Sensing the Full Dynamics of the Human Hand with a Neural Interface and Deep Learning

Raul C. Sîmpetru[1], Andreas Arkudas[2], Dominik I. Braun[1], Marius Osswald[1], Daniela Souza de Oliveira[1], Bjoern Eskofier[1], Thomas M. Kinfe[3] and Alessandro Del Vecchio[1]*

Abstract—Theories on the neural control of movement are largely based on movement-sensing devices that capture the dynamics of predefined anatomical landmarks. Neuromuscular interfaces, such as surface electromyography (sEMG), can in theory surpass the limitations imposed by motion-based technologies by sensing the motor commands transmitted by the final pathway of movement, the motor units. The recording of motor unit activity may allow the prediction of the kinetics and kinematics continuously in time and space, without being limited to several biological and physical boundaries that digital cameras or inertial sensors suffer. However, current sEMG decoding algorithms can only predict few degrees of freedom (<3). By combining markerless machine vision and high-density sEMG electrodes, we aimed to test the hypothesis that a physiologically inspired deep neural network can reconstruct the movement of the human hand as precise as digital cameras and with the additional benefit of predicting the underlying forces (i.e., grasping a cup of coffee). We demonstrate that our deep learning model can continuously predict all degrees of freedom of the hand with negligible errors during natural motion tasks from 320 sEMG sensors placed only on the extrinsic hand muscles. Our deep learning model was able to display the 3D hand kinematics and the full force range of the hand digits during isometric contractions. The current results demonstrate that deep learning applied to EMG signals gives access to an unprecedented representation of the final neural code of movement.

Index Terms—Pose Estimation, Force Estimation, Neural Control of Movement, Electromyography

1 Introduction

Capturing the dynamics of moving body parts with a non-invasive neural interface sounds like science fiction, but what if it was real?

For kinematics acquisition, all notable methods are camera-based as they are intuitive and provide optimal results [1]–[4]. These camera systems are used in a variety of areas such as sport, rehabilitation, entertainment or robotics. Even theories of movement control [2], [5]–[7] use them to understand human movement, treating the movement itself as the only available information on which to build hypotheses. This overlooks the fact that the movement is only a reaction to an intentional input signal that should be studied instead - the neural information encoding the movement.

While these camera systems achieve optimal results, they require a pre-set recording volume and restrict the subject's movements so that they are within the camera's field of view. In addition, when used with human subjects, special care must be taken to protect privacy by, for example, obscuring the face. The main limitation of such systems is that they can only record movements after detecting changes in body position. Therefore, when an animal or human interacts with an object or assumes a static posture, there is no information about the forces exerted by the biological system.

Neural interfaces, such as surface electromyography (sEMG), can access an estimate of the neural drive to the muscle, which is highly correlated with the relative isometric muscle force [8]–[10]. Unlike cameras, the sEMG does not require translation from visual data to kinematics and kinetics data. Additionally, the sEMG does not record identifying information (face for cameras) and is therefore also less intrusive.

The sEMG literature is mainly concerned with the movement of body parts of humans and other animals, of which the hand represents the most complex system with many degrees of freedom and highly dexterous movements. Although the estimation of joint forces using the sEMG has been successfully addressed [11]–[14], the prediction of hand kinematics has not yet been answered satisfactorily. The neural interface has been hampered by a number of factors of varying severity and prevalence, theoretical shortcomings being the most important. Much attention is paid to discrete methods such as gesture recognition [15], [16] and discrete joint angle classification [17], [18], both of which are by definition incapable of providing continuous output, leading to limitations in motion output. The solution to this problem is to continuously predict either the joints as 3D points in Euclidean space [19] or the angles between the joints [20]–[28]. Although continuous prediction has no theoretical limitations, it is fraught with numerous technical problems. These problems represent the other limiting factors and take the form of high errors ($\geq 10°$ for joint angles and ≥ 15 mm for 3D points) and a small number of degrees of freedom (DoF) (< 14 - 14 is the number of flexion-only DoFs of the hand) [19]–[28].

In this study, we demonstrate that these limitations are not an inherent flaw of the neural interface but were caused by inadequate technological approaches that have now been

[1] *Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91052, Germany*
[2] *Department of Plastic and Handsurgery, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91054, Germany*
[3] *Division of Functional Neurosurgery and Stereotaxy, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91054, Germany*
* *Correspondence to: alessandro.del.vecchio@fau.de*

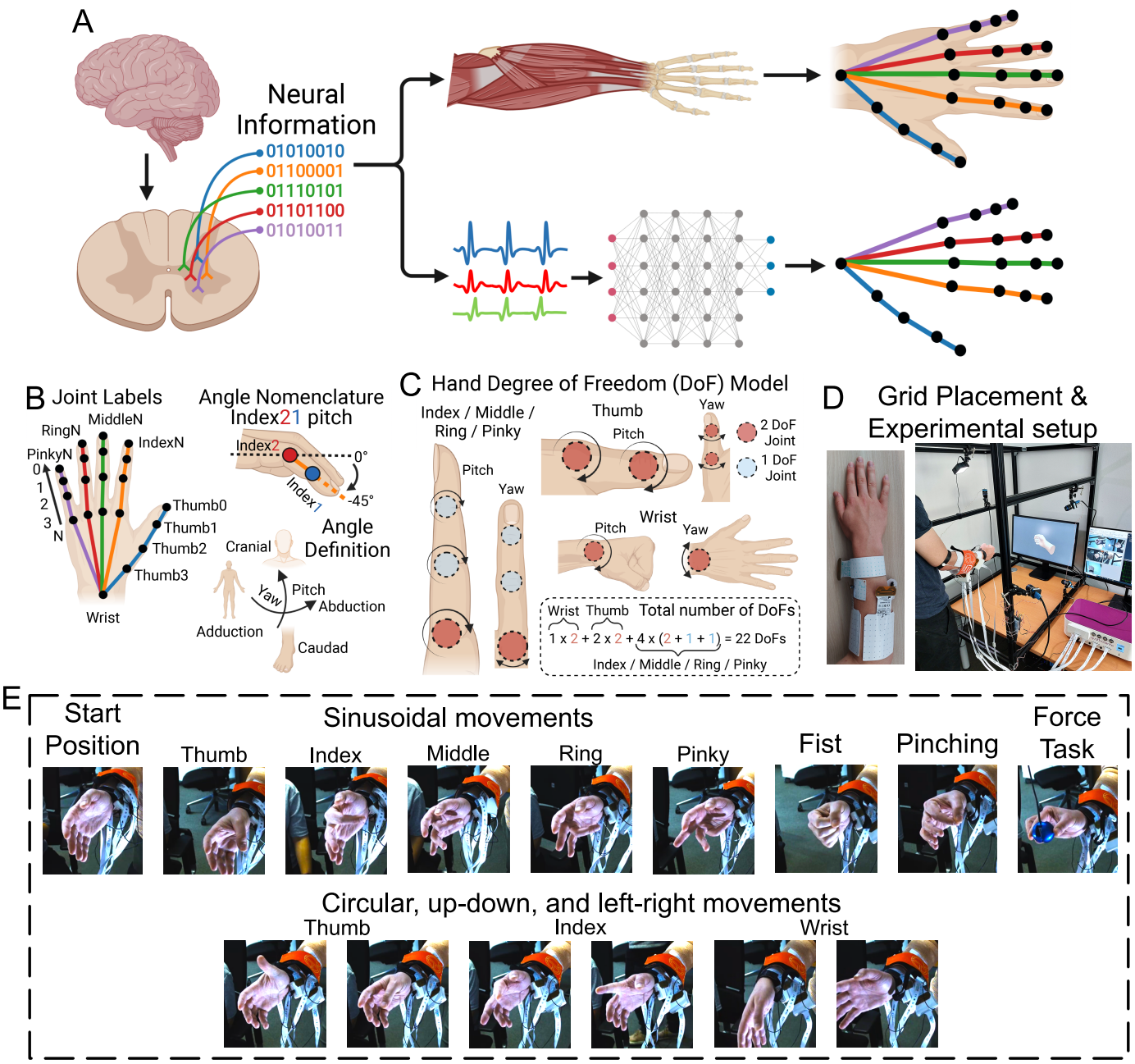

Fig. 1. **A.** High-level overview of the study showing the neurophysiological pathway used for EMG generation together with the framework used for the training of the machine learning model. **B.** Visual explanation of the joint labels and angle definitions. **C.** Hand degree of freedom (DoF) model. **D.** Pictures of the grid placement and the experimental setup. **E.** Representation of all executed movements together with the grid placement and the experimental setup.

surpassed. We recruited 13 subjects (11 male and 2 female), from whom we recorded the sEMG, kinematics, and kinetics synchronously. The sEMG is then used as input to our novel neural network architecture, which was developed specifically for this signal. Our network was used in this study to output not only the 3D points of the hand, but also the joint angles and the isometric joint force. Figure 1A and Video 1 show an overview of our study and the prediction accuracy of our deep learning method.

All 3 output forms achieved exceptionally low errors compared to previous studies. We were able to output 22 degrees of freedom of the hand during slow and fast movements of the hand digits. The model was able to learn with very high accuracy not only combinations of finger movements (grasping, two and three finger pinches) but also individual finger flexion and extension at low and fast speeds, maintaining high continuous and proportional prediction accuracy. The mean Euclidean distance between the predicted and expected 3D joint coordinates was 3.7 $\pm$ 2.14 mm, the mean Pearson's correlation coefficient between predicted and expected joint angles was 0.92 $\pm$ 0.03 with a mean absolute error of 3.14 $\pm$ 1.75 degrees. For force prediction, we achieved a mean absolute error of 12.26 $\pm$ 11.72 mN.

We therefore prove that the sEMG signal is comparable to camera systems in terms of kinematic accuracy (Fig. 2, Video 1), but also provides additional information such as the forces of the joints acting on an object. This work has important implications not only for human-machine interfaces, but also for animal behaviour studies.

2 Results

We acquired sEMG data from 320 electrodes placed on the extrinsic hand muscles together with 5 cameras simultaneously recording different hand movements (Video 1, Fig. 1) at 120 fps. The videos were processed by DeepLabCut [2] and Anipose [29] to create 3D coordinates for each hand joint (Fig. 1B and C). From the 3D coordinates, we further calculated angles as a different means of encoding the kinematics. For the kinetics experiments, we recorded the sEMG signals synchronously with the force measured by a dynamometer (COR2, OT Bioelettronica, Turin, Italy).

The train/test splitting of the data is done differently for kinematics and kinetics, but the way we use it after splitting remains the same. For each training epoch, the bins are reshuffled using a random number generator.

The kinematics contains two sets of tests. The first set, called "test" (Videos 1 and 2), consists of the middle 20% of each motion recording (Fig 1E) without the random one. In total, this set is 2 min 23 s long. The remaining 80% are used as a training set. The second set, called "random movement" (Videos 3 and 4), consists of the random movement task. We use this set to test whether an untrained combination of motions, as it occurs in real life, can be reliably predicted by our system. The training set consists of all other motion tasks.

The kinetics tasks (2- and 3-finger pinching) consist of 4 force ramps. The test set consists of the last ramp of both tasks (2 ramps in total), while the training set is formed of the remaining ramps (6 ramps in total).

A machine learning model is trained with the sEMG as the only input. The ground truth comes from the cameras or the force sensor. For each test subject, a model was trained on a machine with 8x RTX 3080, 2x Intel Xeon Gold 6226R @2.9 GHz and 384 GB RAM for 50 epochs, which meant about 4.5 hours of training each.

The preferred error metric for the output of the 3D points is the mean Euclidean distance (MED). We have noticed that although the model is outputting the correct movement trend, some small yet not insignificant jitter remains. To counteract this we used Procrustes analysis (Algorithm 1). Briefly, this method translates, scales, and rotates the output such that it may best fit the ground truth, while maintaining the same shape of the output and reducing the jitter. A graphical overview of both the Procrustes analysis and how we derive our result plots for the 3D outputs is shown in Fig. 2A. Fig. 2B and C show violin plots of the Euclidean distance between the predicted and expected 3D joint coordinates over the test set. The figures show the results before and after the Procrustes analysis. As mentioned, we also tested our model on a random motion task to evaluate if the system works in an environment more similar to a real life setting. Fig. 2D and E show the results of this analysis both before and after the Procrustes analysis. We were able to achieve errors as small as 3.7 ± 2.14 mm across all the 22 functional degrees of freedom of the hand. During the random tasks, the error increased by 9.78 ± 5.04 mm (Fig. 2E), however, this is to be expected, as we did not train the neural network on that signal. Indeed, the training data and random movement set differed significantly (Video 1 vs Video 3).

We then also aimed at comparing the joint angle kinematics, which differs from the 3D points analysis. For the angle outputs, we chose both the Pearson correlation coefficient (PCC) (Fig. 3B and D) and the mean absolute error (MAE) (Fig. 3C and E) as the two-error metrics for a robust overview of the joint angle predictions. Fig. 3A shows the overview of the analysis and prediction accuracy of the angle kinematics. The force output is evaluated (Fig. 4A) exactly the same as the angles are - using PCC and MAE (Fig. 4B and C).

We then looked at how different EMG input types affect the performance of our system for the 3D points output (Fig. 5). The input types are as follows:

- 5 Hz - 5 Hz low-pass filtered EMG
- |5 Hz| - 5 Hz low-pass filtered rectified EMG
- 20 Hz - 20 Hz low-pass filtered EMG
- |20 Hz| - 20 Hz low-pass filtered rectified EMG
- Raw - unfiltered EMG
- |Raw| - unfiltered rectified EMG
- Raw + 20 Hz - unfiltered and 20 Hz low-passed filtered version of the same EMG

We have used a phase-free 4th order Butterworth filter to low-pass the EMG binwise backwards and forwards. The results shown are for the first 4 subjects only, as training models for the other subjects would unnecessarily increase the usage of our computing resources.

For visual inspection of our system, we created videos (example frame shown in Fig. 6) made from the original camera videos overlaid with the joint labels and 2 skeleton hands. The predicted hand is the 3D points output of our model, while the expected hand is determined by the coordinates obtained from the cameras. All of our videos are accessible at this link. Also, we display the top 10 most active kernels and their activation maps across the EMG grids in Video 5 titled "Visualisation of the model kernel activities". We provided the videos for subject 1 and 7 for both the test and random movement sets.

3 Discussion

We have collected data from 320 sEMG channels together with the kinematics and kinetics of 13 subjects executing different hand movement tasks (Fig. 1E). Our deep learning framework was able to estimate hand kinematics as accurate as the digital cameras (Fig 2C and E, Videos 1-4) and near-perfect force prediction (Fig. 4B and C).

Fig. 2B-E show the results for the 3D points output on the test and random motion sets respectively. Videos 1 and 2 show the results for the test set and Videos 3 and 4 for the random movement set for subjects 1 and 7. Our model performs remarkably well on the test set, but we believe the noteworthy results are seen on the random motion task, with errors smaller than one centimeter. The results for the predicted angles are shown in Fig. 3B-E. The test set shows

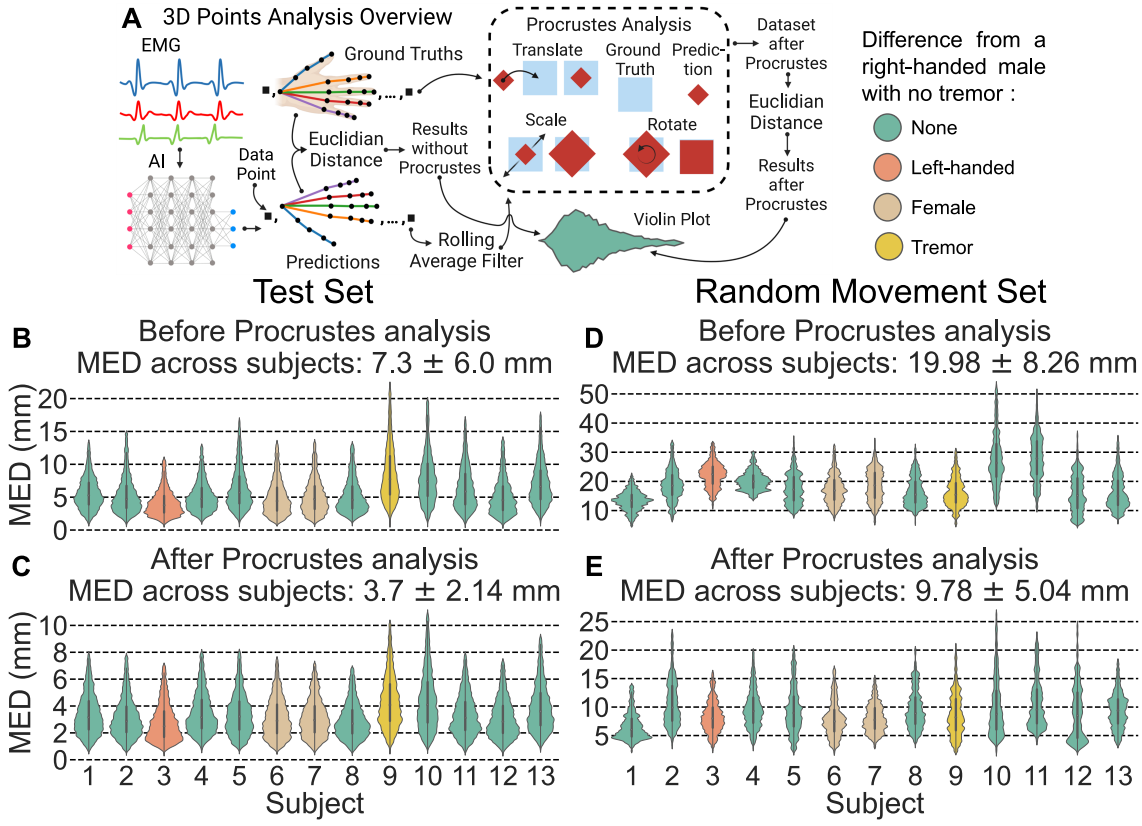

Fig. 2. **A.** Graphical explanation of both the Procrustes analysis and the 3D points result plots. **B. & C.** Violin plots showing Euclidean distance between the expected and predicted 3D joint coordinates both before and after the Procrustes analysis for the test set. **D. & E.** Same as B and C but for the random movement set.

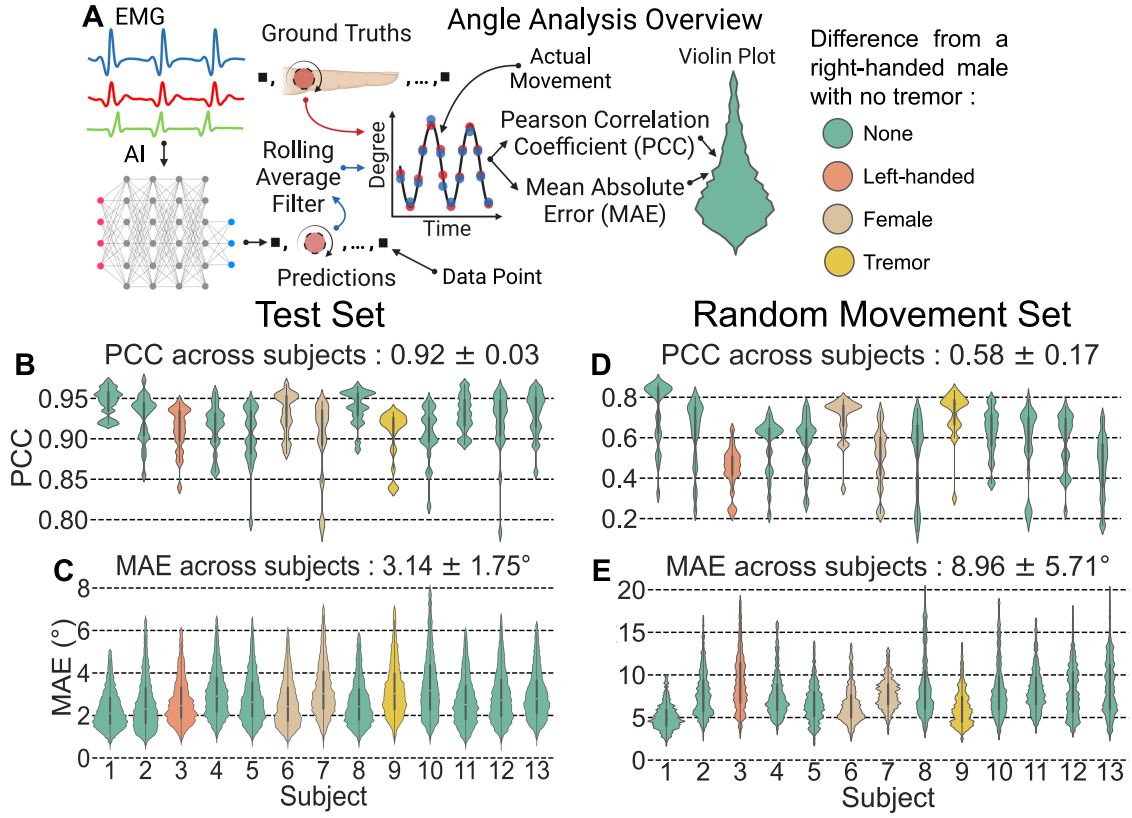

Fig. 3. **A.** Graphical explanation of the angles result plots. **B. & C.** Violin plots showing the Pearsons correlation coefficient (PCC) and mean absolute error (MAE) between the expected and predicted joint angles for the test set. **D. & E.** Same as B and C but for the random movement set.

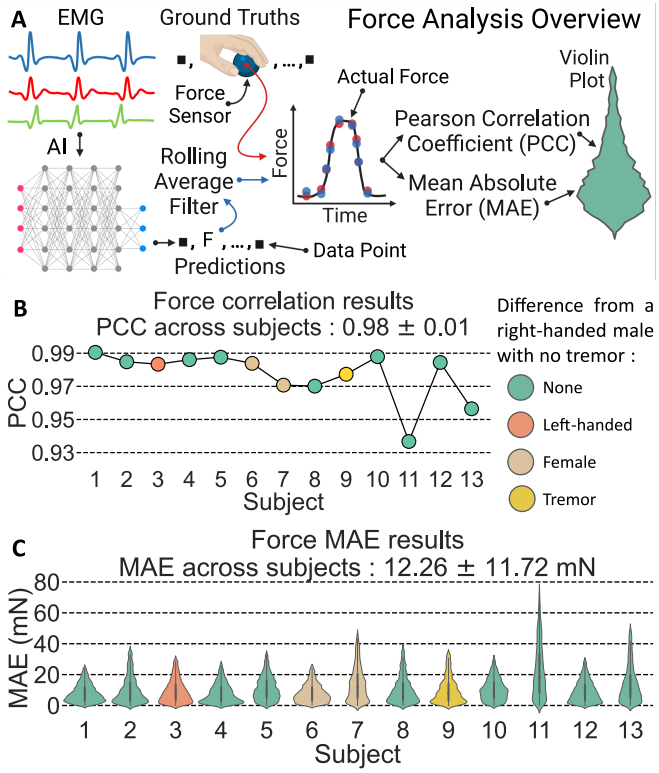

Fig. 4. **A.** Graphical explanation of the force result plots. **B.** Point plot showing the correlation between the expected and predicted force value. **C.** Violin plot showing the mean absolute error between the expected and predicted force value.

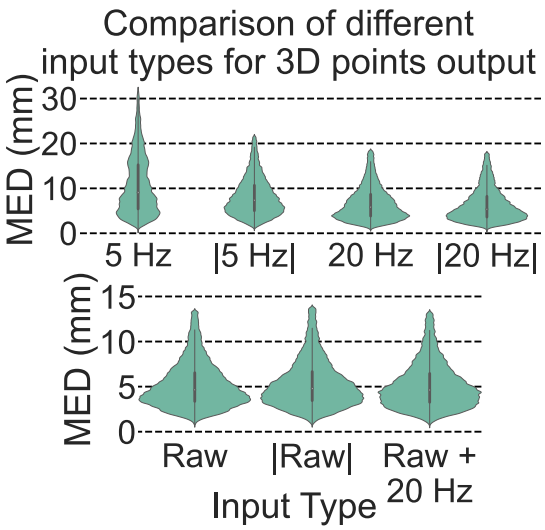

Fig. 5. Comparison between different input types. The input types are described as : "x Hz" is the x Hz low-passed filtered EMG, "Raw" is the unfiltered EMG, "| |" means that the EMG has been rectified, and "Raw + 20 Hz" is the input type we chose, which uses both the unfiltered and the 20-Hz low-passed variant of the same EMG.

perfect results, but as with the 3D points, we are mainly interested in the set with random movements, as this best represents a real life scenario. From the violin plots it can be seen that most angles are moderately ($0.6 \leq x < 0.8$) to strongly (x ≥ 0.8) correlated, but some are poorly correlated. From Fig. A2 we can see that the yaw angles (left-right) are the most difficult to predict. In view of these results, we recommend training the architecture to output 3D points and, if necessary, to calculate the angles in an additional step instead of having them outputted directly from the model.

It is well known that the EMG is highly correlated with the exerted joint force. In Fig. 4 we demonstrate that our model is capable of near perfect (0.98 ± 0.01 correlation with 12.26 ± 11.72 mN of error) force prediction during 2- and 3-finger pinching. This result proves one of the main benefits of neural interfaces over any camera or inertial sensor based system - force prediction.

Our final result (Fig. 5) shows a comparison between different input types for the output of 3D points. We conducted this test because we hypothesised that the common method in the literature of low-pass filtering EMG (rectified or not) below 5 or 20 Hz [15], [22], [28], [30], [31] is an incorrect solution for a learning system like our deep learning model and for an accurate translation of EMG activity into kinematics. The result shows that the input type used in this study (raw and 20 Hz low-passed version of the same EMG bin) outperforms all other types. This can be attributed to 2 factors. First, rectification removes important information that is picked up by the monopolar electrodes. Furthermore, prior rectification of the signal is not necessary for a learning model, as the mathematical operation that such a system should learn is trivial. We supplement the unfiltered (raw) EMG with the 20 Hz low-pass version because the muscle acts like a 5-6 Hz low-pass filter [32]. So by filtering the EMG for our deep learning model, we give some information about the effective neural drive that produces slow movements. From the result, we deduce that the combined input is most effective because the network likely learned the movement space (low-pass filtering) and the motor unit discharge times (Video 5). It is likely that the neural network is learning the motor unit action potentials, considering that the raw monopolar derivation shows the highest accuracy among all other inputs. This is due to the fact that motor unit action potential frequencies occur in a relatively large range (40-300 Hz) [33].

This proves that the sEMG outperforms camera-based solutions, as it not only rivals cameras in predicting kinematics, but is also able to estimate something cameras and inertial sensors cannot - the exerted force during isometric contractions. Our framework opens several perspectives at the consumer level (e.g., virtual reality), medical applications (robotics), and hazardous environments (teleoperation of robots) and for basic neurophysiological research. For example, the presented results demonstrate that the extrinsic hand muscles contain most of the information for predicting the 22 functional degrees of freedom of the hand, which likely contains the presence of strong common input between extrinsic and intrinsic hand muscles [34], [35]. Our results find relevance not only in humans but also in animal research. Although non-invasiveness is very important for humans, EMG sensors could be implanted

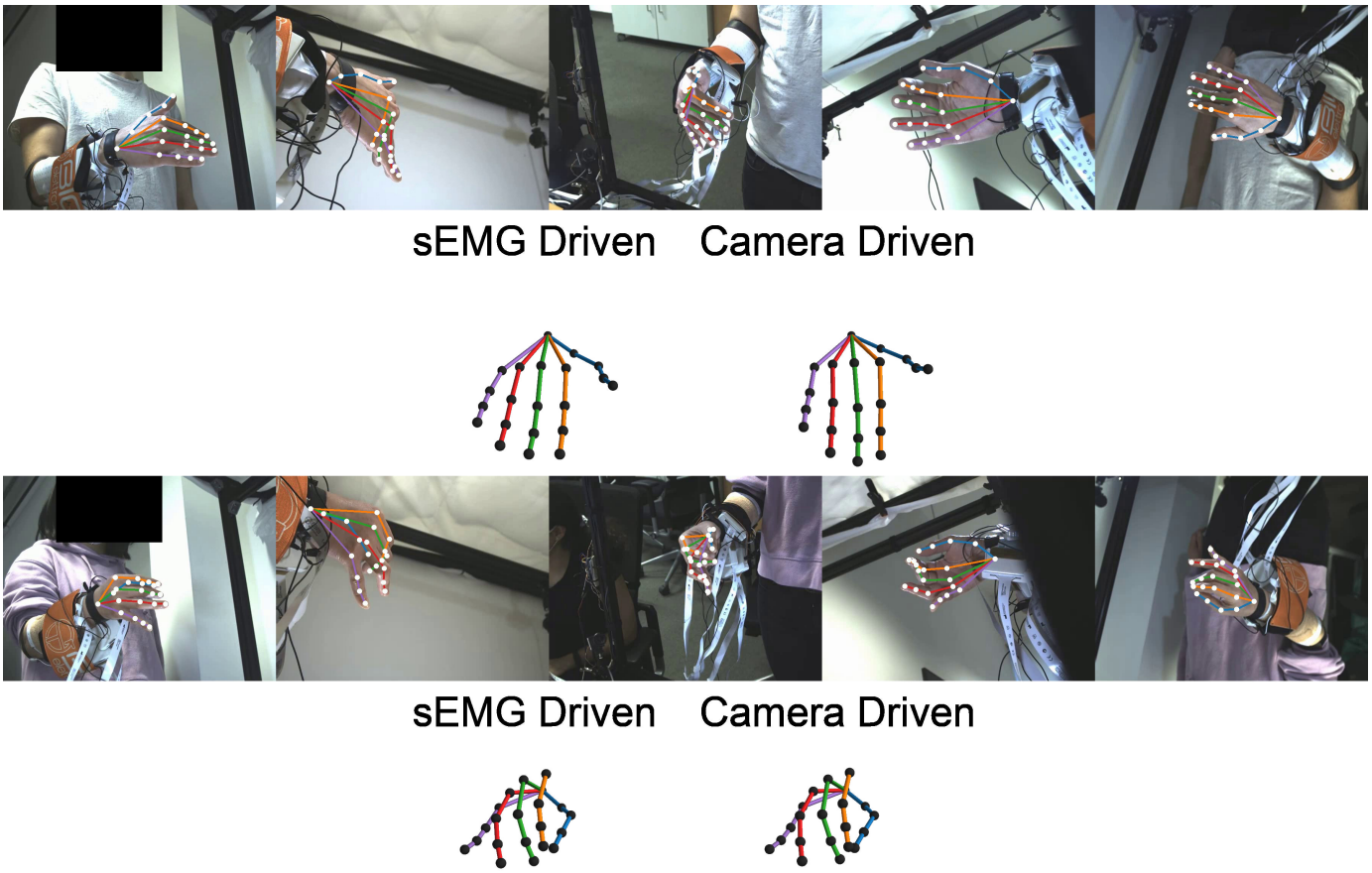

Fig. 6. Example images from the videos used for visual inspection showing subject 1 flexing the wrist (top) and subject 7 performing a 3-finger pinch (bottom). The sEMG-driven hands use the 3D point output of our network, while the camera-driven ones use the 3D coordinates of Anipose [29]. All of our videos are accessible at this link.

in multiple muscles in animals, as we have already done in macaque monkeys [36]. This would provide a robust, continuous representation of animal dynamics without the limitations imposed by camera systems or wearable devices (accelerometers).

4 Conclusion

In this paper, we present a new interface capable of reliably extracting hand kinematics and kinetics from high-density sEMG data. The recording setup consists of 320 non-invasive electrodes spread over 5 64-channel grids (3 around the circumference of the forearm and 2 around the wrists) recorded synchronously with the hand kinematics by a high precision markerless camera setup and deep learning methods. Our results surpass previous decodings of hand kinematics from any neural or muscular interface, both in number of DoFs and accuracy of the prediction. We show very high prediction reliability when training the system for 3D points output on a random movement task and close to perfect prediction of force when doing a 2- or 3-finger pinch. We contend that our model is implementable as a hand interface on already existing hardware and could be used for assistive device control and more immersive virtual reality controllers.

5 Methods

Before we get into the complex solutions required to realise our novel model, we would like to give the reader an extended high level overview to refer to if needed (Fig. A1).

Similar to our prior conference study [37, accepted for publication], we collected sEMG data from 320 electrodes and recorded simultaneously the hand movements using 5 cameras. We recorded 13 subjects out of which 11 were male and 2 female. Both female (6 and 7) and 9 males (subjects 1, 2, 4, 5, 8, 10, 11, 12, and 13) are right-handed. From the remaining 2 males, one (subject 3) is left-handed and the other (subject 9) is right-handed with essential tremor.

The subjects were asked to perform 20 different movements (Fig. 1E gives a visual representation of these movements) for 40 s each:

- sinusoidal movement (flexion extension) of each digit at 0.5 and 1.5 Hz. These two frequencies have been experimentally chosen to represent slow/fast movements.
- closing and opening the fist.
- pinching between index and thumb.
- pinching between index, middle finger, and thumb.
- circular (clockwise and counterclockwise), left-right, and up-down movement of the index, thumb, wrist. We recorded 2 different sets of movement consisting of different sequences of the aforementioned movements.
- random movement that consists of every task described above.

To aid and guide the subject in keeping a consistent movement frequency, a realistic virtual hand model created in Blender (Blender 3.0, Blender Foundation) was displayed on a monitor (Fig. 1C). In addition to kinematics, kinetics synchronised with the sEMG during 2- and 3-finger pinching were recorded with a dynamometer (COR2, OT Bioelettronica, Turin, Italy). Before recording the force, we determined the maximum voluntary contraction (MVC) of each subject. We then recorded 4 force ramps for each type of pinch. One such ramp consists of 3 parts: the increase (from 0 to 40 % MVC - 10 s duration), the hold (constant 40 % MVC - 3 s duration), and the decrease (from 40 % MVC to 0 - 10 s duration).

The experiments were reviewed and approved by the ethics committee of the Friedrich-Alexander University (application no. 21-150_3-B) for compliance with the Declaration of Helsinki, and the subjects signed an informed consent form.

5.1 Data acquisition

The sEMG is acquired from 3 electrode grids (8 rows x 8 columns, 10-mm interelectrode distance; OT Bioelettronica, Turin, Italy) placed around the thickest part of forearm and 2 around the wrist (13 rows x 5 columns, 8-mm interelectrode distance; OT Bioelettronica, Turin, Italy), proximal to the head of the ulna (Fig. 1C).

Prior to the electrode grids placement, the skin is shaved and cleaned with an alcoholic solution. The monopolar sEMG signals were recorded using a multichannel amplifier (EMG-Quattrocento, A/D converted to 16 bits; bandwidth 10–500 Hz; OT Bioelettronica, Turin, Italy), amplified (×150) and band-pass filtered (10–500 Hz) at source, prior to offline analysis. The signals were sampled at 2044 Hz and collected using the software OT BioLab (OT Bioelettronica).

While the sEMG signal is acquired, 5 720x540 pixel videos of the hand movement are recorded synchronously at 120 Hz by cameras (DFK-37BUX287, The Imaging Source™, Bremen, Germany) placed in different corners (upper/lower left/right) of a rectangular metal frame to capture multiple views of the same movement (Fig. 1D).

The videos were processed by a custom trained neural network using the DeepLabCut [2] framework. The network was trained on manually labeled frames of hand and digit positions and outputs the 2D location of all hand and digit joints. The 2D positions were then triangulated into 3D Kinematics using the extension library Anipose [29].

The achieved mean absolute error (MAE) between user label position and predicted position is 3 mm, indicating high reliability in tracking of the hand digits kinematics.

From the 3D coordinates we then compute the yaw and pitch angles by defining one joint as the origin of a sphere coordinate system and the next joint in series as a point on said sphere, e.g. Index2 as origin and Index1 on the sphere (Fig. 1B).

To ensure accurate synchronisation both between cameras and between cameras and sEMG recordings, a trigger signal was sent from a microcontroller (Arduino UNO, Arduino LLC) to each camera and the sEMG recording device.

The force is recorded by the dynamometer (COR2, OT Bioelettronica, Turin, Italy) at the same frequency as the sEMG (2044 Hz). To obtain the force information in newtons, we calibrated the sensor with several weights (10 g, 20 g, 50 g, 100 g, 200 g, 250 g, 500 g, 550 g, 700 g, and 750 g).

5.2 Preprocessing

After the data has been recorded, we need to preprocess it so that it can be used by our model. The preprocessing pipeline is shown in Fig. A3. The sEMG and videos are recorded while the hand moves according to a task. Since there is a small hardware delay between the sEMG and the videos when the recording starts, we need to synchronise them. The 3D points and angles were then oversampled from 120 to 2044 Hz to match the sEMG. The first 4.5 s of the recordings were removed, as this is the resting state used to check for EMG artefacts during recording. Since the termination of the sEMG recording is not automatic, the sEMG was cut to the length of the video. Then the recording was divided into bins of 192 samples in length (about 94 ms) with a shift between bins of 64 samples (about 31.3 ms). From this point on the kinematics and kinetics were treated differently from the sEMG.

5.2.1 sEMG

The sEMG bins are copied, low-pass filtered forwards and backwards with a 4th order Butterworth filter, and depth stacked with the original unfiltered bins. The combined bins are then augmented with Gaussian noise and magnitude warping [38], providing a threefold data increase.

5.2.2 Kinematics and Kinetics

We took the mean of the individual kinematics (3D points and joint angles) or kinetics (force) bins because they do not change significantly within 94 ms. This reduces the output form of our model from a matrix to only one vector per sEMG bin. The vectors are then normalised component-wise between -1 and 1 using the min-max normalisation function

$$x' = (b - a)\frac{x - \min(x)}{\max(x) - \min(x)} + a \tag{1}$$

where a is the new desired minimum, b the new desired maximum, x the original signal and x′ the normalised signal. The minimum and maximum for the sEMG is only found within the bin, as otherwise the entire signal would have to be known, which would not be possible in a future real-time setup. For kinematics and kinetics, the minimum and maximum are determined from the entire signal. This function is necessary because differently scaled inputs can cause neural networks to become unbalanced and underperforming. Since the augmented sEMG bins must be paired with ground truth bins (kinematics or kinetics), we copied the original ground truth bins corresponding to the augmented bins.

5.3 Connection between EMG and convolutions in deep learning

Before we go into the actual model architecture, we would like to set out the motivation for our choice by taking a closer look at EMG generation theory.

The smallest unit that can be activated by the brain to perform a motor task is called the motor unit (MU) [39]. The

MU consists of a α motor neuron and many muscle fibres that are innervated by its axon.

One MU can innervate several muscle fibres, which means that the specific MU action potential (MUAP) does not have to be recorded at a specific point in the muscle, but can be recorded at several points simultaneously. Moreover, because of the properties of the volume conductor, the MUAP propagates through the muscle fibers and the biological tissue layers that separate the action potential (source) and the recording electrodes. For these reasons, the MUAP shows distinct waveform signatures in the high-density sEMG.

The EMG signal represents the sum of the individual MUAPs at different times [40]. These signals are recorded by non-invasive electrodes (usually arranged in a grid pattern) placed on the skin. The different firing patterns are used by the brain to encode information about the force required for a particular action. This process is called rate coding. In addition to the frequency of firing, the brain can also choose the number of motor neurons needed to perform an action, with more neurons meaning more power and fewer the exact opposite. This process is called MU recruitment and, together with rate encoding, forms the most important variables that the brain can control muscle forces.

An ideal one neuron system can be formulated mathematically [41] as

$$n(t) \coloneqq \sum_{i=0}^{\tau} \delta(t - t_i) \cdot h(t) \tag{2}$$

which is a discrete convolution with $h(t)$ being the MUAP, $\delta(t - t_i)$ the Dirac delta impulse symbolising the distinct moments in time t_i when the motor neuron fired, and $n(t)$ the activity of the neuron over a specific discrete time interval from 0 to τ.

Expanding the system to include N neurons recorded by one electrode gives us the EMG signal equation

$$\begin{aligned} \text{EMG}(t) \coloneqq \text{noise}(t) + \sum_{j=0}^{N} n_j(t) = \\ \text{noise}(t) + \sum_{j=0}^{N} \sum_{i=0}^{\tau} \delta(t - t_{j,i}) \cdot h_j(t) \end{aligned} \tag{3}$$

where $n_j(t)$ is the jth neuron activity from equation (2), $t_{j,i}$ the ith moment in time when the jth neuron fired, and noise(t) the additive system noise.

Finally we can arrange X times Y electrodes in a grid and formulate the EMG signal as the 2D time dependant function

$$\text{EMG}(t,\, x,\, y) \coloneqq \text{noise}(t,\, x,\, y) + \sum_{j=0}^{N} n_{x,\, y,\, j}(t) \tag{4}$$

where $n_{x,\, y,\, j}(t)$ is the jth neuron activity from equation (2) recorded by the electrode at the grid position $(x,\, y)$ and noise$(t,\, x,\, y)$ the additive noise at the same grid position.

The challenge of learning patterns from an EMG signal can be mainly attributed to three facts that arise from the generation formula described in equation (4):

Fact 1. The durations of the MUAPs vary between the neurons.

Fact 2. The number of neurons recruited and their respective firing frequencies are unknown.

Fact 3. One electrode can detect multiple MUAPs that are possibly happening at the same time, thus making the recorded signal an overlap of multiple individual signals.

From Fact 3 we derive that each sample of the signal must be treated as a possible combination of multiple MUAPs. The overlaps are not stochastic but are a result of the brain's MUs activation choice. In this way, we can use the overlaps for pattern recognition and thus learn the specific combinations that contribute to different movements. Fact 1 and 2 tell us that the specific amount of time information we need to make a prediction is unknown. We decided to use 192 samples ($\approx$ 94 ms).

Using the derived knowledge, we identify a convolutional neural network as the perfect solution for information extraction. For simplicity, we will explain our choice on a 1D convolution operation for the system described in equation (3) and then extend it to the multi-electrode system.

$$O_{K,B}(s, I) \coloneqq B + \sum K \odot W_{K_{\text{len}}}(s,\, I) \tag{5}$$

$$W_{K_{\text{len}}}(s,\, I) \coloneqq \left[H\left(s - \left\lfloor \frac{K_{\text{len}}}{2} \right\rfloor\right) - H\left(s + \left\lfloor \frac{K_{\text{len}}}{2} \right\rfloor\right) \right] \odot I \tag{6}$$

Equation (5) shows the discrete 1D auto-correlation function where I is the input signal, K the learnable kernel, B the learnable bias, $W_{K_{\text{len}}}(s,\, I)$ (6) the windowed input I at a specific step s given the kernel size K_{len}, and $O_{K,B}(s, I)$ the output derived by applying the kernel and bias to the input signal at the step s. Equation (6) describes the windowing function for an 1D signal given a specific kernel length K_{len} by utilising the Heaviside step function $H(s)$ (1 if $s \geq 0$ and 0 otherwise) as a mathematical model that sets all information outside of the kernel's overlap to 0. For both equations the symbol $\odot$ denotes the Hadamard product (element-wise multiplication).

Before continuing we wish to acknowledge that equation (5) does in fact represent an auto-correlation and not a convolution, however we do this deliberately as the convolution operation is implemented as an auto-correlation in all notable machine learning frameworks because it removes the need of flipping the kernel prior to applying it.

From equation (5) we can derive that at every single step s a kernel K is applied to the input signal I. This property is called translational equivariance, and since we have derived from Fact 3 that the MUAPs can be taken at multiple points simultaneously, it is perfectly suited for our problem.

For the multi-electrode setup from equation (4) we can extend equation (5) using matrix notation in two ways. The first preserves the grid structure and allows indexing with x and y equation (7) while the other one flattens the grid to a one dimensional list of electrodes equation (8).

$$\begin{aligned} O_{K,B}(s, I) \coloneqq \begin{pmatrix} B & \cdots & B \\ \vdots & \ddots & \vdots \\ B & \cdots & B \end{pmatrix} + \sum \left[\begin{pmatrix} K & \cdots & K \\ \vdots & \ddots & \vdots \\ K & \cdots & K \end{pmatrix} \odot \right. \\ \left. \begin{pmatrix} W_{K_{\text{len}}}(s,\, I_{0,0}) & \cdots & W_{K_{\text{len}}}(s,\, I_{0,x}) \\ \vdots & \ddots & \vdots \\ W_{K_{\text{len}}}(s,\, I_{y,0}) & \cdots & W_{K_{\text{len}}}(s,\, I_{y,x}) \end{pmatrix} \right] \end{aligned} \tag{7}$$

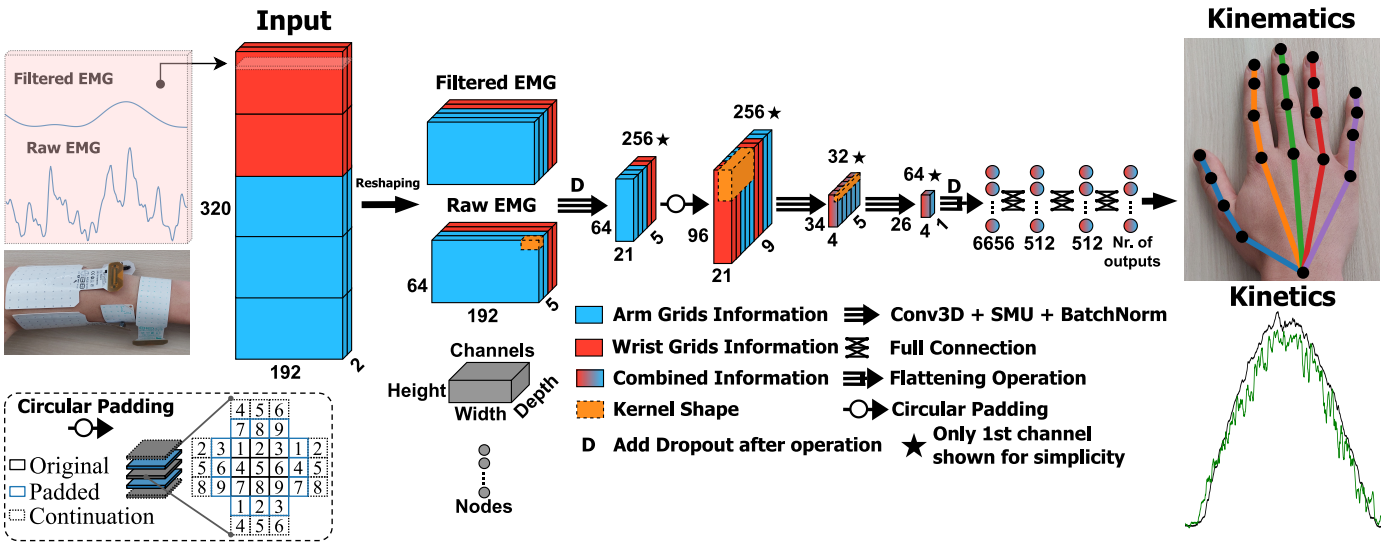

Fig. 7. Architecture overview with a visualisation of the circular padding.

$$O_{K,B}(s,I) := \begin{pmatrix} B \\ \vdots \\ B \end{pmatrix} + \sum \left[\begin{pmatrix} K \\ \vdots \\ K \end{pmatrix} \odot \begin{pmatrix} W_{K_{\text{len}}}(s,\, I_0) \\ \vdots \\ W_{K_{\text{len}}}(s,\, I_{x \cdot y}) \end{pmatrix} \right] \tag{8}$$

For the model described in this paper we have decided to use the flattened representation from equation (8) because by flattening the grid we can use the free dimension to represent multiple grids that have been recording at the same time. We could also have used the xy representation from equation (7) but a 4D operation is needed to represent multiple grids, which is very unoptimised and expensive to compute on modern hardware due to its relatively small demand.

5.4 Model

This work has been implemented in Python using PyTorch (version 1.11.0+cu113) [42] and PyTorch-Lightning (version 1.6.0) [43]. Fig. 7 shows a high level overview of the AI model architecture together with the input and output.

5.4.1 Architecture

The input is one or more sEMG matrices of shape 320 (total number of electrodes) $\times$ 192 samples ($\approx$ 94 ms). For this study we used two matrices. The first matrix contains band-passed (10 - 500 Hz) sEMG ("Raw EMG" in Fig. 7) and a second one that is a 20 Hz low-passed filtered copy of the first one ("Filtered EMG" in Fig. 7). These two matrices are concatenated in depth, giving us an input of shape 2 $\times$ 320 $\times$ 192. The input is then split by the amount of electrodes per grid (in our case 64 per grid) inside the model into a 4D tensor of shape 2 (number of inputted matrices) $\times$ 5 (number of grids) $\times$ 64 (number of electrodes) $\times$ 192 samples.

The output depends on the required format. In this study, we trained our network to output the entire hand skeleton in Euclidean space, all joint angles and the force applied to a force sensor.

When the whole hand has to be output in 3D, the output is a 60D vector (20 labels $\times$ 3 coordinates = 60 values). The wrist is stabilised and does not need to be output by the model as it has a constant position.

For the output of the angles, we have chosen to model the degrees of freedom of the hand as close as possible to the anatomical reality (Fig. 1C), not limiting ourselves to making logical mathematical simplifications when these have no effect on the hand model. We would therefore like to explicitly state that we are aware that anatomically the thumb consists of the carpometacarpal joint, the metacarpophalangeal joint and the interphalangeal joint (thumbs 3, 2 and 1 in Fig. 1B), but the overall position of the thumb can just as well be encoded by two 2D ball joints located at the position of thumb 2 and 1, as can be seen in Fig. 1C. With our hand model we get a 22D vector as output for the angle-based representation.

In the case of force prediction, only one variable needs to be output, which means we get a 1D scalar.

The model (Fig. 7) begins with a 3D convolutional layer (kernel size = 1x1x31; stride = 1x1x8; output channels = 256). We search each electrode in a window of 31 samples ($\approx$ 15.2 ms) with a shift of 8 samples ($\approx$ 4 ms) for action potentials in the sEMG.

Each 3D convolution used is followed by 3D batch normalisation and the learnable activation function SMU [44], which has been shown to perform better than most common activation functions such as ReLU, Swish or Mish [44].

Before the next layer, we added a 3D dropout layer (p = 0.25) to prevent overfitting.

The next step is a circular padding. This layer extends a specified dimension by looping around to the other side of the same dimension. Fig. 7 illustrates the circular padding more clearly. The circular padding (2 for back and front; 16 for top and bottom) is applied to the output of the first step and ensures that the synchronicity of the recording is preserved for the next 3D convolution.

The subsequent layer is again a 3D convolution (kernel size = 5x32x18; dilatation = 1x2x1; output channels = 32). Dilation is used to look at events that are further apart. The kernel looks at five grids at once, giving us the grid combinations (1 is the first and 5 the last grid) of 45123, 51234, 12345, 23451 and 34512. This layer acts as an encoder and creates a representation of the extracted action potentials with lower dimensionality.

The last 3D convolution step's (kernel size = 5x9x1; output channels = 64) task is to collect refined information and discard unimportant ones.

Afterwards, the output is flattened, has an 1D Dropout (p = 0.4) applied to it, and passed through a simple 3-layered perceptron (512, 512, number of output variables). The first two layers of the perceptron are followed by an SMU, while the last layer is followed by nothing, as this is the output.

5.4.2 Loss function

The loss function used is the mean absolute error (L1Loss in PyTorch). We chose this function because we want to linearly penalise any deviation from the ground truth.

5.4.3 Optimiser and learning rate scheduler

For this architecture, we use the AdamW optimiser described in [45] with the AMSGrad correction from [46]. The weight decay is 0.01. We used the one cycle approach described in [47] as the learning rate scheduler with the upper bound being $10^{-2.5}$, the lower bound 10^{-7} and the initial learning rate 10^{-4}.

5.5 Postprocessing

All output forms are filtered with a moving average filter of 150 ms. This is used to smoothen the output and eliminate the fast peaks in the output while preserving the overall trend.

For part of the 3D points analysis, we use the Procrustes method [48] shown in Algorithm 1.

Algorithm 1 Procrustes Analysis

Require: predicted_hand, expected_hand # matrices of shape: samples $\times$ 60 (x, y, z for 20 points)

Both hands are stabilised by the wrist $\Rightarrow$ x, y, z = 0. The wrist must be concatenated
wrist_point $\leftarrow (0, 0, 0)$
predicted_hand $\leftarrow$ concat(wrist_point, predicted_hand, axis=0)
expected_hand $\leftarrow$ concat(wrist_point, expected_hand, axis=0)

Center hands around origin
predicted_hand $\leftarrow$ predicted_hand $-$ mean(predicted_hand, axis=0)
expected_hand $\leftarrow$ expected_hand $-$ mean(expected_hand, axis=0)

Scale hands to same size and save norms for descaling
predicted_hand_norm $\leftarrow$ norm(predicted_hand, axis=0)
$$\text{predicted_hand} \leftarrow \frac{\text{predicted_hand}}{\text{predicted_hand_norm}}$$
expected_hand_norm $\leftarrow$ norm(expected_hand, axis=0)
$$\text{expected_hand} \leftarrow \frac{\text{expected_hand}}{\text{expected_hand_norm}}$$

Use rotation and/or reflection to minimise disparity
$$U,\ \Sigma,\ V^\top \leftarrow \text{svd}\left(\left[\text{predicted_hand}^\top \cdot \text{expected_hand}\right]^\top\right)$$
$R \leftarrow U \cdot V^\top$ # Compute rotation and/or reflection matrix
$s \leftarrow \text{sum}(\Sigma)$ # Compute scaling factor for applying after rotation and/or reflection
expected_hand $\leftarrow \left(\text{expected_hand} \cdot R^\top\right) \cdot s$ # Apply rotation and/or reflection

Descale hands and compute the mean euclidean distance
predicted_hand $\leftarrow$ predicted_hand $\cdot$ predicted_hand_norm
expected_hand $\leftarrow$ expected_hand $\cdot$ expected_hand_norm
mean_euclidean_distance $\leftarrow$ MED(predicted_hand, expected_hand)

Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG).

Fig. 1A-C, 2A, 3A, 4A, and A1 are created with BioRender.com for which we wish to acknowledge it.

References

[1] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3D pose estimation and tracking by detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 623–630, June 2010. ISSN: 1063-6919.

[2] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, pp. 1281–1289, Sept. 2018.

[3] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3D Human Pose Estimation With Spatial and Temporal Transformers," pp. 11656–11665, 2021.

[4] Oculus, "Presence Platform's Hand Tracking API Gets an Upgrade," Apr. 2022.

[5] D. M. Wolpert and Z. Ghahramani, "Computational principles of movement neuroscience," *Nature Neuroscience*, vol. 3, pp. 1212–1217, Nov. 2000. Number: 11 Publisher: Nature Publishing Group.

[6] A. A. Faisal, L. P. J. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nature Reviews Neuroscience*, vol. 9, pp. 292–303, Apr. 2008. Number: 4 Publisher: Nature Publishing Group.

[7] M. Latash, M. Levin, J. Scholz, and G. Schöner, "Motor control theories and their applications," *Medicina*, vol. 46, p. 382, June 2010.

[8] R. Merletti and P. Parker, eds., *Electromyography: physiology, engineering, and noninvasive applications*. IEEE Press series in biomedical engineering, Hoboken, NJ: IEEE/John Wiley & Sons, 2004. OCLC: ocm56361488.

[9] E. Musk and Neuralink, "An Integrated Brain-Machine Interface Platform With Thousands of Channels," *Journal of Medical Internet Research*, vol. 21, p. e16194, Oct. 2019.

[10] Meta, "Inside Facebook Reality Labs: Wrist-based interaction for the next computing platform," Mar. 2021.

[11] H. S. Milner-Brown and R. B. Stein, "The relation between the surface electromyogram and muscular force.," *The Journal of Physiology*, vol. 246, pp. 549–569, Apr. 1975.

[12] S. Dosen, M. Markovic, K. Somer, B. Graimann, and D. Farina, "EMG Biofeedback for online predictive control of grasping force in a myoelectric prosthesis," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, p. 55, Dec. 2015.

[13] N. Celadon, S. Došen, I. Binder, P. Ariano, and D. Farina, "Proportional estimation of finger movements from high-density surface electromyography," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, p. 73, Dec. 2016.

[14] M. Stachaczyk, S. F. Atashzar, S. Dupan, I. Vujaklija, and D. Farina, "Toward Universal Neural Interfaces for Daily Use: Decoding the Neural Drive to Muscles Generalises Highly Accurate Finger Task Identification Across Humans," *IEEE Access*, vol. 8, pp. 149025–149035, 2020. Conference Name: IEEE Access.

[15] S. Zabihi, E. Rahimian, A. Asif, and A. Mohammadi, "TraHGR: Transformer for Hand Gesture Recognition via ElectroMyography," *arXiv:2203.16336 [cs, eess]*, Mar. 2022. arXiv: 2203.16336.

[16] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, "Hand Gesture Recognition Using Temporal Convolutions and Attention Mechanism," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1196–1200, May 2022. ISSN: 2379-190X.

[17] A. E. Olsson, N. Malešević, A. Björkman, and C. Antfolk, "Learning regularized representations of categorically labelled surface EMG enables simultaneous and proportional myoelectric control," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, p. 35, 2021.

[18] N. Jose, R. Raj, P. K. Adithya, and K. S. Sivanadan, "Classification of forearm movements from sEMG time domain features using machine learning algorithms," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 1624–1628, Nov. 2017. ISSN: 2159-3450.

[19] F. Quivira, T. Koike-Akino, Y. Wang, and D. Erdogmus, "Translating sEMG signals to continuous hand poses using recurrent neural networks," in *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pp. 166–169, Mar. 2018.

[20] D. Yang and H. Liu, "An EMG-based Deep Learning Approach for Multi-DOF Wrist Movement Decoding," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021. Conference Name: IEEE Transactions on Industrial Electronics.

[21] K. Kumarasinghe, N. Kasabov, and D. Taylor, "Brain-inspired spiking neural networks for decoding and understanding muscle activity and kinematics from electroencephalography signals during hand movements," *Scientific Reports*, vol. 11, no. 1, p. 2486, 2021.

[22] C. Ma, W. Guo, L. Xu, and G. Li, "Finger Joint Angle Estimation based on sEMG signals and deep learning method," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 638–643, July 2021.

[23] M. A. García-Vellisca, A. Matran-Fernandez, R. Poli, and L. Citi, "Hand-movement Prediction from EMG with LSTM-Recurrent Neural Networks," in *2021 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE)*, pp. 1–5, Mar. 2021. ISSN: 2327-817X.

[24] W. Guo, C. Ma, Z. Wang, H. Zhang, D. Farina, N. Jiang, and C. Lin, "Long exposure convolutional memory network for accurate estimation of finger kinematics from surface electromyographic signals," *Journal of Neural Engineering*, vol. 18, p. 026027, Apr. 2021.

[25] M. Zanghieri, S. Benatti, A. Burrello, V. J. Kartsch Morinigo, R. Meattini, G. Palli, C. Melchiorri, and L. Benini, "sEMG-based Regression of Hand Kinematics with Temporal Convolutional Networks on a Low-Power Edge Microcontroller," in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, pp. 1–6, Aug. 2021.

[26] K. Z. Zhuang, N. Sommer, V. Mendez, S. Aryan, E. Formento, E. D'Anna, F. Artoni, F. Petrini, G. Granata, G. Cannaviello, W. Raffoul, A. Billard, and S. Micera, "Shared human–robot proportional control of a dexterous myoelectric prosthesis," *Nature Machine Intelligence*, vol. 1, pp. 400–411, Sept. 2019.

[27] C. Chen, Y. Yu, X. Sheng, D. Farina, and X. Zhu, "Simultaneous and proportional control of wrist and hand movements by decoding motor unit discharges in real time," *Journal of Neural Engineering*, vol. 18, p. 056010, Oct. 2021.

[28] A. Nasr, S. Bell, J. He, R. L. Whittaker, N. Jiang, C. R. Dickerson, and J. McPhee, "MuscleNET: mapping electromyography to kinematic and dynamic biomechanical variables by machine learning," *Journal of Neural Engineering*, vol. 18, p. 0460d3, Aug. 2021. Publisher: IOP Publishing.

[29] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: A toolkit for robust markerless 3D pose estimation," *Cell reports*, vol. 36, p. 109730, Sept. 2021.

[30] A. Del Vecchio, A. Holobar, D. Falla, F. Felici, R. Enoka, and D. Farina, "Tutorial: Analysis of motor unit discharge characteristics from high-density surface EMG signals," *Journal of Electromyography and Kinesiology*, vol. 53, p. 102426, Aug. 2020.

[31] M. M. Liu, W. Herzog, and H. H. Savelberg, "Dynamic muscle force predictions from EMG: an artificial neural network approach," *Journal of Electromyography and Kinesiology*, vol. 9, pp. 391–400, Dec. 1999.

[32] F. Negro, A. Holobar, and D. Farina, "Fluctuations in isometric muscle force can be described by one linear projection of low-frequency components of motor unit discharge rates," *The Journal of Physiology*, vol. 587, no. 24, pp. 5925–5938, 2009.

[33] A. Del Vecchio, F. Negro, F. Felici, and D. Farina, "Associations between motor unit action potential parameters and surface EMG features," *Journal of Applied Physiology*, vol. 123, pp. 835–843, Oct. 2017.

[34] J. E. Ting, A. Del Vecchio, D. Sarma, N. Verma, S. C. Colachis, N. V. Annetta, J. L. Collinger, D. Farina, and D. J. Weber, "Sensing and decoding the neural drive to paralyzed muscles during attempted movements of a person with tetraplegia using a sleeve array," *Journal of Neurophysiology*, vol. 126, pp. 2104–2118, Dec. 2021.

[35] S. Chandra, J. Li, B. Afsharipour, A. F. Cardona, N. L. Suresh, L. Tian, Y. Deng, Y. Zhong, Z. Xie, H. Shen, Y. Huang, J. A. Rogers, and W. Z. Rymer, "Performance Evaluation of a Wearable Tattoo Electrode Suitable for High-Resolution Surface Electromyogram Recording," *IEEE Transactions on Biomedical Engineering*, vol. 68, pp. 1389–1398, Apr. 2021.

[36] A. Del Vecchio, F. Sylos-Labini, V. Mondì, P. Paolillo, Y. Ivanenko, F. Lacquaniti, and D. Farina, "Spinal motoneurons of the human newborn are highly synchronized during leg movements," *Science Advances*, vol. 6, p. eabc3916, Nov. 2020.

[37] R. C. Sîmpetru, M. Osswald, D. I. Braun, D. S. Oliveira, A. L. Cakici, and A. Del Vecchio, "Accurate Continuous Prediction of 14 Degrees of Freedom of the Hand from Myoelectrical Signals through Convolutive Deep Learning," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, p. 5, July 2022.

[38] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "Data Augmentation of Surface Electromyography for Hand Gesture Recognition," *Sensors*, vol. 20, p. 4892, Aug. 2020.

[39] D. Purves and S. M. Williams, eds., *Neuroscience*. Sunderland, Mass: Sinauer Associates, 2nd ed ed., 2001.

[40] S. J. Day and M. Hulliger, "Experimental Simulation of Cat Electromyogram: Evidence for Algebraic Summation of Motor-Unit Action-Potential Trains," *Journal of Neurophysiology*, vol. 86, pp. 2144–2158, Nov. 2001. Publisher: American Physiological Society.

[41] J. V. Basmajian and C. J. De Luca, *Muscles alive: their functions revealed by electromyography*. Baltimore: Williams & Wilkins, 5th ed ed., 1985.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *33rd Conference on Neural Information Processing Systems*, p. 12, 2019.

[43] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," 2019.

[44] K. Biswas, S. Kumar, S. Banerjee, and A. K. Pandey, "SMU: smooth activation function for deep networks using smoothing maximum technique," *arXiv:2111.04682 [cs]*, Apr. 2022. arXiv: 2111.04682.

[45] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv:1711.05101 [cs, math]*, Jan. 2019. arXiv: 1711.05101.

[46] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," Feb. 2018.

[47] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates," *arXiv:1708.07120 [cs, stat]*, May 2018. arXiv: 1708.07120.

[48] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, no. 2, pp. 285–321, 1991.

Appendix A
Supplementary Figures

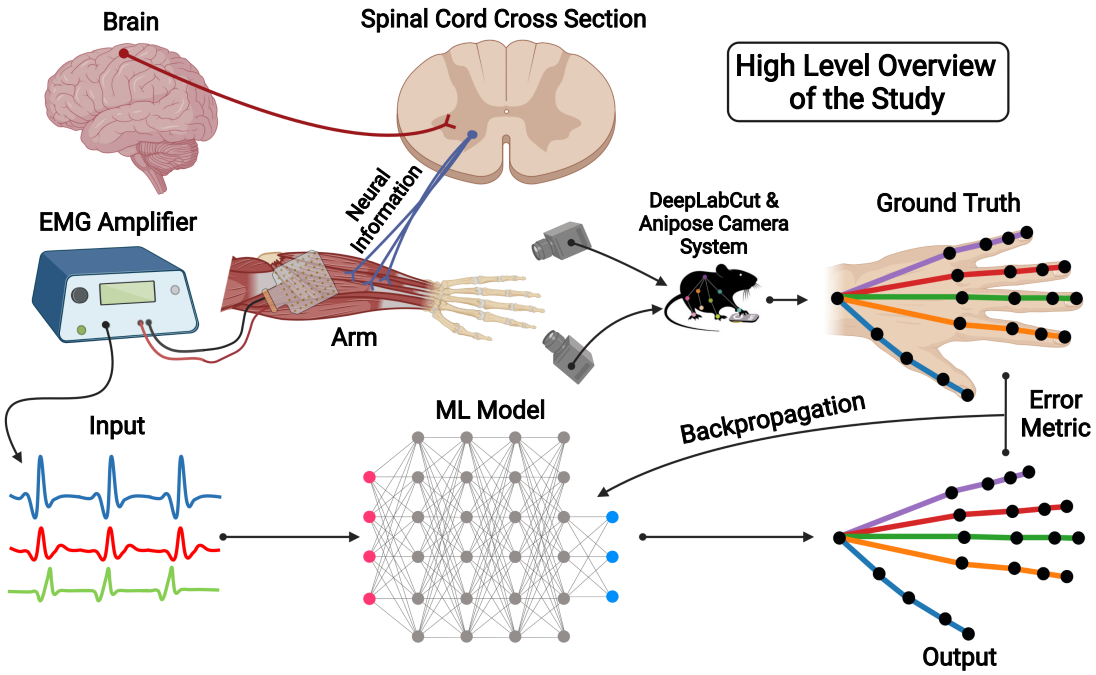

Fig. A1. Extended high level overview of the study showing the neurophysiological pathway used for EMG generation (brain $\longrightarrow$ spinal cord $\longrightarrow$ arm muscles) and the framework used for the training of the ML model.

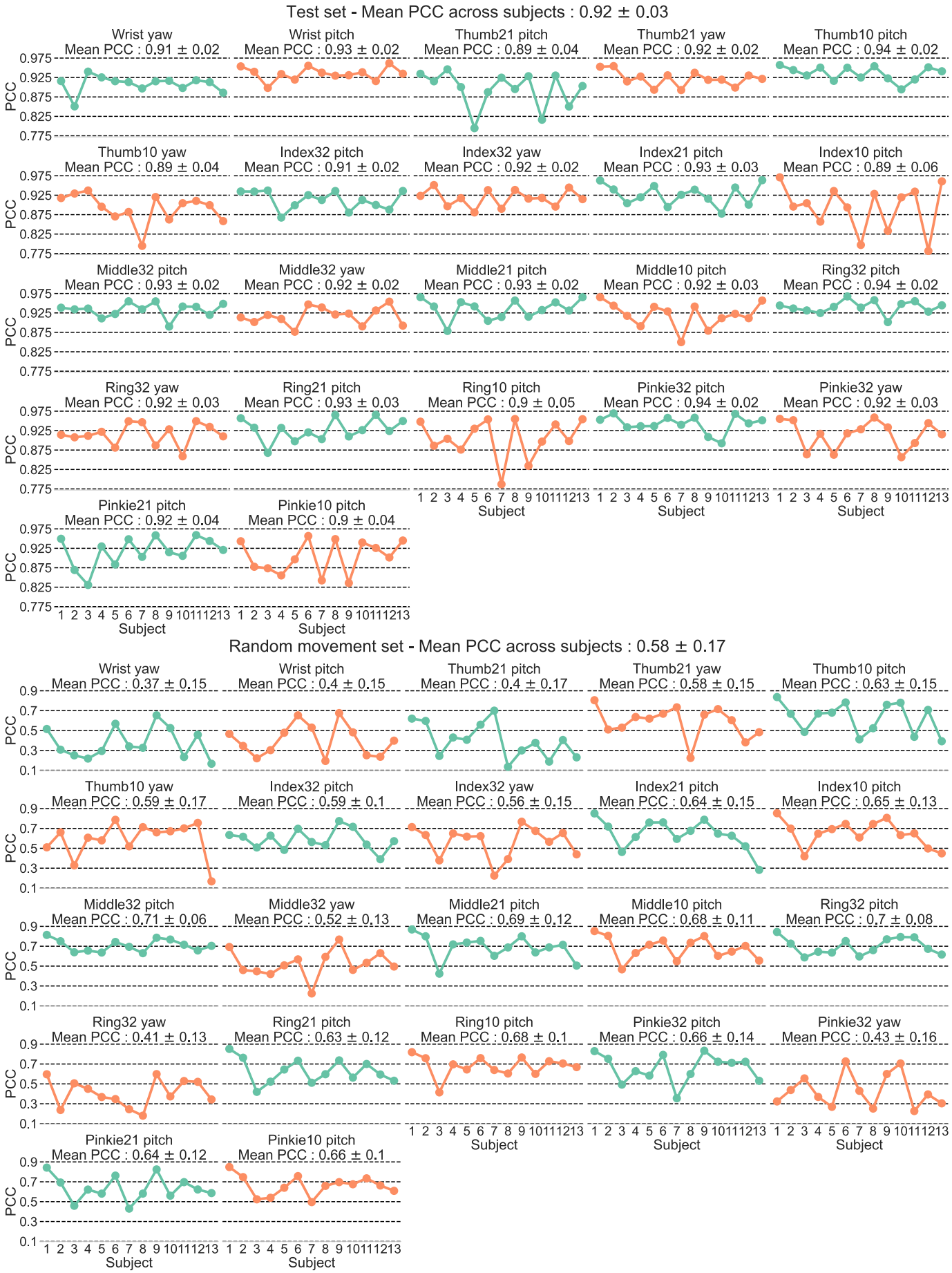

Fig. A2. Individual joint angle correlations across subjects.

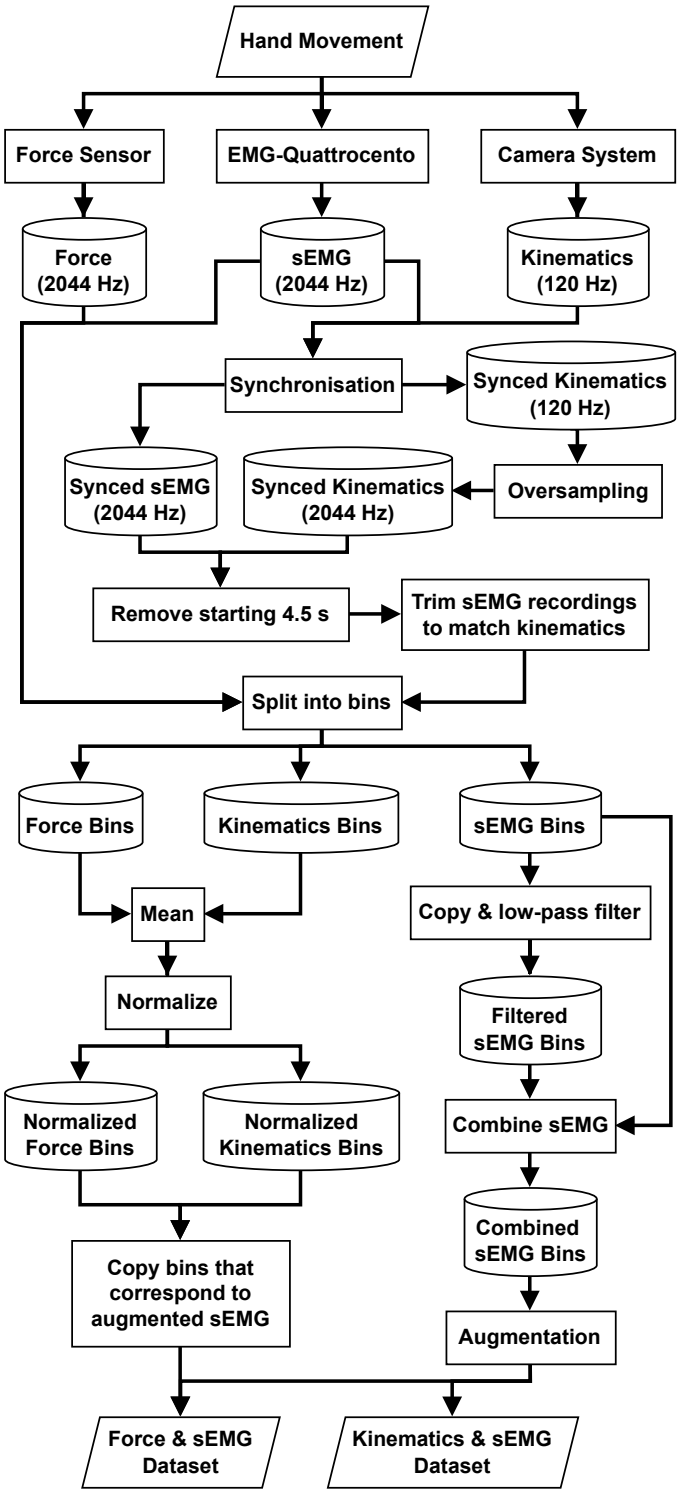

Fig. A3. Data preprocessing flowchart.