

Stepping-stones to Transhumanism: An EMG-controlled Low-cost Prosthetic Hand for Academia

Karen Tatarian

*Department of Mechanical Engineering
American University of Beirut
Beirut, Lebanon
tatariankaren@gmail.com*

Diego R. Faria

*School of Engineering & Applied Science
Aston University
Birmingham, UK
d.faria@aston.ac.uk*

Micael S. Couceiro

*Ingeniarius, Lda
Institute of Systems and Robotics,
University of Coimbra,
Coimbra, Portugal
micael@ingeniarius.pt*

Eduardo P. Ribeiro

*Department of Electrical Engineering
Federal University of Parana
Curitiba, Brazil
edu@eletrica.ufpr.br*

Abstract—Over the past decades, humans have redesign themselves with the intent to evolve beyond their current physical and mental limitations. This phenomenon has been known as transhumanism, wherein robotics and biomimetics have been exploiting the unique designs of the human body with the intent to develop disruptive anthropomorphic artificial appendages. Nevertheless, while lower extremity prosthetics have evolved to the point at which lower leg amputees may be competitive with professional runners in the world, there is still a gap between upper extremity prosthetics and real hands. This work intends to be a pioneer into developing a low-cost multipurpose robotic hand for research and academia. This paper describes the robotic hand, including its electromechanical development and full ROS integration. Moreover, the paper also presents a MatLab framework designed to introduce sequence data classification, namely providing the ability to control the robotic hand using electromyography (EMG) signals from the forearm. This paper expects to contribute to an ever-increasing human-robot symbiosis by motivating students to engage in transhumanism studies using more sophisticated technologies and methods.

Keywords—*prosthetic hand, myoelectric hand, LSTM, EMG, ROS, MatLab, transhumanism, low-cost robotic hand.*

I. INTRODUCTION

With the fast-paced advances in Artificial Intelligence and Robotics, humans have been increasingly feeling the need to redesign themselves, be it to overcome the inevitability of aging, the limitations on human intellect, a given unchosen psychology, or any other “boundary”. This belief, or theory, that the human race can evolve beyond its current physical and mental limitations, especially by means of science and technology, has been known as transhumanism. Transhumanism is currently an “*international intellectual movement that aims to transform the human condition by developing and making widely available sophisticated technologies to greatly enhance human intellect and physiology*” [1]. The transhumanist movement led to a large investment into the potential for human “augmentation”, especially by researchers and technological companies working in the field of robotics who believe that mankind should adapt to an upcoming robotic society. While this might still be a controversial topic, an ever-increasing symbiosis between

human and robot seems inevitable, being currently divided in three categories: Robotic surrogates [2]; Wearable robotics [3]; Robotic Prosthetics [4].

This paper falls within the latter, contributing to the development of disruptive anthropomorphic artificial appendages which might, to some extent, exhibit human-like stability, strength, and speed within real-world context [5]. Such endeavours are relevant for a multitude of applications, including limb prosthetics for amputees, robotic surrogates for underwater and space applications, and exoskeletons for military combat enhancement [4]. Nevertheless, while lower extremity prosthetics have evolved to the point at which lower leg amputees may be competitive with the best runners in the world [6], there is still a gap between upper extremity prosthetics and real hands [7]. The human hand is a complex organ with several purposes, such as handling and manipulating delicate objects. It is considered the most relevant tactile organ that connects the body with the environment, being highly pertinent for both verbal and nonverbal communication [8]. Given the complex interface of the human hand in performing several tasks in the everyday life, such as grasping objects, playing musical instruments and communicating, studying it has been a source of fascination for philosophers, engineers, sports scientists, and many others. As stated by Aristotle, “nature has admirably contrived the actual shape of the hand so as to fit in with this arrangement (...) for taking hold of things and for exerting pressure” [9].

Although some few companies, like Shadow Robotics, have been designing and manufacturing anthropomorphic robot hands, the cost has made their widespread distribution negligible the state-of-the-art Shadow Robotics Shadow Dexterous Hand (full setup) costs around £100K. Even though some lower cost solutions have been made available in the market over the past years, most are still heavy, short in battery life, and lack the necessary dexterity. This paper presents the development of a low-cost multipurpose robotic hand for research and academia.

The robotic hand was fully integrated on Robot Operating System (ROS)¹ and designed to offer a high degree of functionality and control possibilities for electrical engineering students (e.g. closed-loop force control), computer science (e.g. learning-by-imitation with computer vision methods), biomedical engineering (e.g. myoelectric control), and others. The full development of the robotic hand also encompasses a framework, developed as an interactive application in MatLab, designed to introduce the topic of sequence data classification in academia. The framework allows to control the robotic hand using electromyography (EMG) to detect and identify muscle activity patterns of the forearm. This paper expects to contribute to an everincreasing human-robot symbiosis by motivating students to engage in transhumanism/robotics and machine learning studies using more sophisticated technologies and methods.

II. ROBOTIC HAND FOR ACADEMIA

This section describes the development of a low-cost robotic hand (Fig. 1), collaboratively developed by a private company, Ingeniarius2 Ltd., and Aston University (UK). The design of the anthropomorphic hand followed a biomechatronic approach, similarly to Massa et al. [10] and Zollo et al. [11]. Nevertheless, to maintain the cost as low as possible for subsequent distribution as an open-source platform, the solution was built based on the InMoov3 3D printed hand, with one controllable degree of freedom per finger, with additional force feedback sensing in each fingertip. Put it differently, as the hand was conceived for educational and research purposes, the mechanical design and the anthropomorphism of the hand were harmonized with the design of a simplistic control system to keep the solution affordable.

As an educational solution, and considering the insights provided in the previous section, the robotic hand is fully integrated on ROS. ROS Kinetic runs in an internal Raspberry Pi 34, which exchanges messages with a Pololu's Mini Maestro 18-Channel Servo Controller⁵ by serial communication. The serial communication protocol between ROS and the low-level control was developed as a ROS driver which publishes and subscribes the topics presented in Table I.

III. EMG-CONTROL FRAMEWORK

Engaging engineering students in a topic as broad, and rather theoretical, as transhumanism requires a practical approach. This work benefits from the premise that robotics can be used to overcome patient's impairments and motor disabilities, namely through the adoption of prosthetic limbs. The work resorts to an Arduino-based low-cost hardware to extract the electrical activity produced by forearm muscles, which is then recognized, through the LSTM sequence data classifier, triggering a pre-defined sequence or action reproduced by the robotic hand through the `/fingers_action` ROS topic. Fig. 2 presents a general overview of the herein described framework. Next sections further describe the proposed framework.

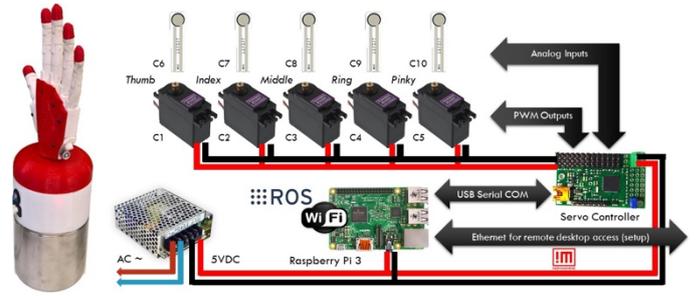


Fig. 1. In-house robotic hand electromechanical design.

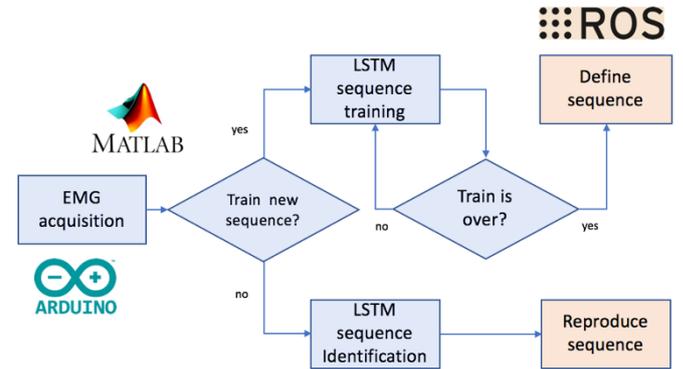


Fig. 2. General overview of the EMG-control framework architecture.

TABLE I.
TOPICS PUBLISHED AND SUBSCRIBED BY THE ROBOTIC HAND ROS DRIVER NODE.

Msg. name	Msg. Type	Description
<code>/fingers_setpos</code>	<code>std_msgs::UInt16MultiArray</code>	Subscribe desired position of each of the five fingers (degrees)
<code>/fingers_setvel</code>	<code>std_msgs::UInt16MultiArray</code>	Subscribe desired velocity of each of the five fingers (degrees/second)
<code>/fingers_getforce</code>	<code>std_msgs::UInt16MultiArray</code>	Publish force feedback of each of the five fingers (mN)
<code>/fingers_getpos</code>	<code>std_msgs::UInt16MultiArray</code>	Publish the current position of each of the five fingers (degrees)

A. Setup

To maintain the low-cost strategy of the solution, the EMG setup comprises an Arduino Uno controller board and an Advancer Technologies MyoWare muscle sensor (Figure 3). Despite the limitations of the Arduino, especially the time needed to read values from analog pins and to write to serial port, the firmware was prepared to acquire EMG data at 1000Hz; a frequency often adopted in many other works revolving around EMG analysis [12]. The Arduino Uno is plugged to a computer running MatLab R2017b, which receives the data and pre-processes it in order to feed the LSTM network, be it for training or testing. A MatLab application, described in the next section, then manages the acquired data according to the user preferences.

¹<http://www.ros.org/>

²<http://ingeniarius.pt/>

³<http://inmoov.fr/>

⁴<https://www.pololu.com/docs/0J40/all>

⁵<http://www.advancertechnologies.com/p/myoware.html>

B. MatLab Interface

Nowadays, most roboticists consider ROS as the de facto standard, being an open-source meta-operating system for robots, providing hardware abstraction, low-level device control, message-passing between processes, and package management [13]. ROS popularity among the robotic community led to a strong cross-compatibility with many third-party software. While most tools, such as javascript clients and other C/C++ applications, can communicate with ROS through JSON-based commands using `roslaunch`, MatLab can inherently create ROS nodes and exchange messages with other nodes on the ROS network. MatLab's Robotics System Toolbox enables to communicate with a ROS network, interactively exploring robot capabilities and visualizing sensor data. The toolbox allows to test and verify robotic algorithms on ROS-enabled solutions, combining MatLab features, namely its wide range of available machine learning techniques, with the many robotic approaches developed by the ROS community. This cross-compatibility with ROS along the recent implementations of deep learning approaches in MatLab made it the chosen framework for the core implementation for the EMG-control of the robotic hand. MatLab has tools and functions designed to facilitate transfer learning, can use available graphics processing units (GPUs) to speed up the training deep learning process, and includes multiple deep neural network models, such as AlexNet, VGG-16, VGG-19, and Caffe models. With the introduction of a new environment for building apps in MatLab R2016a, the App Designer, it became easy to develop professional looking and intuitive interfaces, as the one presented in Fig. 4.

The app easily allows to create new actions, or classes, associate EMG data for deep learning LSTM network training, and classify new sequence of data to control the robotic hand. It is noteworthy that the output defining the action is based on the class scores, between 0 and 1 as a probability for each class, illustrated at the bottom of Fig. 4 as a bar graph. Next section describes the implementation of the LSTM network behind this output.

C. LSTM network

Human movement studies have been motivated by a wide range of applications in different fields [15]. Contrary to the majority of the presented work, which reside on inertial measurement sensing [16] or cameras [17], this work relies entirely on EMG sequence data classification for action recognition. As previously stated, the aforementioned framework receives the EMG raw data through serial at 1000Hz. Even considering short hand actions (e.g. opening/closing hand, waving, etc.), this would correspond to about 2 seconds of a one dimension 8-bit EMG signal, i.e. a sequence with a length up to 2000, 8-bit values. The aim is to classify new incoming data in real-time, regardless of having a GPU or not. The sampling rate was downsampled to 200Hz by applying a Chebyshev Type I infinite impulse response low-pass filter of order 8 [18]. This is a realistic down sampling as the dominant energy is concentrated in the range of 20 – 500Hz, even though the frequency range of EMG is within 0 – 1000Hz [19].

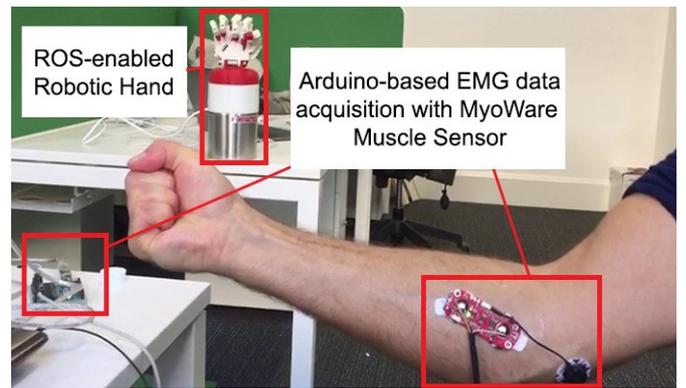


Fig. 3. Setup comprising *Arduino Uno* and *MyoWare* for EMG data collection, which communicates with a computer running MatLab for sequence data classification, which then publishes the recognized action under the `/fingers_action` ROS topic, thus enabling the robotic hand to reproduce the action.

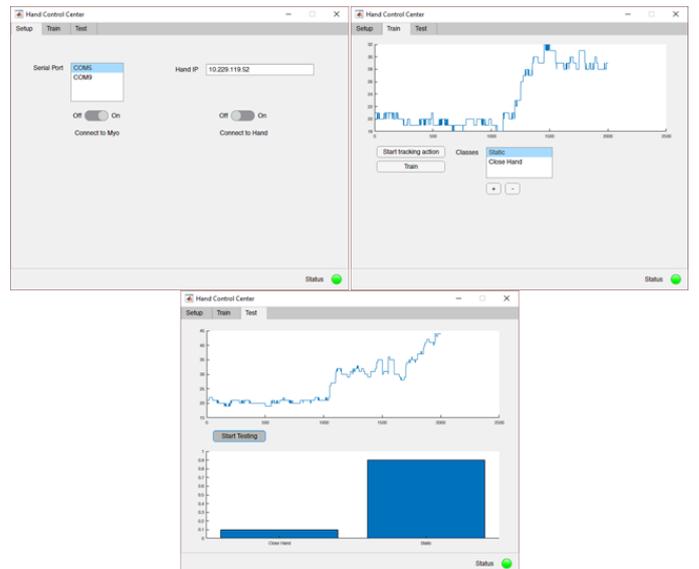


Fig. 4. MatLab app to train new hand sequences/actions and control the robotic hand. Top left) first menu to connect to the Arduino board (EMG setup) and to the ROS master (robotic hand); Top right) second menu to create new actions (classes), track EMG data related to each of these actions, and train the LSTM network; Bottom center) third menu to test new incoming EMG data and control the robotic hand.

Introduced in 1997 by Hochreiter and Schmidhuber [20], Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) architecture including a gradient-based learning algorithm. Intended to overcome the error block-flow problems brought up by conventional Back-Propagation Through Time (BPTT) and Real Time Recurrent Learning (RTRL), LSTM solves the long-term dependency problem faced by RNN. In order to allow the recurrent nets to proceed in learning over many time steps (e.g. greater than 1000), LSTM enforces a constant error flow through the internal states of the special units. Moreover, LSTM incorporates gated cells, which contain information outside the normal flow of the recurrent network allowing it to act on the signals they receive. Those gated cells act as analog cells, implemented with element-wise multiplication by sigmoids, and are all in the range of 0-1,

where 0 represents blocking of information and 1 the passing of all content depending on the data's strength and importance. The data are filtered with their own sets of weights, which are modified through the recurrent networks learning process [21].

LSTM have a chain structure of repeating modules with four interacting neural network layers composed of three sigmoid (σ) having a gate activation function and another layer having an input/output activation function (usually \tanh). The cell state (Ct) in each repeating module can have information added or removed via the gates before passing to the next LSTM block. Furthermore, each block goes over several steps before transferring the updated cell state (Ct) and hidden state (ht) to the next repeating unit and similarly receives the previous Ct-1 and ht-1 from the previous block as well as the input xt. First, the forget gate f at time t decides what information to remove from the cell state:

$$f_t = \sigma(W_{f_-} \cdot [h_{t-1}, x_t] + b_f), \quad (1)$$

where W is the weight matrix (in this case of the forget gate f) and b is the bias vector. Second, the decision for which information to store in the cell state is made using the input gate layer i at time t , in combination with the \tanh layer, which creates a vector of the new candidate values, C_t , that have the potential to be added to the cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \quad (3)$$

Now that the decisions on how to modify the cell state were made, it should be updated as:

$$\tilde{C}_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \quad (4)$$

Finally, the output gate o at time t generates the output and consequently \tanh , acting as the output activation function updates the hidden state h :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t * \tanh(\tilde{C}_t). \quad (6)$$

With the LSTM architecture introduced, it is important to explore the computation for parameter update of the weights and biases in the network which was done using the gradient descent algorithm optimizing (minimizing) the loss function. The mode of gradient descent chosen for this application was mini-batch, which calculates the model error and updates its coefficients by dividing the training dataset into small batches. Setting up the batch size B as the training set size can lead to a high computational complexity, taking too long for the gradient to converge.

On the other hand, a small batch size of $B = 1$ would result in a stochastic gradient descent which, on the other hand, generates a lot of noise and circles around the minimum without converging to it. For these reasons, combining those two opposite effects results in a U-curve with the most suitable values for B being at the intermediate [22]. This achieves faster

learning while being consistent in moving towards the minimum. Furthermore, one should note that even though the true gradient direction, calculated by averaging over the whole training set, is locally the steepest descent direction, it may not lead to the right direction when coping with larger steps. In addition, having small values of B may cause better observed results since it may allow further exploration in parameter space and a form of regularization both due to the noise injected in the gradient estimator [22].

In this application, as all sequences have similar sizes (short hand actions of about 2 seconds), the smallest training sequence was calculated and was considered as the mini-batch size. In datasets where the sequences might significantly vary in size, an alternative would be to choose a mini-batch size in such a way that it would allow to divide the training data sequence lengths evenly so as to reduce the amount of padding in the mini-batches. Moreover, for computational efficiency purposes, the batch size could be rounded to the power of two to fit CPU and GPU memory.

As any neural network-based approach, the number of hidden layers is another key feature of the LSTM. Even though there are empirically-derived rules-of-thumb presented in the literature, the "optimal" size of hidden layers of the LSTM do not follow most traditional approaches. In many applications, the most common rule-of-thumb defines the optimal size of the hidden layer between the size of the input layers and the size of the output layers. The output dimension in this application was considered to be 256, which is the size of all possibilities in terms of the next value of the time series (8 bits).

Additionally, choosing the maximum number of epochs, which is the number of passes of the full training set, is a critical step. A too small number of epochs may not be sufficient for the networks to properly learn the data; however, a large number may result in over-fitting. The maximum number of epochs chosen in this application was 200. To test the validity of the chosen number, one can deduce the line plot of the series of root-mean-square error (RMSE) scores on the train and test sets for different training epochs. A downward direction in RMSE over the training epochs for almost all of the experimental runs is a good sign; in contrast, a rising error trend is a sign of overfitting.

IV. PRELIMINARY EXPERIMENTS AND ASSESMENT

A. Experiments using MyoWare Raw Data

In order to validate our proposed robotic hand and the EMG-based architecture to control it, we have conducted a preliminary experiment to classify two simple hand gestures such as opening and closing the hand. Given our architecture consisting of only one *MyoWare* sensor (EMG), we were limited to detect only these two gestures, excluding fine movements (i.e. individual finger motion). Nevertheless, in order to show the potential of our framework, we kept our preliminary tests, when it comes to

classification, simple and easy. Using our designed intuitive Matlab interface for classification and control of the low-cost hand, we have asked six individuals, aged between 30 and 45 years old, five males and one female to use and play with it. One male individual has acquired data to train both gestures (open and close), by performing 50 trials, following the down sampled data (200Hz), as explained in the previous section, for two seconds each. Our strategy for classification is defined by using a time-window of half second, so that for each repetition lasting two seconds we will have four local batches of data for classification.

For off-line tests using our collected dataset, we have used the strategy of leave-one-out cross-validation to verify the classification performance for two classes: opening and closing the hand. The overall accuracy attained was 67%. This result is due to the use of a single EMG sensor on the forearm. For on-the-fly tests, and with the intention to demonstrate the generalization capacity of the LSTM for gesture recognition, five individuals tested the gesture recognition using the same training data (i.e. from only one individual). The on-the-fly tests consisted of each individual (excluding the one that trained the network) opening and closing the hand within intervals of two seconds, thirty times each individual. The average accuracy for the five individuals during the thirty tests were 53%, 61%, 63%, 67% and 71% with an overall accuracy of 63%. Given that we trained the network using data from only one individual (performing 50 times both gestures), we can state that the LSTM has potential for generalization to classify the normalized sequential data. Fig. 5 presents the overall accuracy for the on-the-fly tests performed by the five individuals. Despite of our results are too preliminary for the robotic hand control, we can state that the control based on primitive actions defined for specific gestures and individual finger control are working very well in terms of actuators performance, ROS implementation and Matlab interface, which is useful for academia.

B. Experiments using Myo Gesture Control Armband

Our proposed initial low-cost setup is very interesting; however, it is very limited when it comes to the sensor to measure the muscles activities. It allowed us to recognize only two classes, opening and closing the hand. Thus, we decide to extend our setup by using the Myo Gesture Control Armband by Thalmic Labs instead, which is still a low-cost sensor, however more sophisticated, which eight channels to measure the forearm muscle activities as shown in Fig. 6. This sensor allows to recognize a variety of gestures due to the number of sensors and also because it returns accelerometer and orientation data, since it has an inertial measurement unit attached to it as well. This sensor works at 200Hz per second. In this specific case, with eight channels, our classification strategy is different than the previous one. We have to extract discriminative features instead of working with the raw data. This strategy is based on detecting the features in each individual signal and later on concatenate all features form all eight signals into a matrix, where the columns are the features types and the rows the frames.

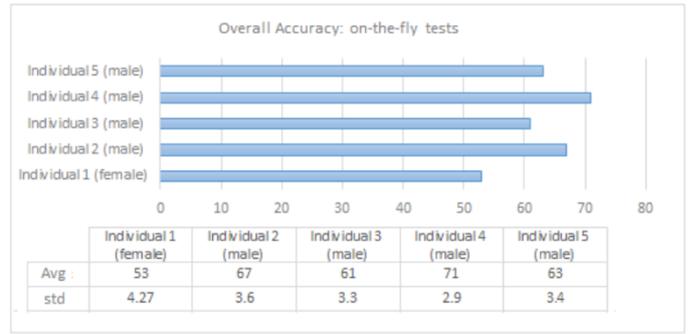


Fig. 5. Overall results (on-the-fly) in terms of accuracy after 30 experiments.

It is worthy to mention that our Matlab interface communicates with the Myo Armband sensor can easily make the data acquisition and classification. We have created a dataset with six individuals performing three gestures (opening and closing the hand, and random motions including the previous two) repeating them 10 times for 5 seconds each session, so that each individual performed 50 seconds for each gesture. The idea of adding a random gesture is to check if the network can discriminate well the classes even when some part of the sequence of the gesture coincide to other classes. Then, the next challenge is the features extraction. EMG signals are complex and non-linear. The best practice is to apply short-time windowing technique. We have considered a set of features that adequately discriminate different classes of EMG signals. These features rely on statistical techniques (e.g. mean, standard deviation, autocorrelation of the signals, statistical moments of 3rd and 4th order to get the skewness to measure the asymmetry of the data, and also the kurtosis to measure the peakedness of the probability distribution of the data), time-frequency based on fast Fourier transform (FFT), Shannon entropy, log-energy entropy, max-min features, log-covariance after splitting the data into batches and using the 1D Euclidean distances of the mean values, max-min features from multiples batches to compute the covariance matrix.

We also computed the derivatives of all features using the previous time window. All features are computed in terms of the temporal distribution of the signal in a second for each time window. An overlap of 0.5 second is used when moving the window, i.e. the temporal window 1 (w1) starts at 0 sec. and finishes at 1 sec.; w2 starts at 1.5 sec. and finishes at 2.5 sec.; w3 starts at 2 sec. and finishes at 3 sec.; w4 starts at 2.5 sec. and finishes at 3.5 sec., and so on. Details about these feature models can be found in [14] and [15], where similar features were used to classify EEG data of mental states and multiple IMU data of daily activities, respectively. After the features extraction, the data was normalized using the max-min, individually for each type of feature. Once the set of features was computed and normalized, we have used them as input to train and test the LSTM network. Likewise, leave-one-out cross validation tests were done. The overall result for this preliminary test is presented in a confusion matrix in Fig. 7, attaining an overall accuracy of 81.92%.



Fig. 6. Myo armband sensor by Thalmic labs. The bottom images present the possibilities for gestures recognition when using this sensor.

Overall Accuracy (Individual 1...6)		Accuracy		
		Opening	Closing	Random
Results	Opening	83.7	10.15	6.15
	Closing	9.73	82.16	8.11
	Random	9.33	10.76	79.91

Fig.7. Overall accuracy: leave-one-out cross validation tests. Data acquired using a Myo armband; features extracted from eight signals to serve as input for the LSTM network.

We can state that by extracting discriminative features using a time window from the different signals of this sensor is crucial to attain a good performance. This setup is better than the previous one due to the sensor being more robust.

V. CONCLUSION

This paper presented the development of a robotic hand for academia, focusing on the electromechanical design and ROS integration. This was followed by the description of an educational classification framework developed on MatLab, thus allowing students to control the robotic hand through the electrical activity produced by the forearm muscles. The proposed framework intends to foster students' engagement in transhumanist technologies, namely in human-robot "symbiosis" through robotic prosthetics, by easily guiding them through the process of training and testing a LSTM sequence data classification architecture. Preliminary results using two setups show that our proposed framework has potential for generalization given the amount of data used when relying only on an EMG sensor with a single channel (overall accuracy up to 67% with offline tests and 63% on-the-fly) and also using a more sophisticated sensor with eight channels such as Myo Armband (overall accuracy around 82%), when extracting a set of discriminative features within a short-time windowing process. As future directions, other existing gestures will be

integrated within the same framework, also providing a higher degree of gesture recognition. Moreover, we intend to add other classical classification models and ensembles within the MatLab framework as an alternative to deep learning.

REFERENCES

- [1] M. More and N. Vita-More, *The Transhumanist Reader: Classical and Contemporary Essays on the Science, Technology, and Philosophy of the Human Future*. John Wiley & Sons, 2013.
- [2] S. Tachi, *Telexistence*. Cham: Springer Int. Publishing, 2015, pp. 229–259.
- [3] J.L. Pons, *Wearable Robots: Biomechatronic Exoskeletons*. J.W. & S, 2008.
- [4] R. Bogue, "Exoskeletons and robotic prosthetics: a review of recent developments," *Industrial Robot*, vol. 36, no. 5, pp. 421–427, 2009.
- [5] Y. Bar-Cohen, Ed., *Smart Structures and Materials 2002: Electroactive Polymer Actuators and Devices (EAPAD)*, vol. 4695, 2002.
- [6] P. G. Weyand, M. W. Bundle, C. P. McGowan, A. Grabowski, M. B. Brown, R. Kram, and H. Herr, "The fastest runner on artificial legs: different limbs, similar function?" *J. of App Phys*, pp. 903–911, 2009.
- [7] D. A. Zlotolow and S. H. Kozin, "Advances in upper extremity prosthetics," *Hand clinics*, vol. 28, no. 4, pp. 587–593, 2012.
- [8] R. M. Krauss, Y. Chen, and P. Chawla, "Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us?" *Advances in experimental social psychology*, vol. 28, pp. 389–450, 1996.
- [9] S. T. Venkataraman and T. Iberall, *Dextrous robot hands*. Springer Science & Business Media, 2012.
- [10] B. Massa, S. Roccella, M. C. Carrozza, and P. Dario, "Design and development of an underactuated prosthetic hand," in *Robotics and Automation, 2002. ICRA'02*, pp. 3374–3379.
- [11] L. Zollo, S. Roccella, E. Guglielmelli, M. C. Carrozza, and P. Dario, "Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 4, pp. 418–429, 2007.
- [12] K. A. Farry, I. D. Walker, and R. G. Baraniuk, "Myoelectric teleoperation of a complex robotic hand," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 775–788, 1996.
- [13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009.
- [14] J. Bird, Luis J. Manso, E. Ekart, E. P. Ribeiro, D. R. Faria, "A Study on EEG-based Mental States Classification for Human-Robot Interaction" *The 9th IEEE International Conference on Intelligent Systems*, 2018.
- [15] J. P. Vital, D. R. Faria, G. Dias, M. S. Couceiro, F. Coutinho, and N. M. Ferreira, "Combining discriminative spatiotemporal features for daily life activity recognition using wearable motion sensing suit," *Pattern Analysis and Applications*, vol. 20, no. 4, pp. 1179–1194, 2017.
- [16] D. R. Faria and J. Dias, "3d hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach," in *Intelligent Robots and Systems, 2009*, pp. 1284–1289.
- [17] D. R. Faria, C. Premebida, and U. Nunes, "A probabilistic approach for human everyday activities recognition using body motion from rgb-d images," *2014 RO-MAN*, pp. 732–737.
- [18] I. Acoustics and S. P. S. D. S. P. Committee, *Programs for digital signal processing*. IEEE, 1979.
- [19] J.-U. Chu, I. Moon, and M.-S. Mun, "A real-time emg pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand," *IEEE Transactions on biomedical engineering*, vol. 53, no. 11, pp. 2232–2239, 2006.
- [20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation* 9, vol. 8, pp. 1735–1780, 1997.
- [21] D. D. Team. (2017) *Deeplearning4j: Open-source distributed deep learning for the jvm*, apache software foundation license 2.0. [Online] Available: <http://deeplearning4j.org>.
- [22] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *CoRR*, vol. abs/1206.5533, 2012.