar, and D. Farina, "Multi-channel intramuscular and surface emg decomposition by convolutive blind source separation," *Journal of neural engineering*, vol. 13, no. 2, p. 026027, 2016.

[26] M. Chen and P. Zhou, "A novel framework based on fastica for high density surface emg decomposition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 1, pp. 117–127, 2015.

[27] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface," *Pattern Recognition Letters*, vol. 119, pp. 131–138, 2019. Deep Learning for Pattern Recognition.

[28] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface emg images," *Scientific reports*, vol. 6, no. 1, pp. 1–8, 2016.

[29] K.-T. Kim, C. Guan, and S.-W. Lee, "A subject-transfer framework based on single-trial emg analysis using convolutional neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 94–103, 2019.

[30] J. E. Downey, L. Brane, R. A. Gaunt, E. C. Tyler-Kabara, M. L. Boninger, and J. L. Collinger, "Motor cortical activity changes during neuroprosthetic-controlled object interaction," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[31] P. Heo, G. M. Gu, S.-j. Lee, K. Rhee, and J. Kim, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 5, pp. 807–824, 2012.

[32] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in neural information processing systems*, pp. 211–217, 1990.

[33] C. Dai and X. Hu, "Independent component analysis based algorithms for high-density electromyogram decomposition: Experimental evaluation of upper extremity muscles," *Computers in biology and medicine*, vol. 108, pp. 42–48, 2019.

[34] C. Dai and X. Hu, "Independent component analysis based algorithms for high-density electromyogram decomposition: Systematic evaluation through simulation," *Computers in biology and medicine*, vol. 109, pp. 171–181, 2019.

[35] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.

[36] Y. Ning, X. Zhu, S. Zhu, and Y. Zhang, "Surface emg decomposition based on k-means clustering and convolution kernel compensation," *IEEE journal of biomedical and health informatics*, vol. 19, no. 2, pp. 471–477, 2014.

[37] F. Xu, Y. Zheng, and X. Hu, "Real-time finger force prediction via parallel convolutional neural networks: A preliminary study," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3126–3129, IEEE, 2020.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] Y. Zheng and X. Hu, "Concurrent estimation of finger flexion and extension forces using motoneuron discharge information," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 5, pp. 1638–1645, 2021.

[41] Y. Zheng and X. Hu, "Real-time isometric finger extension force estimation based on motor unit discharge information," *Journal of neural engineering*, vol. 16, no. 6, p. 066006, 2019.

[42] C. Dai and X. Hu, "Extracting and classifying spatial muscle activation patterns in forearm flexor muscles using high-density electromyogram recordings," *International journal of neural systems*, vol. 29, no. 01, p. 1850025, 2019.

[43] C. Dai and X. Hu, “Finger joint angle estimation based on motoneuron discharge activities,” *IEEE journal of biomedical and health informatics*, vol. 24, no. 3, pp. 760–767, 2019.

[44] M. H. Schieber and M. Santello, “Hand function: peripheral and central constraints on performance,” *Journal of applied physiology*, vol. 96, no. 6, pp. 2293–2300, 2004.

[45] M. Santello and A. J. Fuglevand, “Role of across-muscle motor unit synchrony for the coordination of forces,” *Experimental brain research*, vol. 159, no. 4, pp. 501–508, 2004.

[46] J. Leijnse, S. Carter, A. Gupta, and S. McCabe, “Anatomic basis for individuated surface emg and homogeneous electrostimulation with neuroprostheses of the extensor digitorum communis,” *Journal of neurophysiology*, vol. 100, no. 1, pp. 64–75, 2008.

[47] M. Santello, G. Baud-Bovy, and H. Jörntell, “Neural bases of hand synergies,” *Frontiers in computational neuroscience*, vol. 7, p. 23, 2013.