

# A convolutional neural network for robotic arm guidance using sEMG based frequency-features

Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall,

Philippe Giguère, Clément Gosselin, *Fellow, IEEE*, François Laviolette and Benoit Gosselin, *Member, IEEE*

**Abstract**—Recently, robotics has been seen as a key solution to improve the quality of life of amputees. In order to create smarter robotic prosthetic devices to be used in an everyday context, one must be able to interface them seamlessly with the end-user in an inexpensive, yet reliable way. In this paper, we are looking at guiding a robotic device by detecting gestures through measurement of the electrical activity of muscles captured by surface electromyography (sEMG). Reliable sEMG-based gesture classifiers for end-users are challenging to design, as they must be extremely robust to signal drift, muscle fatigue and small electrode displacement without the need for constant recalibration. In spite of extensive research, sophisticated sEMG classifiers for prostheses guidance are not yet widely used, as systems often fail to solve these issues simultaneously. We propose to address these problems by employing Convolutional Neural Networks. Specifically as a first step, we demonstrate their viability to the problem of gesture recognition for a low-cost, low-sampling rate (200Hz) consumer-grade, 8-channel, dry electrodes sEMG device called *Myo armband* (Thalmic Labs) on able-bodied subjects. To this effect, we assessed the robustness of this machine learning oriented approach by classifying a combination of 7 hand/wrist gestures with an accuracy of  $\sim 97.9\%$  in real-time, over a period of 6 consecutive days with no recalibration. In addition, we used the classifier (in conjunction with orientation data) to guide a 6DoF robotic arm, using the armband with the same speed and precision as with a joystick. We also show that the classifier is able to generalize to different users by testing it on 18 participants.

## I. INTRODUCTION

The commoditization of robots and sensors creates new opportunities to integrate robotics into day-to-day life. In particular, some of these developments aim at easing or aiding in common everyday tasks. For those who depend on prostheses and assistive robots, such developments can significantly improve their quality of life [1], [2]. In order to leverage the full potential of robotic devices in this context, it is essential to develop novel and intuitive ways to control them. An ideal interface would also be as intuitive and inconspicuous as possible, to provide a seamless experience to non-expert users.

One possible way to achieve such a natural interface is through Surface electromyography (sEMG). It is a non-invasive technique, extensively adopted in clinical and research works related to muscular activities. sEMG signals are non-stationary, and represent the sum of subcutaneous motor

unit action potentials generated during a muscular contraction [3]. The use of sEMG signals, combined with pattern recognition systems, has been proposed in the literature as an effective avenue to provide a more intuitive control of devices such as prosthesis or assistive robots [3], [4]. Studies in this topic mainly employ several sEMG electrodes placed on specified muscles to perform forearm pattern recognition. Furthermore, when using gel-based electrodes, the user's skin has to be shaved and washed to obtain optimal contact between the electrodes and the skin. This severely limits the practicability of such systems by making the preparation step a long, delicate and complex process.

In order to be able to use sEMG signals for robotic guidance, pattern recognition must be performed to identify a user's gesture. The two main components of pattern recognition are feature extraction and classification. For sEMG, features extracted from the time-domain have been extensively studied [5], [6] (*e.g.* Mean Absolute Value, Zero Crossing, Willison amplitude and Integrated EMG). However, as mentioned in [7], if temporal features are fast and easy to implement, they are sensitive to frequent amplitude fluctuations compared to features from the frequency-domain (*e.g.* Fourier Transform, Median Frequency, Mean Frequency [8], [6]). Features in the time-frequency domain (*e.g.* spectrograms, wavelet transform, wavelet packet transform) provide a richer way to extract pattern information [9]. For classification, many methods (linear, non-linear, supervised or unsupervised) have been employed to estimate unknown patterns from a set of features [3], [4]. The most common methods are the linear discriminant analysis (LDA) and artificial neural networks (ANN). Even if each classifier presents its own advantages, they remain too sensitive to electrode displacement and positioning when used with sEMG electrodes placed on specific muscles [10].

In the case of a prosthesis, the guidance system should ideally be small, inexpensive, lightweight, require minimal preparation and be robust to a small displacement of the electrodes while still achieving excellent classification performance. Dry electrodes should also be preferred over gel-based ones, as they are inherently more convenient to use. However, they are less accurate and less robust to motion artifact, compared to gel-based ones [11].

The work presented in this paper addresses these severe limitations while still achieving state of the art results. Our approach is based on employing convolutional neural networks (CNN) to perform the classification of spectrograms of the sEMG signals, in order to identify a number of

Ulysse Côté Allard, François Nougrou, Cheikh Latyr Fall and Benoit Gosselin are with the department of Computer and Electrical Engineering, François Laviolette and Philippe Giguère are with the Department of Computer Science and Software Engineering, and Clément Gosselin is with the Department of Mechanical Engineering, Université Laval, Québec, Québec, G1V 0A6, Canada. Contact author email: [ulyssse.cote-allard.1@ulaval.ca](mailto:ulyssse.cote-allard.1@ulaval.ca)

gestures. One of the contributions of this work is thus the use of a CNN to classify the very noisy sEMG data [12]. Indeed although they have been used before in speech recognition [13] and EEG classification [14], we believe this is the first time that they are used to classify sEMG data for gesture recognition. Importantly, the use of CNNs shifts the focus from *feature engineering* to *feature learning*. Indeed, because of the nature of CNNs, features are automatically learned via the convolutional layers. Those features are then transferred to fully connected layers that associate sEMG signals to specified gestures.

Although deep networks are often seen as computationally expensive, recent development in hardware for deep learning makes complex algorithms implementation in embedded systems a reality [15]. Additionally, dedicated deep learning materials such as Eyeriss [16] are able to run CNNs with up to 60 millions parameters at 35 fps using only 278 mW. Furthermore, those very-low power systems only need to handle the inference step since training can be done on a desktop and the weights of the parameters simply sent via bluetooth once the optimization is completed. Finally, using network pruning, one can achieve a compression rate over 10x [17] which significantly reduces both inference time and memory space requirement of the hardware.

Once the classifier is able to reliably identify these gestures, one can easily create a guiding system by associating a particular robot motion primitive to a gesture. To obtain the sEMG data in the least possible intrusive way, a Myo armband by Thalmic Labs<sup>1</sup> can be used. This consumer-grade device includes a 9-degree-of-freedom (DoF) Inertial Measurement Unit (IMU) and 8 dry surface electromyogram sensors. The proposed system use both the IMU and sEMG data for the guidance of the robotic arm.

The rest of the paper goes as follow. First, Section II provides an overview of our guidance system. The architecture of our CNN classifier is then described in Section III, along with the spectrogram features that are employed. The ability of the classifier to generalize well to different users is showcased in Section IV-A. We then demonstrate the precision and the robustness of this guidance system by performing precise and complex tasks in a speed test in Section IV-B. Finally, Section IV-C establishes the long term stability of the classifier and its robustness to muscle fatigue by testing it over two periods of six consecutive days on a healthy subject. For each period, the emplacement of the electrodes on the forearm were not marked and no recalibration was performed after the initial training. This naturally led to small displacements of the electrodes between each recording session, which the classifier was not specifically trained to resist but was nonetheless robust to.

## II. PROPOSED GUIDANCE SYSTEM OVERVIEW

The description of the robotic arm, the Myo and the gestures used in this work are given in this Section. The classifier itself is detailed in Section III.

### A. Myo Armband

In 2014, the Myo armband was released at a purchase cost of 200 \$. As stated previously, it contains a 9DoF IMU and 8 dry surface electromyogram sensors. One of the main advantages of the Myo is that it can simply be slipped on the arm to read sEMG signals with no preparation. The sEMG data from the Myo can be visualized in [18]. The Myo armband provides a sEMG sampling frequency ( $f_s$ ) of 200 Hz per channel (a  $f_s$  of at least 1 kHz is normally preferred to address sEMG signals lying within 5-450 Hz [19]). Electrode placement was dependent on the size of the subject's forearm due to the minimum circumference of the Myo (19.05 cm). Additionally, no shaving of hair or skin-cleaning were performed for this study as these were judged too constraining for a potential end-user. Note that this generates extra noise that can nevertheless be handled by our machine learning approach.

### B. JACO guidance using a Myo armband

The JACO arm by Kinova<sup>2</sup> is a 6DoF robotic arm that is usually manually operated using a 7-button joystick. The coordinate system used by the robot is Euler X-Y-Z. The user can navigate through 3 different modes to access the full motion of the robotic arm: 1) translate (Move the arm along X-Y-Z axes), 2) rotate (Rotation of the robotic hand around the X-Y-Z axes) and 3) grip (open-closing the hand).

The different gestures employed to generate sEMG patterns are explained in Section II-C and replace the rotation around the X-Y axis and the grip mode. The rotation around the Z axis (corresponding to the roll motion on JACO's hand) is not considered in this work since it was deemed not useful in the experiments described in Section IV.

The translate mode is mapped to the orientation of the armband which is obtained using the IMU included inside the Myo. The yaw corresponding to the X-axis, the pitch to the Y-axis and the roll to the Z-axis. Furthermore, for the translate mode, the mapping is proportional to the angle between the orientation of the forearm and a horizontal reference orientation corresponding to a 90 degrees flexion of the elbow. Thus, the larger the angle between the neutral configuration of the forearm and the current orientation of the Myo, the faster JACO will move.

JACO also includes a spasm filter which is used solely for the third task in Section IV-A.3 at the lowest possible setting. The spasm filter limits the acceleration of JACO, while not affecting its deceleration speed. It is important to note that opening and closing the hand is not affected by this filter.

When guiding JACO with the Myo, a system of movement priority is established. Translation along the X and Y axes have the highest priority. If the arm of the user is within the resting position range with respect to the pitch and yaw, then the user can perform a translation of the robotic arm along the Z-axis by rotating his wrist (pronation and supination). Finally, if the roll is also within its neutral range, the classifier output is used to guide the robot. This

<sup>1</sup><https://www.myo.com/>

<sup>2</sup><http://kinovarobotics.com/>

priority list is necessary because inexperienced users, have a tendency to rotate their wrist involuntarily when moving their arm. Furthermore the classifier has a lower priority since the muscle activity generated by the rotation of the wrist, is not an activity the classifier is trained to recognize and therefore the output during this movement is not reliable.

### C. Description of the gestures

Since the final purpose of this experiment is to guide the robotic arm with the Myo armband, seven different hand/wrist gestures are required. The classes are: neutral, hand open, hand close, wrist flexion, wrist extension, radial deviation and ulnar deviation. Fig. 1 shows the different gestures as well as the Myo and JACO. These gestures are chosen because they can intuitively be mapped to the rotate and grip mode of the robotic arm. The gestures and their corresponding action of the robotic arm, are as follows: opening and closing of the hand to represent the opening and closing of the robotic hand, the wrist flexion and extension correspond to moving the joystick to the left and right in the rotate mode. Finally, the radial and ulnar deviation emulate pushing the joystick to the front and the back respectively.



Fig. 1. The 7 gestures considered in this work. The Myo armband (right) is connected through bluetooth to the laptop. The computer, after data analysis, transfer the command to JACO (left) in real-time via a USB connection.

## III. CLASSIFIER OVERVIEW

Processing of the sEMG data signal was necessary to be able to recognize the different hand/wrist movements of the user. In this section, the different steps performed during an online classification are exposed. We first describe how the data is separated into time windows, then pre-processed using consecutive fast Fourier transform (FFT) for forming one spectrogram per channel, and finally fed to a CNN to predict the current hand/wrist movement.

### A. Time-window

As stated previously, the Myo armband includes 8 sEMG channels, each sampled at 200 Hz. For closed loop and online operation, latency is an important parameter to consider. In [20], it was first recommended that the time-window between two predictions be equal to or less than 300 ms, while in [21] it was found that ideally, the latency should be between 100 and 125 ms. However, in [22] it was reported that the performance of the classifier should take priority over speed. In our system, we opted for a maximal latency of 300 ms, in order to accumulate a sufficient number of

samples with the low  $f_s$  of the Myo, and thus increase classification performance. Considering that the time to process and classify one gesture's sEMG-pattern window of  $\sim 300$  ms takes on average  $\sim 15$  ms on our hardware (laptop with an NVIDIA GeForce GT 555M), we used windows of 285 ms. This corresponds to 57 data points per channel per example. Overall, this kept the data capture and processing time below our target latency of 300 ms.

### B. Preprocessing

Spectrograms are calculated for each sEMG channel of 57 samples using windows of 30 points for FFT and an overlap of 21. Based on these parameters, 4 FFT will be contained in the spectrograms. This results in a spectrogram matrix of 16 by 4, with a frequency step of 6.67 Hz. Note that a Hamming window is used to avoid frequency leakage. The spectrograms are calculated using Scipy implementation in Python [23]. The first row of the spectrogram array is removed because it is out of the useful frequency range of the sEMG signal (Section II-A) The final spectrograms have a frequency range of 6.67 to 100 Hz.

### C. Classification Algorithm

We tried most of the state of the art machine learning algorithms (e.g. Support vector machine, Adaboost, Random Forest, Deep neural network). Considering that a CNN achieved by far the best results, it was selected as the classifier for our system. Its architecture is described below.

1) *Description*: The classification algorithm consists of a two-staged CNN, implemented using the python library Theano [24], [25]. This library allows the CNN to run on a GPU, thereby accelerating the training and prediction. The first stage is used to differentiate between the neutral class and the others. If the former is not detected, the algorithm proceeds to Stage 2, which differentiates between the remaining six gestures. Justification for this separation is at the bottom of this *Description*. The architecture of the CNN remains the same in both stages, except for the output layer which contains two and six neurons respectively. The architecture of stage 2 (containing  $\sim 3.6$  millions parameters) was selected as usual in deep learning by trial and error using previously published architecture as inspiration (mainly [26]) and is presented in Fig. 2.

We use ADADELTA [27] for the optimization of the CNN weights. The hyperbolic tangent (tanh) is used as the non-linear activation function. The rectified linear Unit (ReLU) has been considered mainly for speed reasons [28]. However ReLU was not retained because even though each iteration experienced a slight speed boost as expected, the CNN tended to converge faster with tanh and the accuracy in validation was similar between the two. The sigmoid function was also considered, but performed poorly for this task compared to both tanh and ReLU. Additionally, the proposed system uses the dropout approach [29] to prevent overfitting. For the convolutions layers, the dropout is set at 25%. For the two fully connected layers before merging it is at 50%. Finally, the dropout of the last two layers is set at 75%.

In the implementation of Stage 2, we first go through a rescaling step. First, considering the eight spectrograms at time  $T$  as a 3D matrix of shape  $8 \times 15 \times 4$ , we reshape it into a  $480 \times 1$  vector  $x_T$ . Then rescaling is performed with  $\frac{x_T^i}{\text{norm}(x_T)}$ , where  $x_T^i$  is the value in position  $i$  of the vector  $x_T$  and  $\text{norm}$  is the L2 norm. Performing the rescaling on the concatenated spectrograms ensures that their relative intensities are taken into account. After rescaling, the eight spectrograms are reshaped back into their  $15 \times 4$  format, feeding the two-stage CNN with 2D images. This reshaping preserves important correlations between channels and within spectrograms. The L2 norm achieves a trade-off between a quasi-constant power spectrum (within a factor of L2) and putting more weight on frequencies with higher power spectrum, an approach similar to extracting peaks in a power spectrum for time-series classification [30]. This last effect is less noticeable when using the L1 norm instead of the L2 norm. Furthermore, we observed faster convergence rate of the CNN using the L2 norm over the L1 norm. Note that when using a single-stage approach with the rescaling, the performances of the classifier degrades significantly. We attribute this to the fact that the rescaling step tends to normalize the energy level of incoming spectrograms, before they are fed to the CNN. Since discriminating between a neutral gestures (homogeneously low energy) and the other gestures (high muscle activity) probably relies on this energy level, this would explain the poor performance of this rescaled, single-stage CNN approach. The two-staged CNN approach that we have adopted sidesteps this issue completely, by performing rescaling *only* after a gesture has been deemed as non-neutral.

2) *Training and validation*: In the training phase, the process of collecting labeled data for the CNN required the user to hold each gestures for 5 s. These labeled intervals are then divided into time-window of 285 ms, as described in Section III-A. To see more accurately variation within the same class, each window overlapped the previous one by 265 ms. This process is repeated three times, yielding 15 s of data per class. We take three trials of 5 s per gesture as opposed to one trial of 15 s to get more variation on the same gesture. Indeed, a user cannot perform exactly the same motion with the same strength when asked to do the same gesture twice. This also follows the recommendation made by [31] of varying strength recording for the same gesture. Validation data is created in a fourth independent run (5 s per gestures) in an identical manner.

#### IV. EXPERIMENTAL RESULTS

Experiments were conducted to evaluate the performance of the CNN classifier (described in Section III) on three main aspects that correspond to Section IV-A, IV-B and IV-C respectively. The experiments of Section IV-A assessed the ability of the classifier to generalize to different individuals. Then, experiments in Section IV-B were used to compare the task completion time between an expert in guiding the robot with the joystick against one well-versed with our Myo interface. Finally, the classifier’s robustness to sEMG

signal drift [32], small electrode displacement and short term muscle fatigue is evaluated in Section IV-C. All results reported here were based on the zero-one loss accuracy. Meaning that a classification is considered successful only if the predicted gesture is exactly the one being made.

##### A. Generalization Experiment

We tested our system on 18 (11 men and 7 women) healthy subjects aged between 23 and 29 years old. The Myo armband was placed at a single but different location on the forearm, depending on the user. Indeed, since the armband minimum circumference is 19.05 cm and the test subjects had a forearm circumference between 15.5 and 24.0 cm (measured 5 cm above the wrist), it would have been difficult to obtain the same forearm sensor placement for each subject. Therefore, we set the armband at the minimum circumference and simply slid it up until the forearm’s circumference matched the armband’s one. The placement of the armband was thus directly dependent on the subject’s forearm circumference. Consequently, it is important that the performance of our approach be as independent as possible from the placement of the electrodes.

Training of the CNN was realized as described in Section III-C.2. The average accuracy in validation for the participants was 100% for the first stage and 97.71% for the second stage. The participants were then asked to perform three tasks: (1) gesture accuracy, (2) cube holding and (3) picking an object to place it in a specified zone. Details on these tasks are presented below.

1) *Gesture accuracy test (Task 1)*: To evaluate the accuracy of the classifier during short-term muscle fatigue, the participants were asked to hold one of seven gestures, chosen randomly, for 10 s. No rest was given between each gesture. The test length was 5 min, yielding 30 trial-gestures. The participants were noted on the amount of trial-gestures that they succeeded and given a score out of 30. A gesture was considered a success if no more than two false consecutive or no more than four non-consecutive miss-classifications occurred during a 10 s period. Transitioning between gestures was not considered in this task. The average success rate was 93.14% over all participants.

2) *Cube holding (Task 2)*: For the guidance of a robotic arm, the negative impact of miss-classification highly depends on the nature of the error. Indeed, if the user wants to close his hand and the classifier interpreted it as a neutral state, it is easy for the user to perform the gesture again. On the contrary, if the user wants to be in a resting position while the robotic arm is holding a glass of water and the resulting classification is hand open, serious consequences can be envisioned. The second task thus tested the capability of a user to hold an object in the robotic hand while making different gestures. The setup of this task was identical to Section IV-A.1 except that the gesture open hand was not requested and that the duration was 120 s. All participant successfully held the cube for 120 s during this task.

3) *Picking and placing cube (Task 3)*: In the final task, the participants were asked to pick a cube with the robotic

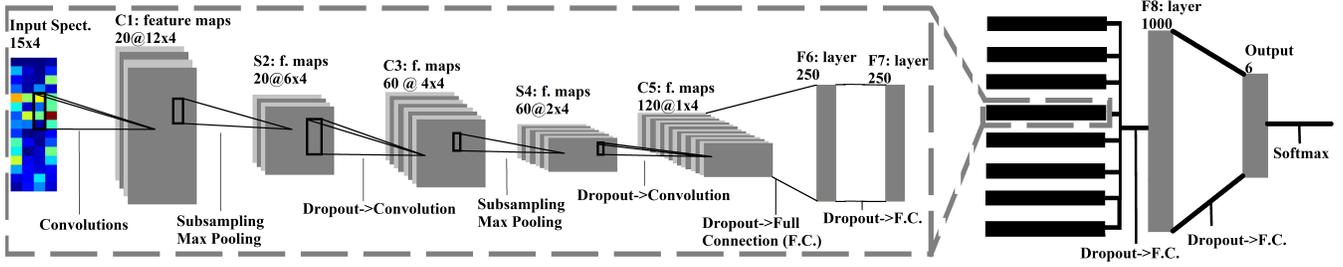


Fig. 2. Stage-2 architecture. Each channel is considered independently at first, going through the convolutional network for feature extraction and then two fully connected layers. The output from all channels on the layer F7 are then connected together on layer F8.

arm and put it in a specified location. The participants were first asked to perform the task with the normal guidance system for the robotic arm (joystick). They were timed for both picking and dropping the cube at the specified place. They then performed the same task with the Myo as the guidance system. For both tasks, they had 10 min of training prior to performing their task. It is important to note that the results that follow only aim at providing an order of magnitude for the time required to complete the task. Indeed, since the participants always started with the joystick, this gives an unfair advantage to the Myo armband system. The task is thus not suited to truly compare them in terms of speed, but simply to show that the classifier is sufficiently accurate to perform similarly to the joystick. The average time to perform the task with the joystick was: 1 min 45 s and with the Myo armband: 1 min 33 s.

4) *Results and discussions of the three tasks:* None of the participants had experience with sEMG-based classifiers or guidance of a robotic arm and had no known physical disabilities. Table I presents the general information on the participants as well as the results for the first two tasks. The circumference of the forearm was measured 5 cm above the wrist. The accuracy reported in Table I is the validation accuracy (Section III-C.2) for the two stages of the classifier. We can immediately see from the validation accuracy that the classifier is always able to learn to distinguish between the 7 gestures. Participants generally achieved high performance on Task 1 and all were perfect on Task 2.

TABLE I

PARTICIPANTS GENERAL INFORMATION AND FIRST TWO TASKS RESULTS

|                                    | Femme   |       | Homme   |       |
|------------------------------------|---------|-------|---------|-------|
|                                    | Moyenne | E.T.  | Moyenne | E.T.  |
| Age (Années)                       | 24.29   | 1.67  | 24.55   | 2.27  |
| Avant-bras<br>circonférence (cm)   | 17.0    | 1.36  | 19.73   | 2.85  |
| stage 1<br>précision en validation | 100.00% | 0.00% | 100.00% | 0.00% |
| stage 2<br>précision en validation | 97.61%  | 1.14% | 97.76%  | 2.93% |
| Task 1                             | 93.81%  | 5.17% | 92.72%  | 6.49% |
| Task 2 (s)                         | 120     | 0.0   | 120     | 0.0   |

Fig. 3 presents the time taken by the participant to complete Task 3. It is important to note that one of the participants did not complete the Myo portion of task 3.

The reason is that the classifier performed too poorly for the precise manipulation required from Task 3. The participant data from the other tasks are included in the statistics (76.66% for task 1 and 120 s for Task 2).

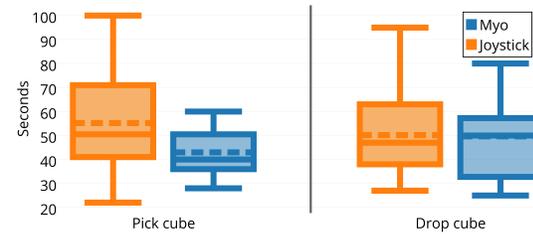


Fig. 3. Box plot of the time taken to complete task 3 (picking and placing the cube). The dotted line represent the mean of the distribution.

For inexperienced users, Fig. 3 shows that the time needed to complete task 3 with the joystick and the Myo are on the same order of magnitude. We cannot conclude that our guidance scheme is more intuitive because the time difference is not statistically significant (ANOVA  $p$ -value  $> 0.05$ ). We can however conclude that our system is robust enough to reliably guide the robotic arm in precise tasks.

### B. Speed test for a complex task

We specifically designed a speed challenge to evaluate the usefulness of the classifier in an online situation and to provide a time comparison between the joystick and the Myo. The task consisted in picking and placing three cubes consecutively, in a similar manner as described in Section IV-A.3. The joystick times were achieved by an expert in guidance of JACO. The expert had no physical disability. The challenge was performed three times for each guiding scheme. The reported results are the average over three runs. Fig. 4 compares each sub-task of the challenge (picking and dropping the three cubes). A video accompanying this article shows the task being performed with the Myo. It should be noted that the first cube grabbed and then dropped correspond exactly to task 3 as described in Section IV-A.3.

### C. Classifier stability

In order to assess the stability of the classifier, a set up similar to the one described in Section IV-A.1 was used.

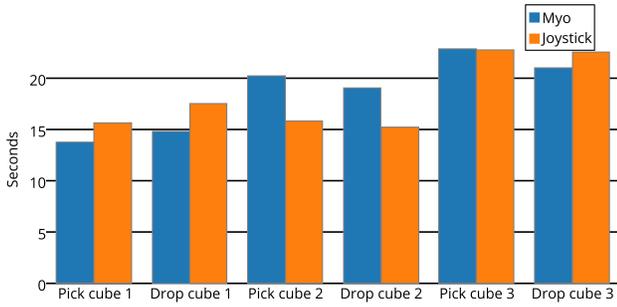


Fig. 4. Average time taken to complete each section of the speed challenge. Both the Myo and the joystick performed similarly, with the total average difference being less than 4 s in favor of the joystick.

The classifier was first trained on a subject at time  $T=0$ . To assess the accuracy of the classifier, the test subject had to hold a random gesture for 10 s was repeated for a full 5 min, without rest. The test was conducted at least twice a day for six consecutive days. No re-training was done after  $T=0$  and the armband was approximately at the same location on the forearm for each experiment (no marking to guide the user).

The accuracy was calculated by comparing the predicted gesture to the one requested by the computer. This however added errors well above the accuracy found when guiding the robot in real-time. Indeed, the time needed to read and start reacting to new instructions from the computer are considered errors in this setup. This is not a factor in a realistic guidance setting, where it is the user who decides when and which gesture to use. To mitigate this, we present two sets of results named : transition and no transition experiment. The first one does not try to alleviate the problems previously mentioned. The second simply does not consider the first 1.2 s after a change of gesture. The purpose of this is to remove the reaction-time, from the moment the subject receives the cue till the action is performed. Fig. 5 clearly shows that only a small performance degradation can be observed 6 days after training the classifier.

The two dotted lines of Fig 5 correspond to the linear regression lines. The first data point (at 1 hour) was not considered for the regression as it appears to be an outlier that would unjustifiably bias the results in favor of our proposed method. The poor performance of the first measurement, compared to the others, can be explained by the fact that the subject is still learning the decision function of the classifier. Another 6-day trial with a new classifier was run which yielded very similar results. Due to space consideration, they are not reported here. The experiment shows that our classifier is robust to small physical variations such as impedance of the skin changes from day to day and electrode placement inconsistency, muscle activity change, etc. In fact, the classifier shown in the video and the results for the Myo presented in Section IV-B were achieved using a classifier that was trained 12 days prior to the speed challenge. Showing that the same classifier can be used extremely efficiently even several days after training.

To assess if short-term muscle fatigue has a significant

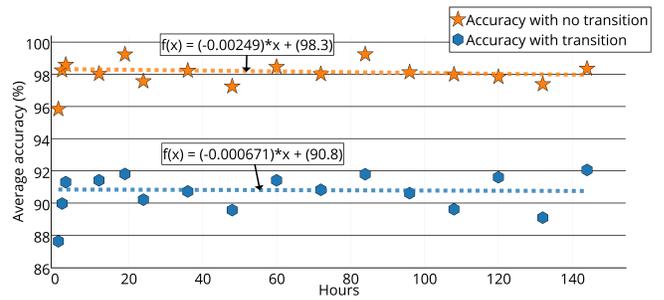


Fig. 5. Average accuracy of the first 6-day trials. The blue hexagons represent the accuracy over the complete 5- min period at different time after the training of the classifier. The orange stars is the accuracy over the 5- min period when omitting the first 1.2 s after each new gesture. The blue and orange dotted line come from the linear regression of the blue hexagons and orange stars data point respectively. The fact that the accuracy is almost constant through the 6-day period indicate that the classifier is robust to long-term use.

impact on the accuracy of the classifier, we examined its accuracy in 10 s intervals for a total of 5 min. We used the data obtained from the two 6-day trials, where a new gesture was requested every 10 s, yielding a total of 30 gestures over the 5 min period. Then the average accuracy for every 10 s gesture, over all the two periods of six days, was combined. This enabled a clear view of any possible degradation in the accuracy, as time elapsed in the 5 min sequence. It is clear from Fig. 6 that muscle fatigue did not degrade the performance of the classifier in any noticeable way.

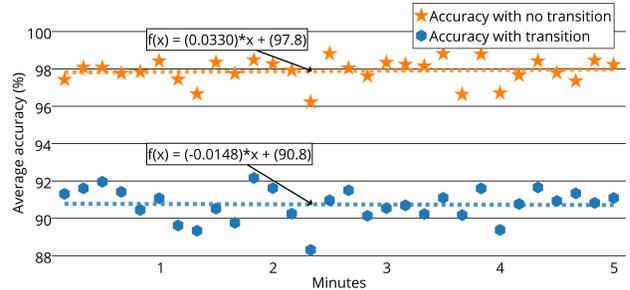


Fig. 6. Average accuracy over 5 min for the two 6-day trials. See Fig. 5 for the description of the dotted lines, blue hexagons and orange stars. The fact that the accuracy does not change over time shows that muscle fatigue is probably not adversely affecting our classifier.

The proposed classifier is thus not only accurate enough to perform complex and precise tasks, but is also robust to short term muscle fatigue, small displacement of electrodes and long term use without the need for recalibration. The average accuracy over the two 6-day trial is 97.9%.

## V. CONCLUSION

In this work, a Myo armband was used to guide a robotic arm. The use of the armband offers several advantages for the intended users (inexpensive, no preparation time, easy to use). However the efficiency of the armband comes at the cost of quantity and quality of information. One of the major accomplishments in this paper has been to show

that one can compensate for this lack of data quality with suitable machine learning approaches. Using the specified CNN architecture with spectrograms as input, our system was able to achieve state of the art results and obtain precise guidance of a 6DoF robotic arm using sEMG and orientation data that rivals the guidance with the joystick. The efficacy of the classifier was established when facing short-term muscle fatigue and long-term use achieving on average 97.9% during the two 6-day periods. The system was also shown to generalize effortlessly to different users.

Future work will focus on three main aspects. First, optimizing the CNN architecture in term of parameters and pruning it will allow both faster training and inference from the system. The presented approach will thus be more easily suitable for very low-power embedded hardware. Secondly, the data collected from the participants of this study will be used to build a classifier that will require significantly less training data than presently required for a new user. To achieve this we intend to make use of training based on domain-adaptation techniques, which recover suitable information from one task and applies it to a similar one (new user). This can be used in a deep learning setting [33]. Finally, the classifier will be tested on upper limb amputees.

#### ACKNOWLEDGMENT

The authors would like to thank Alexandre Campeau-Lecours for sharing his expertise on JACO and his participation in the speed challenge.

#### REFERENCES

- [1] A. M. Cook and J. M. Polgar, *Essentials of assistive technologies*. Elsevier Health Sciences, 2014.
- [2] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, "The role of assistive robotics in the lives of persons with disability," *American Journal of Physical Medicine & Rehabilitation*, vol. 89, no. 6, pp. 509–521, 2010.
- [3] M. Oskoei and H. Hu, "Myoelectric control systems – a survey," *Biomedical Signal Processing and Control*, vol. 2, pp. 275–294, 2007.
- [4] C. L. Fall, P. Turgeon, A. Campeau-Lecours, V. Maheu, M. Boukadoum, S. Roy, D. Massicotte, C. Gosselin, and B. Gosselin, "Intuitive wireless control of a robotic arm for people living with an upper body disability," *IEEE EMBC conf*, pp. 4399–4402, 2015.
- [5] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transaction on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, 2003.
- [6] A. Phinyomark, S. Hirunviriyaya, C. Limsakul, and P. Phukpattaranont, "Evaluation of emg feature extraction for hand movement recognition based on euclidean distance and standard deviation," *IEEE International Conference on Computer Telecommunications and Information Technology*, pp. 856–860, 2010.
- [7] T. Xueyan, L. Yunhui, L. Congyi, and S. Dong, "Hand motion classification using a multi-channel surface electromyography sensor," *Sensors*, pp. 1130–1147, 2012.
- [8] L. Hargrove, K. Englehart, and B. Hudgins, "A comparison of surface and intramuscular myoelectric signal classification," *IEEE Transaction on Biomedical Engineering*, vol. 54, no. 5, pp. 847–853, 2007.
- [9] K. Englehart, B. Hudgins, and P. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transaction on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.
- [10] A. Boschmann and M. Platzner, "Towards robust hd emg pattern recognition: Reducing electrode displacement effect using structural similarity," *IEEE EMBS conf*, pp. 4547–4550, 2014.
- [11] D. Stegeman and B. L. B.U. Kleine, "High-density surface emg: Techniques and applications at a motor unit level," *Biocybernetics and Biomedical Engineering*, vol. 32, no. 3, 2012.
- [12] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O. Aszmann, "The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797–809, 2014.
- [13] T. N. Sainath, A. r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618, 2013.
- [14] H. Cecotti and A. Graeser, "Convolutional neural network with embedded fourier transform for eeg classification," *19th International Conference on Pattern Recognition*, pp. 1–4, 2008.
- [15] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song *et al.*, "Going deeper with embedded fpga platform for convolutional neural network," in *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016, pp. 26–35.
- [16] Y. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers*, 2016, pp. 262–263.
- [17] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.
- [18] C. Yang, S. Chang, P. Liang, Z. Li, and C. Su, "Teleoperated robot writing using emg signals," *IEEE International Conference on Information and Automation*, pp. 2264–2269, 2015.
- [19] C. D. Luca, L. Gilmore, M. Kuznetsov, and S. Roy, "Filtering the surface emg signal: Movement artifact and baseline noise contamination," *Journal of Biomechanics*, vol. 43, no. 8, pp. 1573–1579, 2010.
- [20] B. Hudgins, P. Parker, and R. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transaction on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
- [21] T. Farrell and R. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transaction on Neural systems and rehabilitation engineering*, vol. 15, no. 1, pp. 111–118, 2007.
- [22] B. Peerdeman, D. Boere, H. Witteveen, R. H. in 't Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *Journal of Rehabilitation Research & Development*, vol. 48, no. 6, pp. 719–738, 2011.
- [23] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2016-02-28]. [Online]. Available: <http://www.scipy.org/>
- [24] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2012.
- [25] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2010.
- [26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 11, no. 86, pp. 2278–2324, 1998.
- [27] M. Zeiler, "Adadelata: An adaptive learning rate method," *arXiv preprint*, vol. arXiv:1212.5701, 2012.
- [28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [29] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint*, vol. arXiv:1207.0580, 2012.
- [30] N. Jamalie and C. Sammut, "Majority voting: Classification by tactile sensing using surface texture," *IEEE transactions on, Robotics*, vol. 27, no. 3, pp. 508–521, 2011.
- [31] D. Yang, W. Yang, Q. Huang, and H. Liu, "Classification of multiple finger motions during dynamic upper limb movements," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1–1, 2015.
- [32] S. Amsuss, L. Paredes, N. Rudigkeit, B. Graimann, M. Herrmann, and D. Farina, "Long term stability of surface emg pattern classification for prosthetic control recognition," *Engineering in Medicine and Biology Society (EMBC)*, pp. 3622–3625, 2013.
- [33] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, pp. 1–35, 2016.