



HAL
open science

A convolutional neural network to identify motor units from high-density surface electromyography signals in real time

Yue Wen, Simon Avrillon, Julio Hernandez-Pavon, Sangjoon Kim, François Hug, José Pons

► To cite this version:

Yue Wen, Simon Avrillon, Julio Hernandez-Pavon, Sangjoon Kim, François Hug, et al.. A convolutional neural network to identify motor units from high-density surface electromyography signals in real time. *Journal of Neural Engineering*, 2021, 18 (5), pp.056003. 10.1088/1741-2552/abeead . hal-03298395

HAL Id: hal-03298395

<https://hal.science/hal-03298395>

Submitted on 26 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

PAPER • OPEN ACCESS

A convolutional neural network to identify motor units from high-density surface electromyography signals in real time

To cite this article: Yue Wen *et al* 2021 *J. Neural Eng.* **18** 056003

View the [article online](#) for updates and enhancements.



PAPER

OPEN ACCESS

RECEIVED
17 July 2020

REVISED
26 February 2021

ACCEPTED FOR PUBLICATION
15 March 2021

PUBLISHED
6 April 2021

Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



A convolutional neural network to identify motor units from high-density surface electromyography signals in real time

Yue Wen^{1,2,7} , Simon Avrillon^{1,2,7} , Julio C Hernandez-Pavon^{1,2} , Sangjoon J Kim^{1,2} , François Hug^{3,4,5} and José L Pons^{1,2,6,*}

¹ Legs and Walking Lab, Shirley Ryan Ability Lab, Floor 24, 355 E Erie St, Chicago 60611 IL, United States of America

² Department of Physical Medicine and Rehabilitation, Feinberg School of Medicine, Northwestern University, Chicago, IL, United States of America

³ Nantes University, Laboratory 'Movement, Interactions, Performance' (EA 4334), Nantes, France

⁴ The University of Queensland, School of Biomedical Sciences, Brisbane, Australia

⁵ Institut Universitaire de France (IUF), Paris, France

⁶ Department of Mechanical Engineering and Department of Biomedical Engineering, McCormick School of Engineering, Northwestern University, Evanston, IL, United States of America

⁷ These authors are equally contributed to this work.

* Author to whom any correspondence should be addressed.

E-mail: jpons@sralab.org

Keywords: real-time decomposition, motor units, high-density surface electromyography, convolutional neural network, neural drive

Abstract

Objectives. This paper aims to investigate the feasibility and the validity of applying deep convolutional neural networks (CNN) to identify motor unit (MU) spike trains and estimate the neural drive to muscles from high-density electromyography (HD-EMG) signals in real time. Two distinct deep CNNs are compared with the convolution kernel compensation (CKC) algorithm using simulated and experimentally recorded signals. The effects of window size and step size of the input HD-EMG signals are also investigated. **Approach.** The MU spike trains were first identified with the CKC algorithm. The HD-EMG signals and spike trains were used to train the deep CNN. Then, the deep CNN decomposed the HD-EMG signals into MU discharge times in real time. Two CNN approaches are compared with the CKC: (a) multiple single-output deep CNN (SO-DCNN) with one MU decomposed per network, and (b) one multiple-output deep CNN (MO-DCNN) to decompose all MUs (up to 23) with one network. **Main results.** The MO-DCNN outperformed the SO-DCNN in terms of training time (3.2–21.4 s epoch⁻¹ vs 6.5–47.8 s epoch⁻¹, respectively) and prediction time (0.04 vs 0.27 s sample⁻¹, respectively). The optimal window size and step size for MO-DCNN were 120 and 20 data points, respectively. It results in sensitivity of 98% and 85% with simulated and experimentally recorded HD-EMG signals, respectively. There is a high cross-correlation coefficient between the neural drive estimated with CKC and that estimated with MO-DCNN (range of *r*-value across conditions: 0.88–0.95). **Significance.** We demonstrate the feasibility and the validity of using deep CNN to accurately identify MU activity from HD-EMG with a latency lower than 80 ms, which falls within the lower bound of the human electromechanical delay. This method opens many opportunities for using the neural drive to interface humans with assistive devices.

1. Introduction

The control of human movements relates to the activation of motor units (MUs), which combine a motor neuron and the muscle fibers it innervates. Thus, the force produced by a muscle depends on the neural drive it receives, defined here as the output of the motor neurons that innervate a muscle [1–3]. There is a growing interest of developing human–machine interfaces that decode the neural drive to control assistive devices and prostheses [4, 5]. Because muscle action potentials are generated before the production of force, inducing an electromechanical delay, it is possible to use myoelectric signals to predict motor intent and translate them into commands before the actual movement [6, 7]. To this end, the signal processing must be accurate and fast, typically shorter than the electromechanical delay observed during voluntary movements (i.e., 225 ± 50 ms [8]).

Bipolar surface electromyography (EMG) remains the classical technique to provide insight into muscle activation. However, because the innervation ratio (number of fibers within each unit) varies across MUs, the neural drive is not directly related to the number of muscle fiber action potentials that contribute to the EMG signal [3, 9, 10]. Indeed, larger MUs will have relatively more influence on the EMG signal amplitude than smaller MUs [9]. In addition, the lack of selectivity of surface EMG favors the detection of signals from neighboring muscles and makes the overall interpretation difficult [9, 11]. A possible solution to resolve these drawbacks is the direct estimation of the neural drive with algorithms that decompose the interference EMG into MU spike trains.

Numerous algorithms have been developed to decompose interference EMG signals. The first semi-automatic decomposition techniques have been applied to intramuscular EMG recordings [12], leading to the accurate identification of a small sample of MUs, which, however, is not sufficient to estimate the neural drive [13]. Recently, blind-source separation methods have taken advantage of high-density EMG recordings (HD-EMG) to accurately identify a large number of MUs (up to 50 MUs [14]). Indeed, the large number of recording channels enables the application of latent component analysis approaches to identify MU spike trains [15–17]. Several decomposition methods, such as the convolution kernel compensation algorithm (CKC; [15]) or progressive fast independent component analysis peel-off [17], have been developed and validated. Although automated versions of these methods have been developed to identify MUs without manual edition [16, 18], several steps such as signal extension or whitening increase the computational time and prevent the estimation of neural drive in real time.

Alternative approaches have been developed to identify motor MU spike trains in real time

[19–22]. They consist of the offline pre-training of an algorithm to identify the separation matrix, i.e. the filter that captures the MU properties and separates the overlapping action potentials. Then, this matrix is applied in real time on short segments of HD-EMG signals with or without overlapping [20–22]. For example, Chen *et al* [20] have accurately identified 12 ± 2 MUs during hand movements using EMG segments of 200 ms. In this experiment, the processing time, including signal recording and computation, was about 250 ms [20]. Recently, Clarke *et al* [19] have shown encouraging results using a neural network, which outperformed the CKC algorithm on signals with low signal-to-noise ratio [19]. The processing time was 67 ms for one second of HD-EMG data, which has the potential for real-time implementation. Similar approaches will require alternative machine learning methods with a frame-by-frame processing time below the human electromechanical delay to guarantee synchronous behavior between an assistive device and the user's movement [7, 23].

In this study, we propose the application of a deep convolutional neural network (CNN) to accurately estimate the neural drive to muscles from HD-EMG signals in real time. We aim to estimate the neural drive with a computational delay within the range of the human electromechanical delay. The CKC algorithm was applied to identify the MU spike trains and to train the deep CNN. Then, the deep CNN was used to decompose HD-EMG signals into MU spike trains in real time, which were summed and filtered to estimate the neural drive. We first validated the approach using simulated HD-EMG signals. Secondly, we evaluated the performance of the real-time MU identification (sensitivity and precision) and time cost (training time and prediction latency) with experimentally recorded signals and we compared the estimation of the neural drive to that derived from the offline CKC algorithm.

2. Deep CNN for motor unit identification

2.1. Problem statement

Methods to decompose interference EMG signals into MU spike trains constantly evolve to increase the accuracy and to decrease the time latency. Essentially, the problem is to identify the MU spike trains using temporal-spatial features of motor unit action potentials (MUAPs). Due to the stochastic nature of EMG signals, this information is usually extracted from a window of HD-EMG signals with a sliding-window approach [15, 22]. Here, we use a window of the HD-EMG signals as the input of the deep CNN for MU identification, and we try to minimize its length to perform real-time identification with minimal latency. The length of the sliding window is defined by the window size w . The width of the sliding window is defined by the number of EMG channels

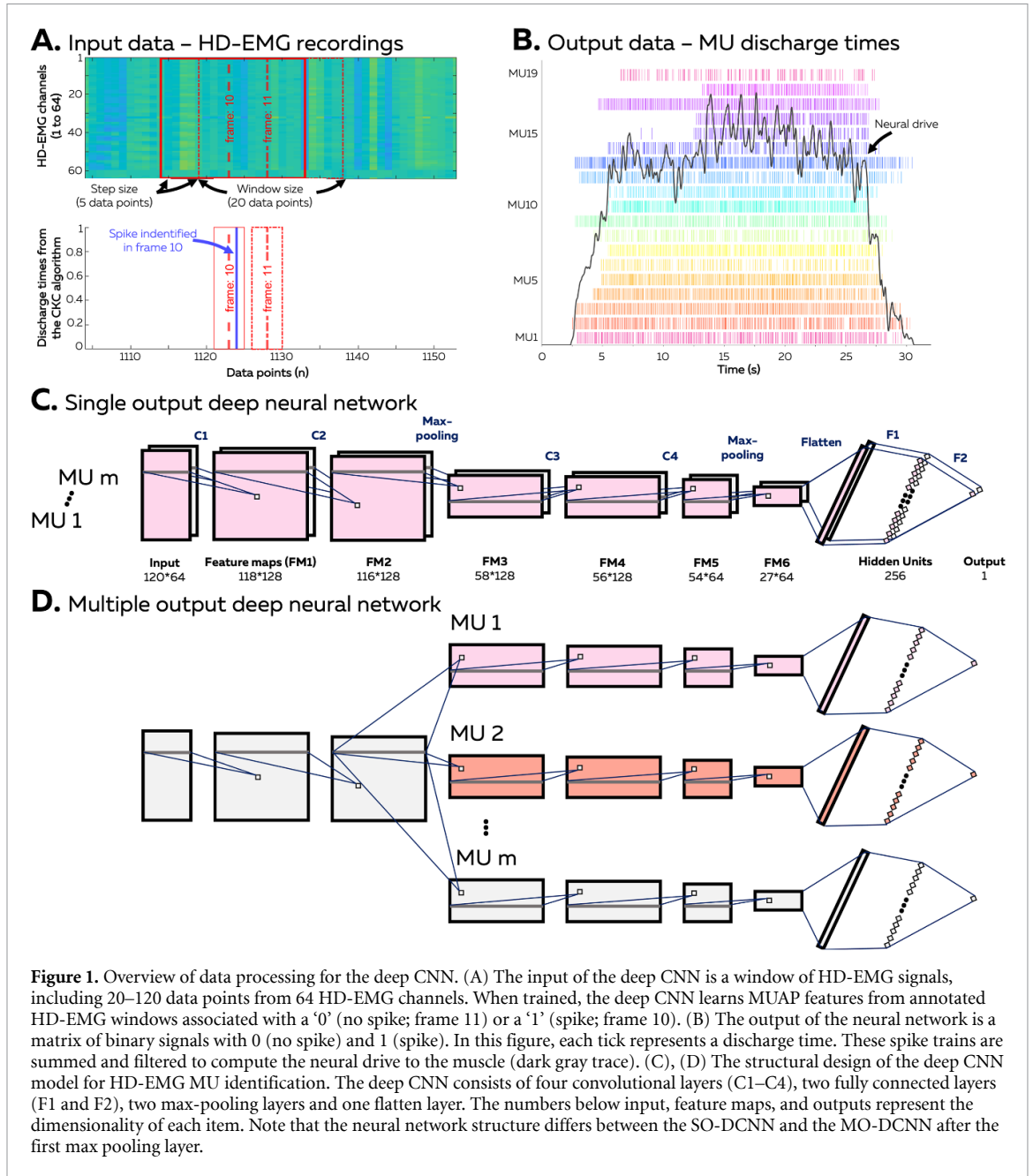


Figure 1. Overview of data processing for the deep CNN. (A) The input of the deep CNN is a window of HD-EMG signals, including 20–120 data points from 64 HD-EMG channels. When trained, the deep CNN learns MUAP features from annotated HD-EMG windows associated with a ‘0’ (no spike; frame 11) or a ‘1’ (spike; frame 10). (B) The output of the neural network is a matrix of binary signals with 0 (no spike) and 1 (spike). In this figure, each tick represents a discharge time. These spike trains are summed and filtered to compute the neural drive to the muscle (dark gray trace). (C), (D) The structural design of the deep CNN model for HD-EMG MU identification. The deep CNN consists of four convolutional layers (C1–C4), two fully connected layers (F1 and F2), two max-pooling layers and one flatten layer. The numbers below input, feature maps, and outputs represent the dimensionality of each item. Note that the neural network structure differs between the SO-DCNN and the MO-DCNN after the first max pooling layer.

M . The sliding distance/increment is defined as step size s (figure 1(A)).

The input of the deep CNN can be written as a $M \times w$ matrix,

$$\begin{aligned} \underline{X}_{s,w}(i) &= [x_{s,w}^1(i), x_{s,w}^2(i), \dots, x_{s,w}^M(i)]^T, \\ x_{s,w}^m(i) &= \left[x^m\left(i*s - \frac{w}{2} + 1\right), x^m\left(i*s - \frac{w}{2} + 2\right), \dots, x^m\left(i*s + \frac{w}{2}\right) \right]^T, \\ i &= 1, 2, 3, \dots, \end{aligned} \quad (1)$$

where \underline{X} is a window of HD-EMG signals, x is the EMG measurement and i is the window index. The

superscript m is the index of EMG channels, ranging from 1 to M .

The MU spike trains corresponding to each HD-EMG window can be written as a $N \times 1$ matrix,

$$\begin{aligned} \underline{Y}_{s,w}(i) &= \left[y_{-s,w}^1(i), y_{-s,w}^2(i), \dots, y_{-s,w}^N(i) \right]^T, \\ y_{-s,w}^n(i) &= \max \left(\left[y^n \left(i * s - \frac{s}{2} + 1 \right), y^n \left(i * s - \frac{s}{2} + 2 \right), \dots, y^n \left(i * s + \frac{s}{2} \right) \right]^T \right), \\ & i = 1, 2, 3, \dots, \end{aligned} \quad (2)$$

where \underline{Y} is a matrix including binary signals from multiple MUs, y is the MU binary signal obtained from the CKC algorithm, which is 1 (spike) or 0 (no spike); N is the total number of MUs identified with the CKC algorithm. The superscript n is the index of MUs, ranging from 1 to N .

Ultimately, the deep CNN predicts if a given MU is activated or not with an output value ranging from 0 to 1 for each window. An output value exceeding a threshold of 0.5 is considered as a spike.

2.2. Structure of the deep CNN model

We propose two deep CNN approaches to identify multiple MU spike trains from HD-EMG signals and estimate the neural drive. The first approach consists of dividing the task into a few simple tasks by constructing multiple single-output deep CNN (SO-DCNN) with one MU identified per network (figure 1(C)). The second approach consists of building one multiple-output deep CNN (MO-DCNN) to identify all MUs with one network (figure 1(D)). To this end, we adopted and modified the deep CNN structure introduced in a previous work on activity recognition from accelerometer data [24]. The structures and hyperparameters were manually optimized using experimental data from one participant. Specifically, we compared three neural networks with different numbers of layers (Case 1: two fully connected layers, Case 2: two convolutional layers and two fully connected layers; Case 3: four convolutional layers and two fully connected layers) and three neural networks with different numbers of nodes (Case 1: 64-64-64-32-128 nodes per layer; Case 2: 128-128-128-64-256 nodes per layer; Case 3: 256-256-256-128-512 nodes per layer). The deep CNN described in this section reached the best performance and was used for this study.

For the SO-DCNN, we used one window of the HD-EMG signal, where the width of the window was M (number of HD-EMG channels) and the length was w (e.g. 120 data points from each channel when the window size is equal to 120) as the input of the deep CNN. The output of the deep CNN was a binary value, indicating the discharge times of a given MU (figure 1(B)). In between, we implemented six parametric layers (four convolutional layers and two fully

connected layers) and three non-parametric layers (two max-pooling layers and one flatten layer). The first three convolutional layers (C1–C3) included 128 parallel kernel filters with a kernel size of 3 to extract the temporal-spatial features, while the last convolutional layer (C4) had 64 kernel filters to decrease the number of trainable parameters of the fully connected layers. The output of convolutional layers (FM6) was flattened to a vector and was fed into two fully connected layers, including 256 nodes and 1 node, respectively. We used a sigmoid activation function for the output layer and a rectified linear unit (ReLU) activation function for all other layers. The sigmoid activation function had an output ranging from 0 to 1, which fitted our output requirement of being a binary value. The ReLU solved the vanishing gradient problem and accelerated the convergence speed. We designed the consecutive convolutional layers (C1 and C2, C3 and C4) to allow the CNN model to learn more complex features. Max-pooling with size of 2 was applied to C2 and C4 to extract representative features while reducing the dimension of the features. Then, the fully connected layers made prediction or classification based on the feature maps from convolutional layers. In addition, dropout with a rate of 50% was applied to C3, F1, and F2 layers to regularize the neural network and to prevent overfitting.

In the case of the MO-DCNN, we used an identical structure for the first three layers (C1, C2, max-pooling) to the SO-DCNN, but we added multiple parallel layer sets (C3–F2), one for each MU (figure 1(D)). For example, to decompose the discharge times of four MUs, the deep CNN included one set of C1–C3 to generate FM3, and four sets of C3–F2, taking FM3 as input and predicting the discharge times for each MU. The number of outputs was determined dynamically by the number of MU used in the training dataset.

2.3. Implementation

We implemented the framework with Tensorflow [25] and Keras library (<https://keras.io/> [26]). Keras functional application programming interface was used to create customized multiple outputs models. As

the goal is to predict the discharge times of an MU, which is a binary state, we used a binary cross-entropy loss function. Then RMSprop optimizer, one of the most influential adaptive stochastic algorithms, was used to update the network weights [27]. We used the F1-score, calculated as $2 \times (\text{sensitivity} \times \text{precision}) / (\text{sensitivity} + \text{precision})$, to determine the performance of the deep CNN during the training process and to adopt the best model. For this purpose, a customized metrics function was implemented to calculate the averaged F1-score across multiple MUs for MO-DCNN at the end of each epoch. The detailed implementation with Python can be found at https://github.com/ywen3/dcnm_mu_decomp.

3. Evaluation methods

3.1. Simulation of HD-EMG signals

The validation of algorithms that identified MU spike trains requires a comparison of MU spike trains between the proposed method and a gold standard method. As no gold-standard method exist for surface HD-EMG signals, we used simulated HD-EMG signals to compare the MU spike trains identified by the CKC algorithm and the deep CNN to the ground truth, i.e. the spike trains of simulated MU. To this end, we have created a cylindrical volume containing three layers (muscle, fat, and skin) with a radius of 10 mm, 4 mm and 1 mm, respectively, and an average fiber length of 200 mm [28–30]. The muscle had 125 664 fibers parallel to the skin that were grouped into 150 MUs. According to the size principle [31], the number of innervations per MU ranged from 52 to 3191, the recruitment thresholds ranged from 0% to 75% of maximal excitation level, and the conduction velocities were normally distributed with a mean value of $3.8 \pm 0.5 \text{ m s}^{-1}$. The recruitment thresholds of the 150 motor neurons followed an exponential distribution, with most of the motor neurons recruited at the lowest intensities [32]. The MU discharge rate at the recruitment ranged from 5 to 10 pulses per second (pps) and linearly increased until the maximal discharge rate. The maximal discharge rate ranged from 30 to 40 pps.

We have simulated a trapezoidal contraction with a plateau of 60 s at 25% of maximal excitation level and a ramp-up with an increase in excitation of 2.5% per second. At 25% of maximal excitation level, 110 MUs were recruited. The variability of discharge rate followed a Gaussian random process with a coefficient of variation of the inter-spike interval set at 15% [32]. The MUAPs have been simulated as detected by a two-dimensional grid of 65 electrodes (13×5) with a radius of 1 mm and an inter-electrode distance of 8 mm. The center of the grid was positioned over the center of the muscle and aligned with the fiber direction. The surface EMG signals were the algebraic sum of muscle fiber action potentials with an additional zero-mean Gaussian noise

with signal-to-noise ratio of 20 dB. A single differential recording was simulated for each longitudinal pair of adjacent electrodes, resulting in 60 simulated detection points. The EMG signals were computed at 2048 Hz.

3.2. Collection of experimental HD-EMG signals

3.2.1. Participants

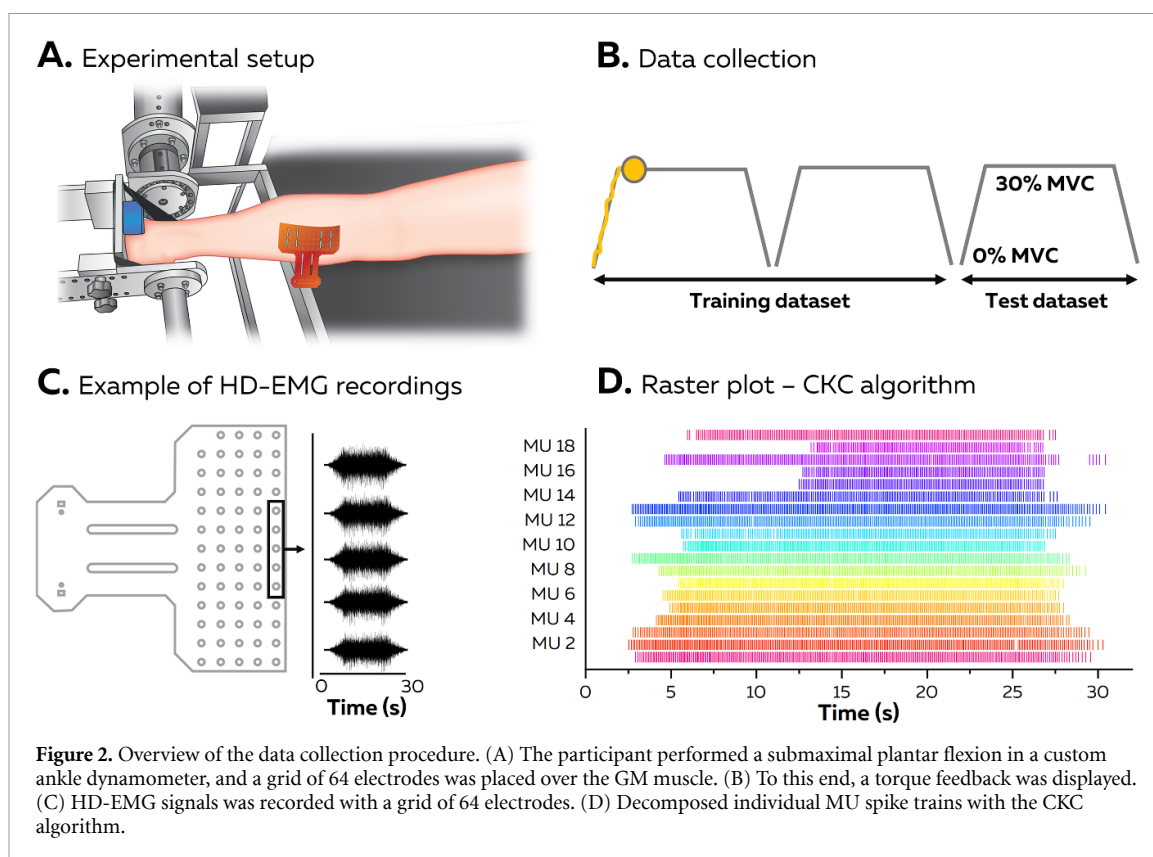
Six physically active males volunteered to participate in this study (mean \pm standard deviation; age 34 ± 8 years, height 178 ± 6 cm, body mass 76 ± 7 kg). Participants had no history of lower leg injury or pain within the previous 6 months that could prevent their full engagement during the experiments. The institutional research ethics committee of the University of Queensland approved this study (No. 2013001448), and all procedures were in accordance with the Declaration of Helsinki. All participants provided their written informed consent prior to participation in the study. Note that the experimental data were a subset of a previously published dataset [33].

3.2.2. Experiments

Participants laid prone on a custom-made dynamometer, which consisted of a bench equipped with a foot pedal and a torque sensor (TRE-50K, Dacell, Rep. of Korea). They positioned their knee in full extension, and their ankle was securely fastened at a plantar flexion angle of 10° (0° being the foot perpendicular to the shank). The experimental setup is shown in figure 2(A). After performing a warm-up set, which included a series of submaximal contractions, participants performed three maximal isometric voluntary contractions (MVC) for 3–5 s with 120 s rest in between. We considered the maximal torque value obtained from a moving average window of 250 ms as the peak torque, and we used this value to set up the torque feedback during the submaximal isometric contractions. Then, participants performed three contractions at each of the following intensities: 10%, 30%, and 50% of their MVC (figure 2(B)). The order of the intensities was randomized. These contractions involved a 5 s ramp-up, a 15 s (50% MVC) or 20 s plateau (10% and 30% MVC), and a 5 s ramp-down phase. In total, 75 s of signals at 50% MVC and 90 s of signals at 10% and 30% MVC were considered for analysis. The contractions were separated by either 60 s (10% MVC) or 120 s (30% and 50% MVC) to reduce the effect of muscle fatigue on HD-EMG signals [10]. The torque signal was digitized at 2048 Hz using the same acquisition system as that used for HD-EMG (EMG-Quattrocento; 400-channel EMG amplifier, OT Bioelettronica, Italy) and was low-pass filtered (Butterworth 3rd order, cut-off frequency: 20 Hz).

3.2.3. High-density surface EMG recordings

A two-dimensional adhesive grid of 64 electrodes (13×5 electrodes with one electrode absent on a



corner, gold-coated, inter-electrode distance: 8 mm; [ELSCH064NM2, SpesMedica, Battipaglia, Italy]) was placed over the gastrocnemius medialis muscle. The grid was aligned with the main fascicle direction as determined using B-mode ultrasound (Aixplorer, Supersonic Imagine, France). Before applying the electrode, the skin was shaved and cleaned with an abrasive pad and alcohol. The adhesive grid was held on the skin using semi-disposable bi-adhesive foam layers (SpesMedica, Battipaglia, Italy). The skin-electrode contact was made by filling the cavities of the adhesive layers with conductive paste (SpesMedica, Battipaglia, Italy). An elastic band was placed over the electrode with slight tension to ensure that it remained in contact with the skin throughout the experiment. Strap electrodes dampened with water were placed around the contralateral (ground electrode) and ipsilateral ankle (reference electrode). The EMG signals were recorded in monopolar mode, band-pass filtered (10–900 Hz), and digitized at a sampling rate of 2048 Hz using a multichannel acquisition system (EMG-Quattrocento; 400-channel EMG amplifier, OT Bioelettronica, Italy).

3.3. Data processing

3.3.1. HD-EMG decomposition with the CKC algorithm

The HD-EMG signal decomposition of the simulated and experimentally recorded signals was carried out with the CKC algorithm [15] implemented in the DEMUSE tool software (v4.9; The University of Maribor, Slovenia) developed on MATLAB (The

Mathworks, Inc, Natick, Massachusetts, USA). The EMG signals were band-pass filtered (Butterworth 2nd order, 20–750 Hz) before the decomposition. The decomposition procedure has been validated using simulated and experimentally recorded signals [16, 34–36]. After the automatic identification of the MUs, all the MU spike trains were visually inspected and manually edited when a false positive (FP; labeled artifact) or a false negative (FN; non-labeled discharges) were observed by an experienced operator [37]. Only the MUs that exhibited a pulse-to-noise ratio (PNR) >30 dB were retained for further analysis [34]. This threshold ensured a sensitivity higher than 90% and a false-positive rate lower than 2% [34]. In addition, MU spike trains with less than 150 spikes in any of the three segments (i.e. contractions) were not considered for the analysis.

3.3.2. Data pre-processing

The HD-EMG signals were normalized to the range of -1 and 1 with the maximal amplitude recorded on the entire grid to retain the temporal-spatial information (e.g. relative magnitude across different channels). The normalization procedure aims to accelerate the learning process by limiting the range of input values such that the weight updates calculated through backpropagation are reasonable. The HD-EMG and spike trains of decomposed MUs were further segmented into windows (with specific window size and step size) based on equations (1) and (2) for deep CNN training and validation.

3.3.3. Training and validation protocol

The training and validation of the deep CNN were done for each dataset. Specifically, three-fold cross-validation was performed to evaluate the robustness of the deep CNN. Instead of randomly dividing the data set into three-folds, we treated each segment as one-fold (figure 2(B)). The deep CNN model was trained with two segments of data and validated with the remaining segment to get the model performance (figure 2(B)). This procedure was repeated three times, each with a different segment as the validation set. The overall performance was computed by averaging the model performance from three different segments.

In the training stage, we used a batch size of 64 to accelerate the training of deep CNN and terminated the training process if the training loss had no improvement for 50 consecutive epochs or the training process reached 100 epochs. The timestamp and F1-score of each epoch during training were recorded in the TensorBoard logs for post analysis, and the model that reached the highest F1-score along the training epochs was saved. In the validation stage, we first loaded the best model with the highest F1-score; then we fed the model with the validation data segment and got the spike trains of the MU. We ultimately saved the spike trains identified by the deep CNN and the time elapsed for each prediction for further analysis.

3.3.4. Validation strategies

In order to use the estimation of neural drive in human-machine interfaces, the deep CNN should decompose HD-EMG into multiple MU spike trains with an excellent sensitivity, a low rate of FPs, and a latency in the range of the electromechanical delay. To this end, we first compared the performance of the SO-DCNN and the MO-DCNN to determine the most efficient structure. Then, we determined how the format of the input data (i.e. window size and step size) influenced the performance of the deep CNN, including prediction outcomes and time efficiency.

3.3.4.1. Comparison between SO-DCNN and MO-DCNN

As a pilot study, we used datasets from the simulation and from participant #1 during a plantar flexion performed at 10% MVC to train the two deep CNNs and validate their performance. As there were 17 and 10 MUs identified by the CKC algorithm from the simulated and experimentally recorded signals, respectively, we constructed 17 and 10 independent SO-DCNNs and two MO-DCNNs with 17 and 10 outputs. The inputs were the windows of HD-EMG ($\underline{X}_{s,w}$) for both the SO-DCNN and the MO-DCNN. The outputs for the SO-DCNN were 17 and 10 vectors of discharge times of the corresponding MUs, while the outputs of the MO-DCNN were two matrices of discharge times for all MUs. Based

on prior knowledge, we compared the performance under four representative test conditions (step size of 5, 10, 20, and 40 data points corresponding to 2, 5, 10, and 20 ms at a frequency of 2048 Hz, and window size of 120 data points corresponding to 59 ms at a frequency of 2048 Hz).

3.3.4.2. Effects of window size and step size

Between the two deep CNN approaches (SO-DCNN and MO-DCNN), the approach that generated the best performance was used to systematically study the effects of window size and step size.

The window size defines the number of data points as input for each prediction. On one hand, according to equations (1) and (2), the deep CNN requires HD-EMG at $i \times s + \frac{w}{2}$ to predict the discharge activity of an MU at time instance $i \times s$, which causes a latency of half of the window size (figure 1(A)). Therefore, a smaller window size decreases the latency. On the other hand, the window size needs to be large enough to cover one MUAP for identification purposes. We selected four window sizes (i.e. 20, 40, 80, and 120 data points; corresponding to 10, 20, 39, and 59 ms at a frequency of 2048 Hz). This is (a) to account for the average length of the MUAP (about 10 ms [15]), and (b) to stay within the range of the muscle electromechanical delay (225 ± 50 ms [8]). For simplicity, we will use the number of data points as the unit in the rest of the manuscript.

The step size is the number of data points that differs between two consecutive predictions (difference/overlap between two consecutive windows). Hence, the step size determines the prediction frequency or cycle time for control purposes (e.g. a step size of 10 ms results in a prediction frequency of 100 Hz). With the sample frequency of 2048 Hz, we chose four step sizes (i.e. 5, 10, 20, and 40 data points) corresponding to prediction frequency of 410, 205, 102, and 51 Hz, respectively.

3.4. Evaluation criteria

We first validated our approach with simulated HD-EMG signals by comparing the spike trains identified by the deep CNN with the spike trains of simulated MU using sensitivity and precision. Then, we evaluated the performance of the deep CNN on experimental HD-EMG signals according to: (a) the accuracy of the deep CNN, including the number of identified MUs, the precision, and the sensitivity, (b) the training time and prediction time, and (c) the cross-correlation between neural drive estimations from the CKC algorithm and the deep CNN.

3.4.1. Accuracy of the deep CNN

We first used the F1-score to assess how successfully the MUs were identified by the deep CNN. We counted an MU as successfully identified by the deep CNN if the F1-score was greater than 0.7 [19].

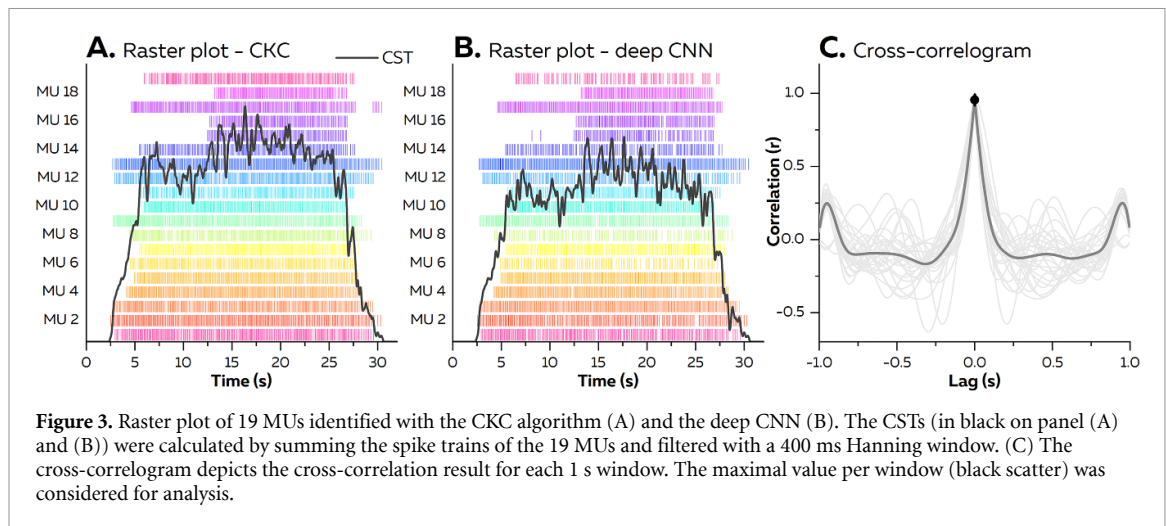


Figure 3. Raster plot of 19 MUs identified with the CKC algorithm (A) and the deep CNN (B). The CSTs (in black on panel (A) and (B)) were calculated by summing the spike trains of the 19 MUs and filtered with a 400 ms Hanning window. (C) The cross-correlogram depicts the cross-correlation result for each 1 s window. The maximal value per window (black scatter) was considered for analysis.

We compared the MU spike trains obtained from the deep CNN, the CKC algorithm, and the simulation with the calculation of the precision and the sensitivity [38]:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100\%, \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%, \quad (4)$$

where true positive is the number of correctly identified discharge times, FN is the number of non-identified discharges, and FP is the number of incorrectly identified discharges. Note that we first realigned the spike trains from the simulation with the spike trains from the CKC algorithm to consider the delay between the spike generated at the motor neuron level (simulated signal) and the spike detected at the skin level (CKC and deep CNN). Then, the spike trains obtained from either simulated signal or CKC method were down sampled for each step size to match the prediction frequency of the deep CNN (i.e. 410, 205, 102, and 51 Hz, for step sizes of 5, 10, 20, and 40 data points, respectively). For the simulation study, we used simulated spike trains to assess the accuracy of both the CKC and the deep CNN method; for the experimental study, we used spike trains from the CKC method to assess the accuracy of the deep CNN.

3.4.2. Training time and prediction time

The computational complexity was quantified by the training time and prediction time. Training time is defined as the time elapsed for each epoch (i.e. s epoch⁻¹) to process all training data and update the network during the training stage. Thus, we loaded the timestamp of each epoch from the log files and calculated the time elapsed for each epoch. The prediction time is defined as the time elapsed to make a prediction from a window of HD-EMG (i.e. s sample⁻¹) during the validation stage. For the SO-DCNN approach, the training time and prediction

time were considered as the sum of the time of all the SO-DCNNs. Training and testing stages were run on a PC with an Intel Xeon W-2123 CPU, a NVIDIA Quadro P1000 GPU, and 32 GB of RAM.

3.4.3. Comparison of neural drive estimations

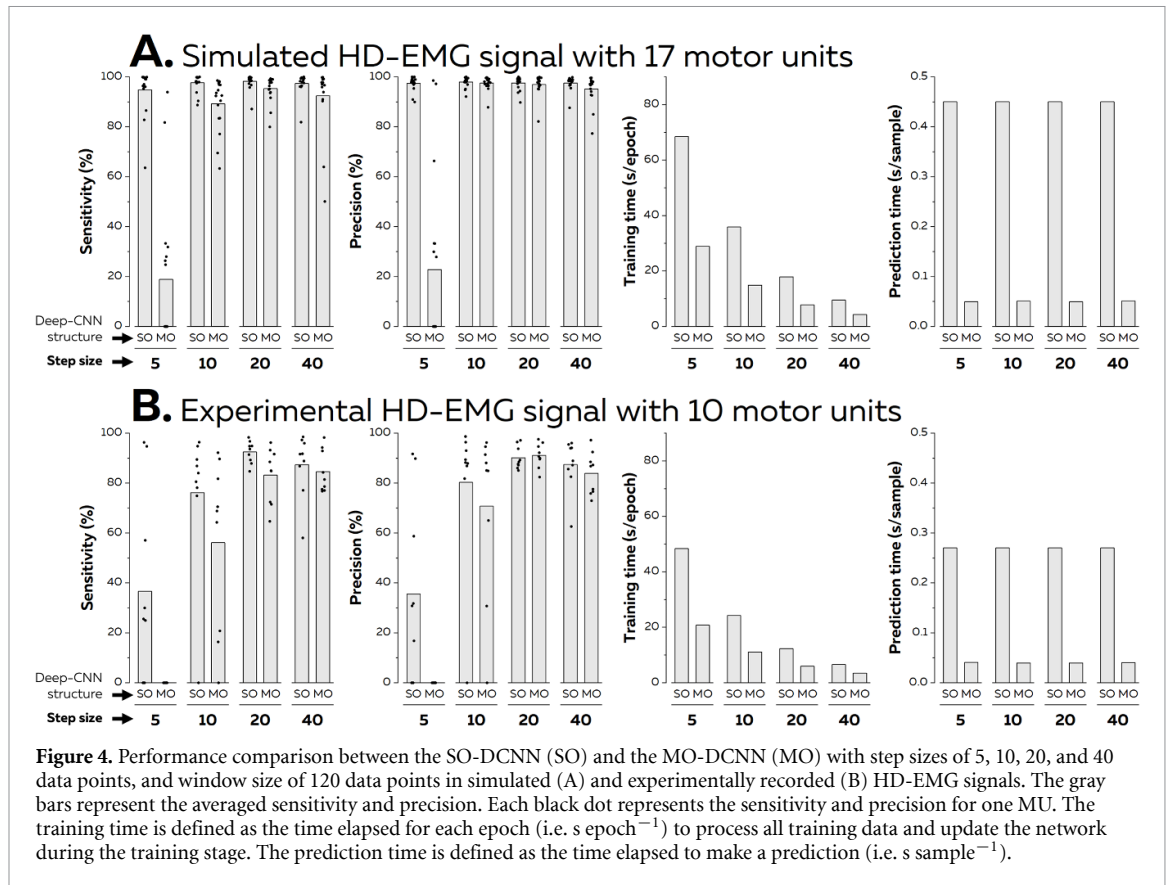
We considered the neural drive to the muscle as the sum of the spike trains of all identified MUs also known as the cumulative spike train (CST). To generate the CST, the individual MU spike trains were summed and filtered with a Hanning window of 400 ms [39]. This procedure was applied to the set of MUs identified with the simulation, the CKC algorithm and the deep CNN. Then, we calculated the cross-correlation coefficient between the neural drive estimated with the different methods on 1 s segments without overlapping (figure 3). The averaged maximal cross-correlation coefficients for an entire contraction were considered for analysis.

4. Results

4.1. Effect of the network structure (SO-DCNN vs MO-DCNN) on the deep CNN performance

We accurately identified 17 out of the 150 simulated MUs using the CKC algorithm (averaged PNR of 35.9 dB) with sensitivity and precision higher than 99%. This allowed us to consider the MU spike trains predicted by the CKC algorithm as a highly reliable estimation and an excellent standard to assess the performance of our deep CNNs. The MO-DCNN achieved comparable sensitivity and precision as the SO-DCNN with step sizes of 10, 20, and 40 data points (figure 4(A)). With a step size of 20 data points, both SO-DCNN and MO-DCNN identified 17 MUs and reached the highest sensitivity (98% for SO-DCNN and 95% for MO-DCNN) and precision (98% for SO-DCNN and 97% for MO-DCNN).

With the experimental data, we identified 10 MUs using the CKC algorithm on HD-EMG signals recorded during a plantar flexion at 10% MVC



(an averaged PNR of 35 dB). Both SO-DCNN and MO-DCNN approaches successfully identified 9 MUs. The sensitivity and precision of both approaches were affected by the step size (figure 4(B), left panels). Both achieved the highest performance with the step size of 20 (figure 4(B)). When compared to the SO-DCNN at a step size of 20 data points, the MO-DCNN reached comparable precision ($91 \pm 5\%$ vs $90 \pm 5\%$) but lower sensitivity ($83 \pm 11\%$ vs $92 \pm 4\%$). One reason for the better performance of the SO-DCNN is the greater number of hyperparameters per network specifically optimized for each MU (590 785 vs 524 199 parameters per MU) due to the independent training of the two first feature maps (figure 1(C)).

However, the training time of the SO-DCNN was doubled when compared to the MO-DCNN, and the prediction time was significantly larger for the SO-DCNN than for the MO-DCNN (figures 4(A) and (B)). Specifically, in the experimental study, the training time ranged from 6.5 to 48.3 s epoch⁻¹ for the SO-DCNN and from 3.4 to 20.7 s epoch⁻¹ for the MO-DCNN (figure 4(B), right panels); the prediction time ranged from 0.27 to 0.28 s sample⁻¹ for the SO-DCNN and from 0.040 to 0.041 s sample⁻¹ for the MO-DCNN (figure 4(B), right panels). Indeed, the SO-DCNN processed each frame of HD-EMG ten times to predict the discharge activities of ten MUs, while the MO-DCNN processed each frame only once. In addition, the time that the Keras library

used to handle logging and memory management is accumulated with the SO-DCNN structure. Considering that the prediction time of the SO-DCNN exceeded the range of physiological electromechanical delay, thus preventing its use in real time, we only investigated the effects of window size and step size on the performance of MU identification with the MO-DCNN.

4.2. Effect of the input format (window size and step size) on the deep CNN performance

4.2.1. Validation with simulated signals

The deep CNN was able to identify the 17 MUs and reached its best performance with a step size of 20 data points and a window size of 120 data points, achieving a precision of 95% and a sensitivity of 97% (figure 5). The sensitivity and precision generally increased with window size, except with window sizes of 80 and 120 data points with a step size of 5 data points, where the sensitivity decreased (figure 5). However, there was not clear and consistent trends with changes in step size. The sensitivity increased with the step size from 5 data points to 20 data points (figure 5(A)), but the precision decreased slightly with step size (figure 5(B)). It is noteworthy that the sensitivity and precision dropped for a window size of 20 data points. This may be because the window is not long enough to capture the properties of a MUAP within the grid of electrodes. In addition, the absence of overlap between

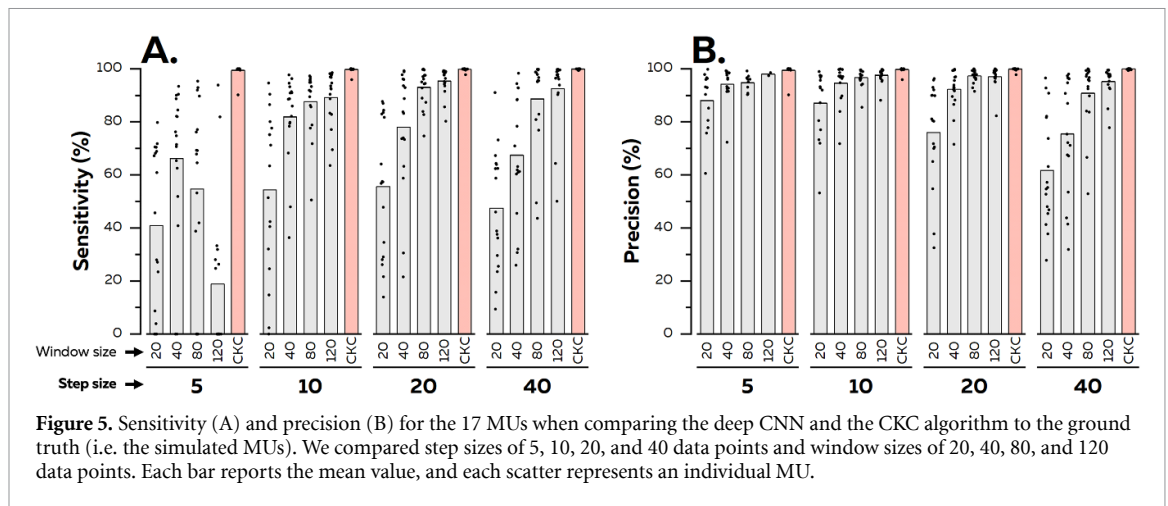


Figure 5. Sensitivity (A) and precision (B) for the 17 MUs when comparing the deep CNN and the CKC algorithm to the ground truth (i.e. the simulated MUs). We compared step sizes of 5, 10, 20, and 40 data points and window sizes of 20, 40, 80, and 120 data points. Each bar reports the mean value, and each scatter represents an individual MU.

Table 1. MUs identified with the CKC algorithm to train the deep CNN. This table shows the number of MUs identified with the DEMUSE software, the PNR of MUs identified by the CKC algorithm, and the number of MUs we used to train the deep CNN (i.e. with more than 150 spikes per contraction).

Intensity	10% MVC	30% MVC	50% MVC
Number of MUs (n)	22 ± 5	23 ± 4	15 ± 5
PNR (dB)	40 ± 4	39 ± 3	40 ± 3
Number of MUs for deep CNN training (n)	13 ± 4	18 ± 4	13 ± 4

consecutive windows (step size 20 and window size 20; step size 40 and window size 40) or the presence of a gap between consecutive windows (step size 40 and window size 20) could decrease the ability of the deep CNN to identify action potentials. Finally, the averaged cross-correlation coefficients between the neural drive estimated with the simulated data and the deep CNN reached 0.84 ± 0.09 , 0.94 ± 0.05 , 0.95 ± 0.05 and 0.91 ± 0.08 for a step size of 5, 10, 20 and 40 data points, respectively. The cross-correlation coefficients between the neural drive estimated with the simulated data and the CKC were higher than 0.99 for all the step sizes.

4.2.2. Experimentally recorded HD-EMG decomposition with the CKC algorithm

We identified 355 MUs from six participants with an averaged PNR of 39.6 ± 3.0 dB using the CKC algorithm. Specifically, the number of MUs ranged from 17 to 30 at 10% MVC, from 17 to 27 at 30% MVC, and from 8 to 21 at 50% MVC. For the analysis, we only considered the MUs with a minimum of 150 discharge times over each contraction to train the deep CNN. The number of MUs for each contraction intensity is summarized in table 1.

4.2.3. Effect of window size and step size on experimentally recorded signals

The performance of the MO-DCNN across six participants and three contraction intensities (i.e. 10%, 30%, and 50% MVC) is presented in figure 6. Step

size showed great effects on MU count, sensitivity, and precision. The sensitivity and precision increased when the step size increased from 5 data points to 20 data points, and when the window size increased for step sizes larger than 5 data points (figure 6). The number of identified MUs, sensitivity and precision were maximized with a step size of 20 data points and a window size of 120 data points (figure 6). The lower performance with a step size of 5 data points can be explained by the absence of significant differences between two consecutive windows. Therefore, the deep CNN cannot distinguish the differences between these HD-EMG windows, which is further confirmed by the oscillating loss function during the training period (unpublished data). In this case, this means that the deep CNN cannot optimize its parameters to improve the identification of MUAPs. The number of identified MUs was 13 ± 4 (range 6–22), 89% of the averaged total number of MUs (i.e. 15). The sensitivity and precision were $84 \pm 5\%$ and $93 \pm 2\%$, respectively. Note that we only considered the MUs with a F1-score above 0.7 [19]. As a comparison, a F1-score threshold of 0.8 led to a number of 11 ± 4 identified MUs, 73% of the averaged total number of MUs (i.e. 15) and sensitivity and precision of $86 \pm 3\%$ and $94 \pm 2\%$, respectively.

With a window size of 120 data points and a step size of 20 data points, the precision for each contraction intensity was $95 \pm 2\%$ for 10% MVC, $93 \pm 2\%$ for 30% MVC, and $91 \pm 2\%$ for 50% MVC across the six participants. The sensitivity for each contraction intensity was $88 \pm 3\%$ for 10% MVC, $83 \pm 6\%$ for 30% MVC, and $82 \pm 2\%$ for 50% MVC.

As shown in table 2, we observed a negative relationship between the training time and the step size (i.e. increasing step size decreased the training time) and a positive relationship between the training time and the window size (i.e. increasing window size increased the training time). However, the averaged prediction time for each window was about 50 ms and did not show a clear relationship with the window size and step size.

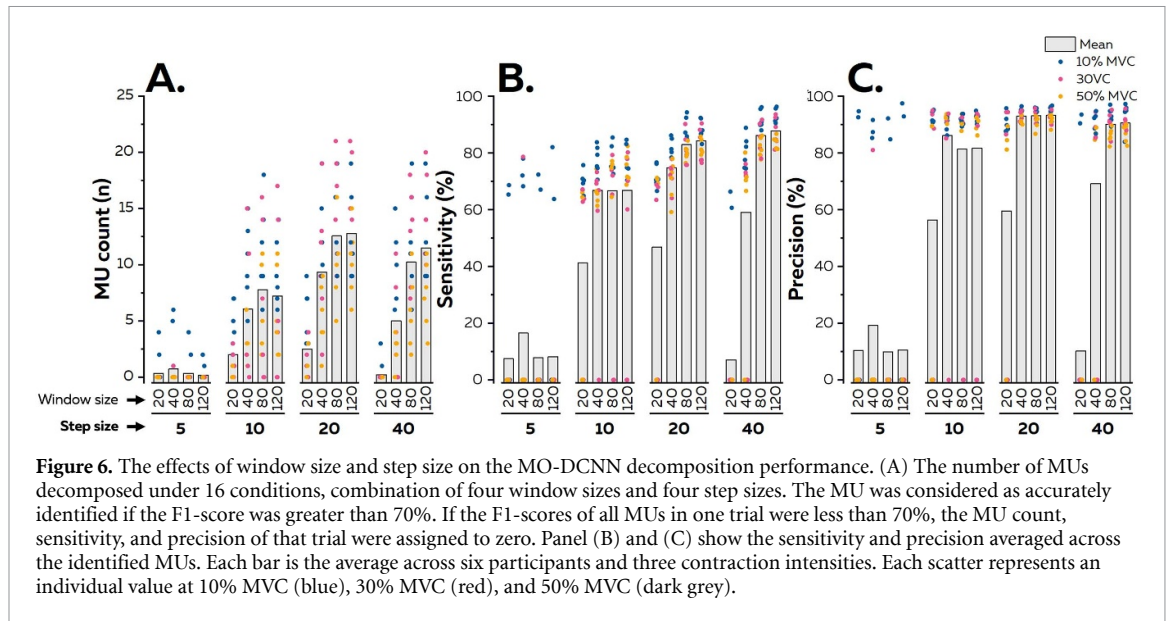
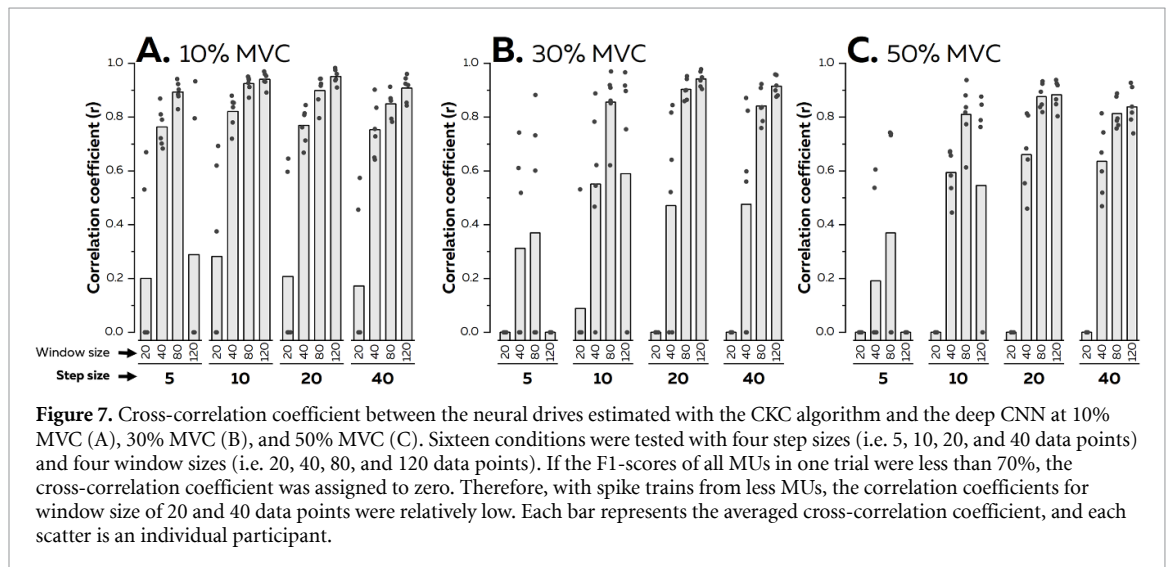


Table 2. The effect of window size and step size on deep CNN computational performance. This table shows the training time and prediction time of MO-DCNN for 16 conditions, combination of four window sizes and four step sizes. The training time for each epoch was averaged across all participants and contraction intensities.

Step size (data points)	5				10				20				40			
	20	40	80	120	20	40	80	120	20	40	80	120	20	40	80	120
Training time (s epoch ⁻¹)	13	16	22	28	7	9	11	15	4	5	6	8	3	3	4	5
Prediction time (ms sample ⁻¹)	46	46	49	49	46	46	48	48	46	46	49	48	46	46	48	48



4.2.4. Assessment of neural drive

All spike trains accurately identified with either the CKC algorithm or the MO-DCNN were summed and filtered with a Hanning window of 400 ms to compute the CSTs, an estimation of the neural drive to the muscle (figures 3(A) and (B)). Figure 7 depicts the cross-correlation coefficients between the CSTs. Regardless of the contraction intensity, the

cross-correlation coefficients were higher with step sizes of 20 and 40 data points and window sizes of 80 and 120 data points. For example, the averaged cross-correlation coefficients calculated at 10% MVC reached $r = 0.90 \pm 0.06$, $r = 0.95 \pm 0.03$, $r = 0.85 \pm 0.05$ and $r = 0.91 \pm 0.05$ with conditions SS20-WS80, SS20-WS120, SS40-WS80, and SS40-WS120, respectively. The highest cross-correlation

coefficients were obtained with a step size of 20 data points and a window size 120 data points ($r = 0.95 \pm 0.03$ at 10% MVC, $r = 0.94 \pm 0.03$ at 30% MVC, and $r = 0.88 \pm 0.05$ at 50% MVC). These results indicate that the estimation of the neural drive was strongly comparable between the CKC algorithm and the proposed MO-DCNN for the previously mentioned four conditions.

5. Discussion

In this study, we show that a deep CNN allows to accurately estimate the neural drive to muscles with the real-time identification of MUs discharge times during submaximal isometric contraction. Specifically, we demonstrate that a deep CNN with a single input (i.e. a window of HD-EMG signals) and multiple outputs (i.e. discharge times of multiple MU) minimizes the latency between the HD-EMG signals and the predicted discharge times (50 ms latency for the MO-DCNN vs 500 ms latency for the SO-DCNN). In addition, the identification accuracy is maximized when the deep CNN uses an input window of 120 data points resulting in precision higher than 90%, sensitivity higher than 80%, and correlation coefficients higher than 0.85, regardless of the contraction intensity. Overall, the total latency of the deep CNN is 80 ms, which is more than two times faster than previous approaches [20]. These results show the feasibility to accurately estimate the neural drive to muscles with a latency within the lowest boundaries of the human electromechanical delay (i.e. from 70 to 385 ms in [8]), opening new perspectives for human-machine interfaces.

We developed a deep CNN to perform the identification of MU spike trains. It is noteworthy that Clarke *et al* [19] have recently validated a similar approach with a two-source method, comparing the performance of a gated recurrent unit network with the gold-standard decompositions from intramuscular EMG recordings and simulations. They used windows of 80 data points of HD-EMG signals as the input of the neural network, leading to a processing time of 67 ms for one second of data. However, due to the different setups and testing conditions, we cannot directly compare the performance of the two implementations. In our settings, we found that windows of 120 data points were sufficient to reach sensitivity and precision higher than 80%, leading to a latency of 80 ms (duration of a half-window plus the prediction time). Thus, the proposed deep CNN outperforms all the previous approaches in terms of predicting MU spike trains in real time (250 ms in [20]; 1 s in [21]; 3 s in [22]). One reason is the ability of the proposed neural network to better identify MUs from the EMG signals with low signal-to-noise ratio [19], which enables to reduce the duration of data segments. In addition, the identification of MU discharge times

without additional preprocessing steps such as signal extension and whitening, which are necessary with blind-source separation approaches, decreased the computational time [15, 16]. However, the deep CNN heavily relies on the CKC algorithm and the quality of its inputs (i.e. all the MUs identified by the deep CNN are completely derived from the decomposition results of CKC algorithm). This means that it is not possible to identify new MUs (e.g. recruited during the contraction) once the deep CNN is trained. In addition, it is noteworthy that the deep CNN approach has a lower time resolution compared to other methods (e.g. 102 Hz with a step size of 20 data points). Therefore, instantaneous discharge times of MUs cannot be accurately estimated with the current method, which limit the estimation of instantaneous discharge rates or inter-spike intervals.

It is important to note that the computational time required to decompose HD-EMG into MU spike trains remains significantly higher than methods that process interference EMG in real time. For example, Durandau *et al* [6] processed bipolar EMG data to predict the force of multiple muscles in 0.31 ms. However, bipolar EMG provides a crude estimate of the neural drive [3]. The decomposition of the HD-EMG into MU spike trains, as done here, confers some advantages. First, this direct information about neural coding enables to reach a nearly perfect classification to predict human movement with shorter segments of data than for the interference EMG [4, 40, 41]. It reveals the potential interest of these data coupled with linear or non-linear decoders to predict limb trajectories and translate the neural information into prosthetic commands [42, 43]. Second, the neural drive can be used to accurately predict joint moment in multiple degrees-of-freedom in combination with information of the muscle mechanical properties from a musculoskeletal model [4, 44, 45]. This approach is advantageous as it delivers commands based on subject-specific neural drive and biomechanical properties, reflecting the individual movement patterns [45].

Although the proposed deep CNN approach already presents a good estimation of neural drive with a decreased latency between HD-EMG signals and predicted discharge times, several improvements are essential to achieve higher performance. The first opportunity for improvement relates to the structure of the deep CNN. Indeed, the performance of the deep CNN is affected by hyperparameters such as the number of convolutional layers, the number of kernel filters, and the size of the kernels. Hyperparameter optimization can be manually performed, but automatic optimization using random search will be more efficient to benchmark the optimal number of layers and nodes [46]. Thus, we could potentially maximize the performance (sensitivity and precision) of the deep CNN [26]. The processing time

is also critical for real-time applications. This could be achieved by CNN pruning; namely, reducing the size and complexity of the CNN structure through compression, which leads to reduced computational cost while maintaining performance [47]. Another interesting path worth exploring is to use specialized hardware components (i.e. neuromorphic processors) that will improve the efficiency of the deep CNN [48].

In a practical perspective, many efforts are needed to investigate and improve the robustness and reliability of the MU identification with deep CNN during varied scenarios (e.g. dynamic contractions, muscle fatigue). These conditions affect the shape of MUAPs and can decrease the performance of the deep CNN. On one hand, muscle fatigue decreases action potential amplitude and conduction velocity due to a lower sarcolemma excitability [49, 50]. On the other hand, dynamic contractions alter geometries of the muscle and the surrounding tissue through which the action potentials propagate, affecting MUAP shapes [51]. To this end, Glaser *et al* [38] have recently extended the CKC algorithm to decompose interference EMG into MUAPs, with continuous updates of the separation matrix to accommodate changes in MUAP shapes. Interestingly, Chen *et al* [20] proposed a similar approach in real time. These approaches are complementary to our proposed deep CNN, as a limitation of the neural network is that it can only track MUs previously identified by the CKC algorithm and used for training. Our future work must investigate how the deep CNN could remain efficient during fatiguing tasks and dynamic contractions with dynamic regularization.

6. Conclusion

We propose a new deep CNN to identify MU spike trains and estimate the neural drive to muscles with minimal latency. The main advantage of this method is the identification of multiple MUs from short windows of HD-EMG signals. This led to a latency of 80 ms between HD-EMG signals and MU prediction with a window size of 120 data points. Implementation of the deep CNN in a human-machine interface should be pursued, especially combining both MU identification and musculoskeletal modelling to predict movement and achieve efficient assistive device commands [5]. Additional experiments will determine whether this interface with assistive devices can improve human movement performance during daily tasks.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work was supported by Shirley Ryan AbilityLab and National Science Foundation / National Robotics Initiative (Grant No: 2024488).

ORCID iDs

Yue Wen  <https://orcid.org/0000-0001-5297-6230>

Simon Avrillon  <https://orcid.org/0000-0002-2226-3528>

Julio C Hernandez-Pavon  <https://orcid.org/0000-0001-6130-169X>

Sangjoon J Kim  <https://orcid.org/0000-0002-3245-2752>

François Hug  <https://orcid.org/0000-0002-6432-558X>

José L Pons  <https://orcid.org/0000-0003-0265-0181>

References

- [1] Farina D, Negro F, Muceli S and Enoka R M 2016 Principles of motor unit physiology evolve with advances in technology *Physiology* **31** 83–94
- [2] Thompson C K, Negro F, Johnson M D, Holmes M R, McPherson L M, Powers R K, Farina D and Heckman C J 2018 Robust and accurate decoding of motoneuron behavior and prediction of the resulting force output *J. Physiol.* **596** 2643–59
- [3] Enoka R M and Duchateau J 2015 Inappropriate interpretation of surface EMG signals and muscle fiber characteristics impedes understanding of the control of neuromuscular function *J. Appl. Physiol.* **119** 1516–18
- [4] Farina D, Vujaklija I, Sartori M, Kapelner T, Negro F, Jiang N, Bergmeister K, Andalib A, Principe J and Aszmann O C 2017 Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation *Nat. Biomed. Eng.* **1** 0025
- [5] Nizamis K, Stienen A H A, Kamper D G, Keller T, Plettenburg D H, Rouse E J, Farina D, Koopman B F J M and Sartori M 2019 Transferable expertise from bionic arms to robotic exoskeletons: perspectives for stroke and duchenne muscular dystrophy *IEEE Trans. Med. Robotics Bionics* **1** 88–96
- [6] Durandau G, Farina D and Sartori M 2018 Robust real-time musculoskeletal modeling driven by electromyograms *IEEE Trans. Biomed. Eng.* **65** 556–64
- [7] Durandau G, Farina D, Asin-Prieto G, Dimbwadyo-Terrer I, Lerma-Lara S, Pons J L, Moreno J C and Sartori M 2019 Voluntary control of wearable robotic exoskeletons by patients with paresis via neuromechanical modeling *J. Neuroeng. Rehabil.* **16** 91
- [8] Del Vecchio A, Ubeda A, Sartori M, Azorin J M, Felici F and Farina D 2018 Central nervous system modulates the neuromechanical delay in a broad range for the control of muscle force *J. Appl. Physiol.* **125** 1404–10
- [9] Farina D, Merletti R and Enoka R M 2014 The extraction of neural strategies from the surface EMG: an update *J. Appl. Physiol.* **117** 1215–30
- [10] Dideriksen J L, Enoka R M and Farina D 2011 Neuromuscular adjustments that constrain submaximal EMG amplitude at task failure of sustained isometric contractions *J. Appl. Physiol.* **111** 485–94
- [11] Del Vecchio A, Germer C M, Elias L A, Fu Q, Fine J, Santello M and Farina D 2019 The human central nervous system transmits common synaptic inputs to distinct motor neuron

- pools during non-synergistic digit actions *J. Physiol.* **597** 5935–48
- [12] McGill K C, Lateva Z C and Marateb H R 2005 EMGLAB: an interactive EMG decomposition program *J. Neurosci. Methods* **149** 121–33
- [13] Vieira T M, Botter A, Muceli S and Farina D 2017 Specificity of surface EMG recordings for gastrocnemius during upright standing *Sci. Rep.* **7** 13300
- [14] Muceli S, Poppendieck W, Negro F, Yoshida K, Hoffmann K P, Butler J E, Gandevia S C and Farina D 2015 Accurate and representative decoding of the neural drive to muscles in humans with multi-channel intramuscular thin-film electrodes *J. Physiol.* **593** 3789–804
- [15] Holobar A and Zazula D 2007 Multichannel blind source separation using convolution kernel compensation *IEEE Trans. Signal Process.* **55** 4487–96
- [16] Negro F, Muceli S, Castronovo A M, Holobar A and Farina D 2016 Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation *J. Neural Eng.* **13** 026027
- [17] Chen M and Zhou P 2016 A Novel Framework Based on FastICA for high density surface EMG decomposition *IEEE Trans. Neural Syst. Rehabil. Eng.* **24** 117–27
- [18] Chen M, Zhang X and Zhou P 2019 Automatic multichannel intramuscular electromyogram decomposition: progressive fastICA peel-off and performance validation *IEEE Trans. Neural Syst. Rehabil. Eng.* **27** 76–84
- [19] Clarke A K et al 2020 Deep learning for robust decomposition of high-density surface EMG signals *IEEE Trans. Biomed. Eng.* **68** 526–34
- [20] Chen C, Ma S, Sheng X, Farina D and Zhu X 2020 Adaptive real-time identification of motor unit discharges from non-stationary high-density surface electromyographic signals *IEEE Trans. Biomed. Eng.* **67** 3501–09
- [21] Zheng Y and Hu X 2019 Real-time isometric finger extension force estimation based on motor unit discharge information *J. Neural Eng.* **16** 066006
- [22] Glaser V, Holobar A and Zazula D 2013 Real-time motor unit identification from high-density surface EMG *IEEE Trans. Neural Syst. Rehabil. Eng.* **21** 949–58
- [23] Zhang J, Fiers P, Witte K A, Jackson R W, Poggensee K L, Atkeson C G and Collins S H 2017 Human-in-the-loop optimization of exoskeleton assistance during walking *Science* **356** 1280–4
- [24] Song-Mi L, Sang-Min Y and Heeryon C 2017 Human activity recognition from accelerometer data using convolutional neural network 2017 *IEEE Int. Conf. on Big Data and Smart Computing (Bigcomp)* (13–16 February 2017) (<https://doi.org/10.1109/BIGCOMP.2017.7881728>)
- [25] Abadi M et al (eds) 2016 Tensorflow: a system for large-scale machine learning *12th {USENIX} Symp. on Operating Systems Design and Implementation ({OSDI} 16)* (<https://doi.org/10.5555/3026877.3026899>)
- [26] Andonie R and Florea A 2020 Weighted random search for CNN hyperparameter optimization *Int. J. Comput. Commun. Control* **15** 3868
- [27] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT press)
- [28] Carriou V, Boudaoud S, Laforet J and Ayachi F S 2016 Fast generation model of high density surface EMG signals in a cylindrical conductor volume *Comput. Biol. Med.* **74** 54–68
- [29] Farina D, Mesin L, Martina S and Merletti R 2004 A surface EMG generation model with multilayer cylindrical description of the volume conductor *IEEE Trans. Biomed. Eng.* **51** 415–26
- [30] Konstantin A, Yu T, Le Carpentier E, Aoustin Y and Farina D 2020 Simulation of motor unit action potential recordings from intramuscular multichannel scanning electrodes *IEEE Trans. Biomed. Eng.* **67** 2005–14
- [31] Henneman E 1957 Relation between size of neurons and their susceptibility to discharge *Science* **126** 1345–7
- [32] Fuglevand A J, Winter D A and Patla A E 1993 Models of recruitment and rate coding organization in motor-unit pools *J. Neurophysiol.* **70** 2470–88
- [33] Hug F, Del Vecchio A, Avrillon S, Farina D and Tucker K 2021 Muscles from the same muscle group do not necessarily share common drive: evidence from the human triceps surae *J. Appl. Physiol.* **130** 342–54
- [34] Holobar A, Minetto M A and Farina D 2014 Accurate identification of motor unit discharge patterns from high-density surface EMG and validation with a novel signal-based performance metric *J. Neural Eng.* **11** 016008
- [35] Marateb H R, McGill K C, Holobar A, Lateva Z C, Mansourian M and Merletti R 2011 Accuracy assessment of CKC high-density surface EMG decomposition in biceps femoris muscle *J. Neural Eng.* **8** 066002
- [36] Vecchio A D and Farina D 2019 Interfacing the neural output of the spinal cord: robust and reliable longitudinal identification of motor neurons in humans *J. Neural Eng.* **17** 016003
- [37] Del Vecchio A, Holobar A, Falla D, Felici F, Enoka R M and Farina D 2020 Tutorial: analysis of motor unit discharge characteristics from high-density surface EMG signals *J. Electromyogr. Kinesiol.* **53** 102426
- [38] Glaser V and Holobar A 2019 Motor unit identification from high-density surface electromyograms in repeated dynamic muscle contractions *IEEE Trans. Neural Syst. Rehabil. Eng.* **27** 66–75
- [39] De Luca C J, LeFever R S, McCue M P and Xenakis A P 1982 Behaviour of human motor units in different muscles during linearly varying contractions *J. Physiol.* **329** 113–28
- [40] Chen C, Yu Y, Ma S H, Sheng X J, Lin C, Farina D and Zhu X 2020 Hand gesture recognition based on motor unit spike trains decoded from high-density electromyography *Biomed. Signal Process. Control* **55** 101637
- [41] Kapelner T, Sartori M, Negro F and Farina D 2020 Neuro-musculoskeletal mapping for man-machine interfacing *Sci. Rep.* **10** 5834
- [42] 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
- [43] Kapelner T, Vujaklija I, Jiang N, Negro F, Aszmann O C, Principe J and Farina D 2019 Predicting wrist kinematics from motor unit discharge timings for the control of active prostheses *J. Neuroeng. Rehabil.* **16** 47
- [44] Sartori M, Llyod D G and Farina D 2016 Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies *IEEE Trans. Biomed. Eng.* **63** 879–93
- [45] Sartori M, Yavuz U S and Farina D 2017 *In vivo* neuromechanics: decoding causal motor neuron behavior with resulting musculoskeletal function *Sci. Rep.* **7** 13465
- [46] Bergstra J and Bengio Y 2012 Random search for hyper-parameter optimization *J. Mach. Learn. Res.* **13** 281–305
- [47] Molchanov P, Mallya A, Tyree S, Frosio I and Kautz J (eds) 2019 Importance estimation for neural network pruning *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (<https://doi.org/10.1109/CVPR.2019.011152>)
- [48] Thakur C S et al 2018 Large-scale neuromorphic spiking array processors: a quest to mimic the brain *Front. Neurosci.* **12** 891
- [49] Enoka R M, Trayanova N, Laouris Y, Bevan L, Reinking R M and Stuart D G 1992 Fatigue-related changes in motor unit action potentials of adult cats *Muscle Nerve* **15** 138–50
- [50] Dimitrova N A and Dimitrov G V 2003 Interpretation of EMG changes with fatigue: facts, pitfalls, and fallacies *J. Electromyogr. Kinesiol.* **13** 13–36
- [51] Merletti R, Rainoldi A and Farina D 2001 Surface electromyography for noninvasive characterization of muscle *Exerc Sport Sci. Rev.* **29** 20–25