Journal of COMPUTER AND FORENSIC SCIENCES

CFS 2024, Vol. 3, Issue 2, pp. 3–27
https://doi.org/10.5937/jcfs3-54725
Original research paper

Surface Electromyography-based Gesture Recognition for Robot Hand Control

Srđan Savić[1]*, Dunja Pavlović[2], and Andrej Čilag[1]

[1] Faculty of Technical Sciences, University of Novi Sad, Serbia; savics@uns.ac.rs, andrej.chilag@uns.ac.rs

[2] No affiliation; dunjapav99@gmail.com

* Corresponding author: savics@uns.ac.rs

Received: November 12, 2024 • Accepted: December 24, 2024 • Published: December 30, 2024

Abstract: This paper presents a human-machine interaction interface for the teleoperated control of a three-finger robot hand based on surface electromyography. Based on the recorded sEMG signals, motion intention of a human operator is recognized and mapped onto corresponding predefined robot hand grasps. First, the experimental setup and underlying methodology of dataset generation are presented. Namely, 4-channel EMG data were collected from the forearm muscles of 4 healthy subjects as they performed a sequence of predefined grasps. Data processing included data filtering and segmentation, feature extraction both in time and frequency domains, data annotation, and gestures classification. Two types of classifiers (with multiple variations of parameters) were implemented and compared: (i) Support Vector Machines with multiple different kernels and (ii) K-Nearest-Neighbors. The k-cross validation procedure was combined with the holdout method to generate training, validation, and test sets. The performance of each classifier is presented in the form of a confusion matrix and a set of statistical measures, including F-measure. Finally, the integration of the proposed sEMG-based HMI interface and the selected classifier with a robot hand motion controller is presented.

Keywords: surface electromyography; motion intention recognition; multiclass classification; robot hand control; EMG dataset generation.

1. INTRODUCTION

The application of robotic systems in human-centered, unstructured, dynamic environments and their cooperation and collaboration with humans within a shared space impose multiple challenges in the wide spectrum from the physical realization of the movement and robot motion control to semantic understanding of current situational context and decision-making. More precisely, considering the class of various robotic manipulators, manipulation tasks may include a vast amount of objects with various sizes, shapes, textures, rigidities, and functions. Additionally, the particular situational context may significantly influence and alter the manipulation task realization, requiring different parameterizations of a single task in different contexts. E.g., human grabbing a cup from a

table would consider different types of grasp and potential contact points, depending on whether the cup is empty or full, whether the liquid is hot or cold, etc. Such an advanced level of decision-making and planning requires structured domain-related knowledge, reasoning capabilities, and full semantic understanding of situational context, which are still far beyond current state-of-the-art robotic systems. At best, it is possible to develop robotic systems that exhibit intelligent-like behavior with some of the mentioned capabilities for narrow, restricted application domains and interaction scenarios, but it is hard to generalize such models to arbitrary domains. Therefore, for demanding applications requiring real-time and complex decision-making, it is worth including human-in-the-loop control. Robot teleoperation, i.e., human-operated remote robot control, provides synergy of human cognitive capabilities and robot manipulation capabilities (repeatability, precision, operation in harsh environments, etc.).

Teleoperation requires an appropriate human-machine interface (abbr. HMI), which enables a mapping of input human-defined motion commands to corresponding reference motion trajectories of a remote robotic system. Typical HMI interfaces for robot hand teleoperation include various types of joysticks with multiple degrees of freedom (abbr. DOFs) or VR controllers [1], exoskeletons [2], vision-based systems [3, 4], data gloves [5–7], or sEMG-based hand gesture recognition [8, 9].

Each type of HMI interface has its pros and cons. E.g., exoskeleton-based haptic devices usually require a certain amount of training for the operator to gain expertise in robot control and constrain natural human movement to a certain extent due to a bulky design [10]. A vision-based approach to gesture recognition, although efficient, quite robust, and well studied, may suffer from inappropriate lighting conditions, complex backgrounds, and occlusions caused by obstacles in unstructured environments [11]. Also, this approach is limited to kinematic analysis and lacks haptic information about grasping and contact force. Haptic data gloves enable pose estimation of the human hand and even may provide haptic (force) feedback to the human operator. However, data gloves, depending on the design, may be bulky and provide certain discomfort to the operator, affecting their user experience [12]. Also, they may interfere with the tactile sensing capabilities of the operator.

Finally, the HMI control interface based on surface electromyography, used in this paper, provides another alternative for robot teleoperation. It is a non-invasive, wearable, and unobtrusive control interface, so far mainly used for control of myoelectric robotic prostheses for amputees or active orthotic devices for rehabilitation. Surface EMG uses surface electrodes to record stochastic signals that control the muscle movements. These signals are weak and very susceptible to noise from multiple sources and hence require amplification and extensive filtering. However, sEMG signals are easily accessible, have direct correlation with movement intentions and minimal delay, and also enable direct mapping between muscle activation magnitude and robot grasping force [13]. This interface is typically placed on the forearm, thus not interfering with fine hand manipulations. The HMI interface should enable operators, even without particular technical knowledge in the robotics domain, to teleoperate a robot in an intuitive and natural manner, thus minimizing the required cognitive capacities of a human operator. Exploiting the natural movement of human hand thus represents a natural choice.

Teleoperated robot hand control includes two types of tasks: (i) hand pose estimation for continuous teleoperation and (ii) gesture (motion intention) recognition for mapping of a discrete human hand pose onto the predefined finite set of possible robot hand motions. Hand pose estimation usually requires an appropriate motion retargeting strategy, i.e., mapping between two different kinematic structures (human and robot hand), which may be performed at the level of fingertips, joints, poses, etc. [14]. However, gesture recognition task is usually based on an end-to-end learning approach that maps any input state onto the corresponding output class. The systematic survey of various approaches to "human hand to robot hand" motion mapping is given in [15]. Gesture recognition may be static or dynamic. Static gesture recognition, addressed in this paper, maps a discrete feature vector onto the set of output labels, representing possible robot grasps. It can be modeled as a multi-class classification problem and treated with some of the machine learning (abbr. ML) approaches. Dynamic gesture recognition is based on a classification of a sequence, i.e., a temporal pattern of feature vectors.

2. RELATED WORK

Early works on sEMG-based gesture recognition were mostly based on well-established machine learning techniques for classification, like Support Vector Machines (abbr. SVM), Linear Discriminant Analysis (abbr. LDA), and K-Nearest Neighbors (abbr. KNN). In [16], an SVM multi-class classifier was used to control the opening/closing of the DLR four-finger hand II, while in [8], SVM has also been used for teleoperation of the robot hand/arm system comprising of the 7 DOFs DLR LWR III robot arm and the five-finger anthropomorphic DLR-HI hand II. In this case, SVM was used to map muscle activity to hand position/orientation in Cartesian space and grasping force. Furthermore, the proposed setup did not require precise positioning of the electrodes. In [17], a 2-channel sEMG was used in combination with a KNN classifier to control a custom-designed bionic hand. The system was trained to distinguish 4 different gestures. In [18], an adaptive KNN algorithm was proposed to tackle the problem of model generalization across novel users due to the variability of sEMG signals among different individuals. Two systematic surveys and comparative analysis of various machine learning techniques for the classification of hand gestures based on sEMG signals are given in [19, 20].

In the case of multichannel sEMG, a feature vector may become high-dimensional, and thus dimensionality reduction techniques are often applied prior to the main classifier. In [21], different combinations of dimensionality reduction and classification approaches were systematically compared. The proposed schemas combined Principal Component Analysis (abbr. PCA) and Orthogonal Fuzzy Neighborhood Discriminant Analysis (abbr. OFNDA) with SVM and LDA classifiers.

Another approach to gesture recognition, particularly for dynamic gestures, utilizes probabilistic graphical models (abbr. PGM), like hidden Markov models, combined with SVM [22] or Gaussian mixture models [23]. The performance of classical ML and PGM-based approaches depends heavily on feature extraction and the design process, i.e., on the skill of the designer. Therefore, contemporary approaches are dominantly based on deep learning techniques, which have the advantage of automatic feature extraction. In [24], the

authors used a convolutional recurrent neural network implemented on an embedded edge AI platform for robot manipulator control. In [25], a residual model and a variant ConvLSTM model were combined into one multi-stream network, which was used to recognize six different dynamic gestures. Finally, in [26], a self-attention-based graph neural network was used for both static and dynamic gesture recognition on a novel sensor, consisting of an array of bipolar stretchable sEMG electrodes. The proposed architecture is very advanced, and it is able to learn online from a single trial per gesture, in contrast to usual data-hungry deep learning models. It is also capable of capturing spatial relationships between sEMG channels [26].

3. MAIN IDEA AND OUTLINE

The contribution of the paper is threefold. Namely, we have: (i) conducted an experimental study and generated a dataset of sEMG signals, (ii) performed systematic comparative analysis of two typically used types of classifiers (SVM and KNN), and (iii) described implementation details of integration with real hardware and performed validation using our custom-designed three-finger robot hand.

Although many papers have been written regarding the EMG-based gesture recognition, we find our work justified for two main reasons. First, the choice of input hand gestures is heavily dependent on the kinematic structure and capabilities of the selected robot hand and the nature of the selected task, i.e., the application domain. Hence, in order to develop a sEMG-based interface for our own custom-designed robotic hand, we generated our original dataset for classifier training. Second, as stated in [19], the majority of studies in this field tend to use the model accuracy as the only benchmark, instead of more general performance metrics such as the F measure. Therefore, in order to provide a methodologically plausible comparison of different classifiers, we have applied a broader set of performance measures typically used in machine learning. Although more general and powerful approaches have been recently reported, like [26], they are too complex for the static recognition problem addressed in this paper, and thus the more conventional approach based on classic ML techniques was utilized.

The rest of the paper is organized as follows. Section 4 presents the problem specification and used methodology, i.e., experimental setup and protocol for sEMG signals recording and data acquisition as well as signal filtering. Section 5 describes processing of the recorded signals and their preparation for the classification task, including annotation and outlier removal. In section 6, the feature extraction process is presented. After that, in section 7, modeling, training, and validation of two types of classifiers are described. Comparative results and evaluation metrics are presented in section 8. The integration of the deployed classifier with the robot hand controller is given in section 9, while the final conclusion is provided in section 10.

4. METHODOLOGY – DATA ACQUISITION

4.1. Emg measurement system

the measurement unit consisted of a *SMARTING mBrainTrain* device combined with *SKINTACT FS-RG1/10 ECG* silver-silver chloride low-impedance electrodes. The *SMARTING* device performs monopolar EMG signal measurement using one active electrode placed onto the skin, above the targeted muscle, and one reference electrode, placed at the electrically neutral region. There is also a third electrode, called driven right leg (abbr. DRL), used for noise cancellation.

4.2. Hrc robot hand

this paper also presents an interface for EMG-based control of a particular human-robot collaborative robotic hand (abbr. HRC hand) shown in Fig. 1. Although the proposed methodology and control interface are general enough to be applied to an arbitrary robotic hand, the set of input gestures of human hand was carefully selected to be compliant with the kinematic structure of this particular robotic hand used in experimental validation. It is a high-power robotic hand with 10 degrees of freedom (abbr. DOFs). The hand has 4 active DOFs, actuated with 4 DC motors, and 6 DOFs passively actuated with springs. It also has custom-designed 3-axis force sensors integrated into soft fingertips, combined with torque estimation via motor current measurement, which provide a wide range of applicable forces and enable manipulation with delicate deformable objects.

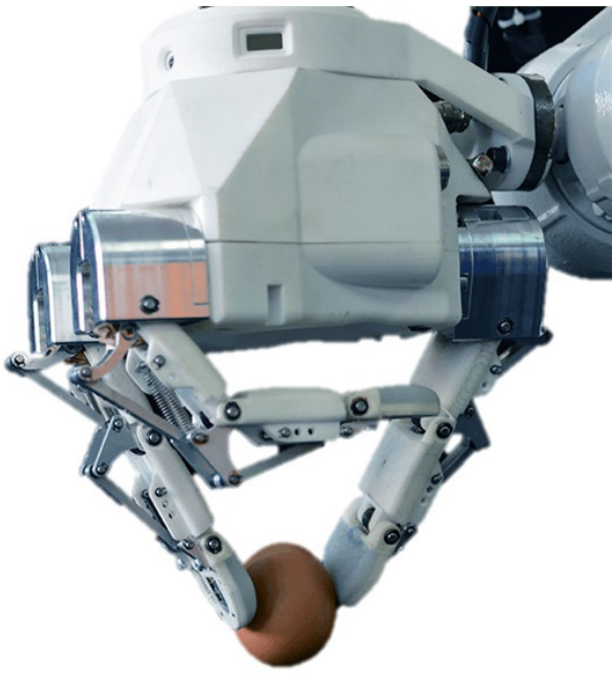

Figure 1. *Functional prototype of HRC robot hand.*

4.3. Input-output mapping and electrodes placement

in order to define the required number of electrodes (channels) and their spatial placement, first we had to define the set of gestures (grips) that we want to record. The number and placement of electrodes was primarily constrained by the hand's kinematic structure (see Fig. 2).

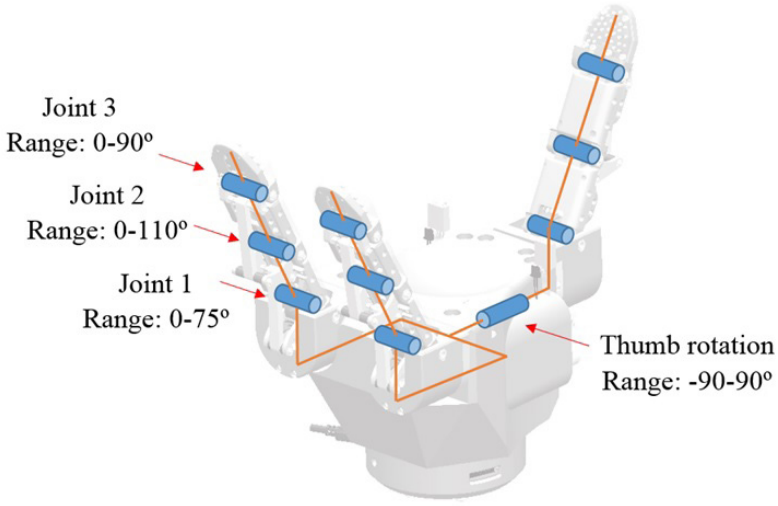

Figure 2. *Kinematic structure of the HRC robot hand.*

Regarding the hand's repertoire of possible motions and distribution of different types of grasp in humans' everyday living activities, we have selected three types of grasp, shown in Fig. 3, as relevant: (i) power grasp, (ii) open hand pinch, and (iii) lateral pinch. We consider (iv) open hand as the fourth default state or the default grasp type. These four gestures constitute the desired set of output gestures that the robot hand should perform. Also, they directly correspond to the set of output labels for a multiclass classifier.

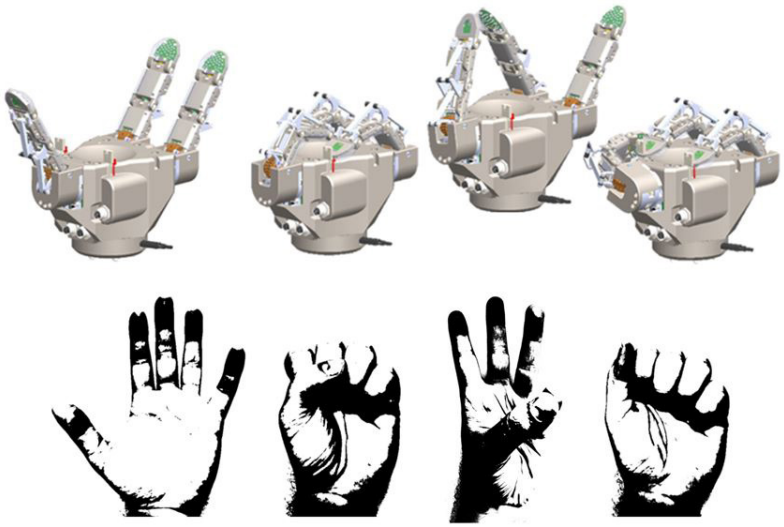

Figure 3. *Predefined input gestures of human hand and corresponding output grasps of robot hand. From left to right: open hand (default state), power grasp, open hand pinch, and lateral pinch.*

It was also necessary to define the set of input gestures that should have been performed by human subjects during the data gathering (recording) process. Since the human hand and the presented three-finger robot hand have different kinematic structures and different numbers of degrees of freedom, it was necessary to account for those particularities and to resolve the mapping between these structures.

Therefore, we analyzed the selected output robot grasps. Namely, the robot hand has only two fingers and a thumb. Two robot fingers may perform only planar motions, i.e., flexion/extension, while the thumb performs both flexion/extension and rotation (opposition). In line with this insight, we imposed a constraint on input gestures to consider joint movement of index and middle fingers as a single input channel and joint movement of ring and little fingers as another one. Therefore, we introduced the following four EMG channels in total to measure muscle activities corresponding to different DOFs of the robot hand:

- Ch_1: flexion/extension of small and ring fingers;
- Ch_2: flexion/extension of index and middle fingers;
- Ch_3: flexion/extension of thumb;
- Ch_4: opposition (rotation) of thumb.

Regarding the open hand pinch grasp, we considered two variations of this input gesture. We compared the raw EMG signals from the four selected channels for the classic pinch (thumb opposing index finger) and alternative pinch (thumb opposing little finger). In the case of the classic pinch, all four channels had almost the same response, while in the case of the alternative pinch, the fourth channel, related to thumb rotation, had significantly higher amplitude. Hence, we decided to use the alternative pinch as the input gesture (Fig. 3, third column) to make it more distinguishable for the classification process.

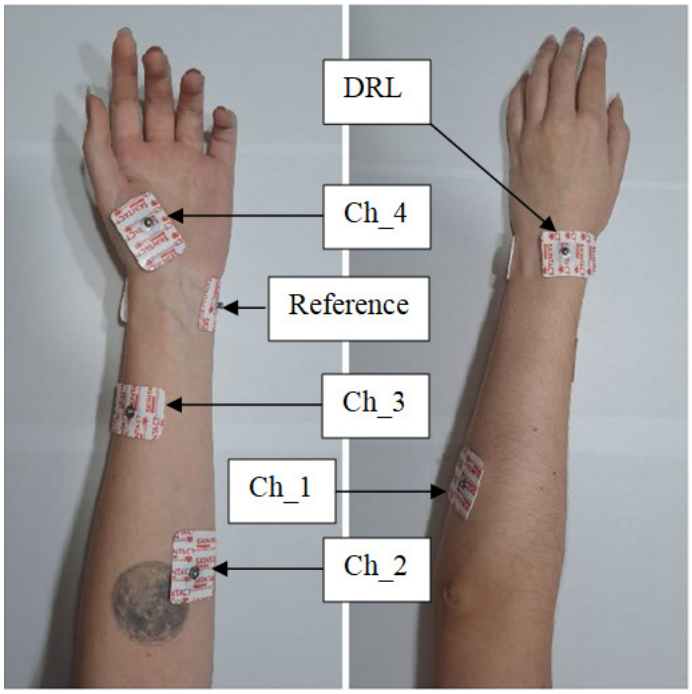

Figure 4. *Electrode placement for EMG signals acquisition.*

Selected primitive movements, measured by four EMG channels, are actuated by the following muscles. Fingers flexion/extension (Ch_1, Ch_2) is performed by muscles m. flexor digitorum profundus and m. flexor digitorum superficialis. Thumb flexion/extension is performed by muscle m. flexor pollicis longus, and thumb opposition by m. opponens pollicis. Placement of electrodes for one of the subjects is given in fig. 4. The positions of electrodes were determined based on the anatomy of muscles and muscular palpation process during the muscle contractions within predefined movements. Electrode placement setup was a bit subject-dependent and varied among the human subjects.

4.4. Experimental Protocol – EMG Signals Recording

Four healthy adult human subjects participated in the signal recording phase. The subjects received instructions via a custom-designed graphic user interface (abbr. GUI), shown in Fig. 5. A single instruction provides the information about which gesture the subject should perform next and for how long it should last. The user interface was designed with the aim of keeping the cognitive capacity of the subject, required for instruction understanding and interpretation, as low as possible. It contains three rectangular fields with the names of three possible input gestures. Each field is associated to a unique color:

- Green – Power grasp;
- Blue – Open hand pinch;
- Red – Lateral pinch;
- White – Open hand (default).

Initially, all the fields are white, and this state corresponds to the default gesture, i.e., open hand. When one of the fields gets colored, it is an instruction for the subject to perform the marked gesture. Also, the duration of the gesture is coded through the duration of the colored state. Additionally, the GUI window graphically presents acquired signals from all four EMG channels in real time. Also, it contains an interface for setting up the connection with the SMARTING device and starting/finishing the recording session.

The instruction set was predefined as a sequence of 500 randomly selected gestures (sampled from the set of 4 available gestures). The distribution of different gestures in the set is approximately uniform. Each of the four participants was asked to perform this same sequence of gestures, presented sequentially, one by one, through a graphical user interface. In order to keep the focus and reduce the number of errors in the subject's performance due to fatigue, the instruction set was divided into 5 subsets, and participants had a 5-minute break between the recording sessions. No subject reported fatigue or cramps during the recording sessions. Each recording session started with a default state, i.e., open hand in rest, in order for the EMG signal to stabilize. Finally, for each of the 4 subjects, we obtained 5 (four-channel) recordings with 100 instructions each. All the sessions were recorded by a video RGB camera. These video recordings were later used in the process of data annotation and outlier removal to provide ground truth signals. All subjects who participated in the experimental study were informed in advance about the experimental protocol and provided written consent for the participation.

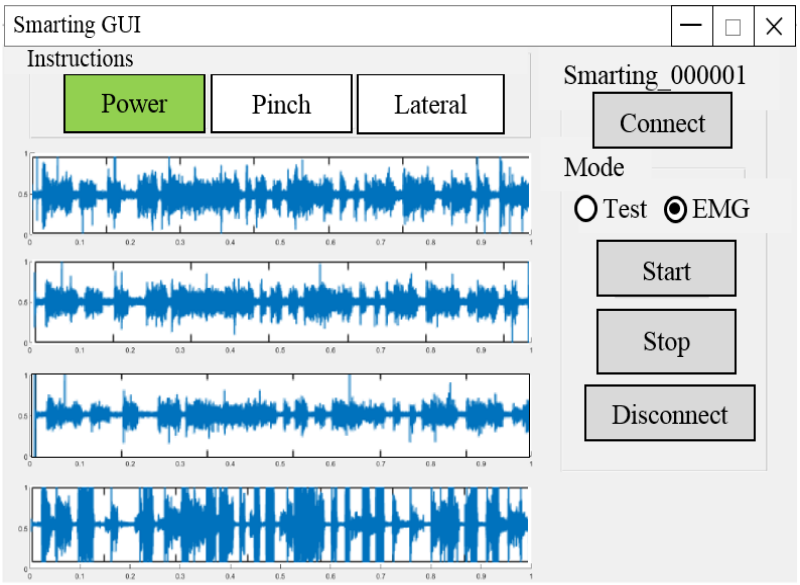

Figure 5. *Illustration of the graphic user interface f or providing instructions to human subjects.*

4.5. EMG Signals Filtering

A band-pass filter was applied to recorded EMG signals to remove the noise. Noise has a low-frequency component caused by motion and a high-frequency component that is not related to muscle activity. The high-frequency component has to be eliminated to prevent the aliasing effect from occurring. Typical values for band limits are 5–10 Hz for the lower limit and 400–450 Hz for the upper limit. It is also necessary to eliminate the signal component at 50 Hz related to network power supply and a direct current (DC) component of the EMG signal to keep the baseline at a zero value.

A band-pass filter was implemented as the first-order Butterworth filter with a (normalized) cutoff frequency of 0.0022. Power supply-related component at 50 Hz was eliminated using a notch (band-stop) filter.

5. EMG SIGNAL PROCESSING

Processing of recorded and filtered EMG signals consisted of four tasks.

5.1. GESTURE EXTRACTION FROM EMG SIGNAL

First, it was necessary to identify time windows in which the recorded gestures occurred. I.e., for each performed gesture within the recorded session, time frames related to the beginning and the end of the gesture had to be detected. We have performed a two-stage semi-automatic procedure for gesture extraction. Namely, in the first phase we have devel-

oped an algorithm for automatic detection of start/end time frames of a gesture, but only for particular limited cases. The algorithm is able to extract the start frame only when the default state (open hand) preceded any other performed gesture. Similarly, it can detect the end frame only in the case when open hand state succeeded the performed gesture.

The algorithm first calculates the root mean square (abbr. RMS) of the original signal and then compares its envelope with a predefined threshold value. RMS provides information about signal power in a certain time period, and it is considered to be the most appropriate representation of the EMG signal amplitude with a waveform suitable for analysis.

The RMS envelope of the EMG signal was calculated using the moving time window according to the following formula:

$$X_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|X_n|^2}, \quad (1)$$

where N is the number of samples within a time window, and X_n is the value of the n-th sample of the EMG signal. Each time window contained N=500 samples, and we used window overlapping of 50%. The obtained RMS envelope is shown in Fig. 6.

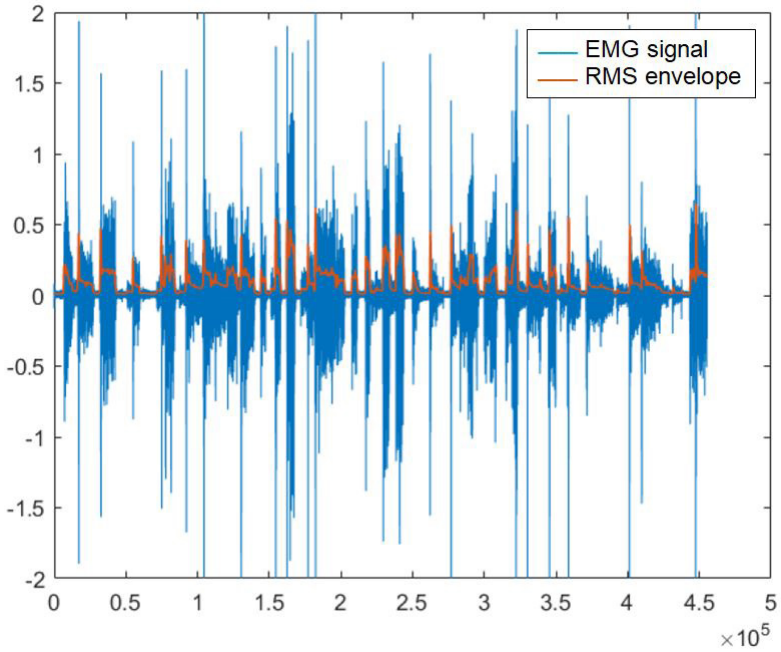

Figure 6. *RMS envelope of EMG signal (subject 1, instruction set 1, Ch_4). Original EMG signal blue line. RMS envelope red line.*

The threshold value was calculated as an average value of RMS envelope amplitudes at points of abrupt transition between the rest state (open hand) and a grip, or vice versa. The calculated value was 0.015 mV. Obviously, in order to determine the threshold for the first time, in this approach, a human expert has to identify the relevant points of signal change. Later this threshold can be used for automatic segmentation. Fig. 7 illustrates the result of the gesture extraction algorithm, which distinguishes three active grips from the default rest state (open hand).

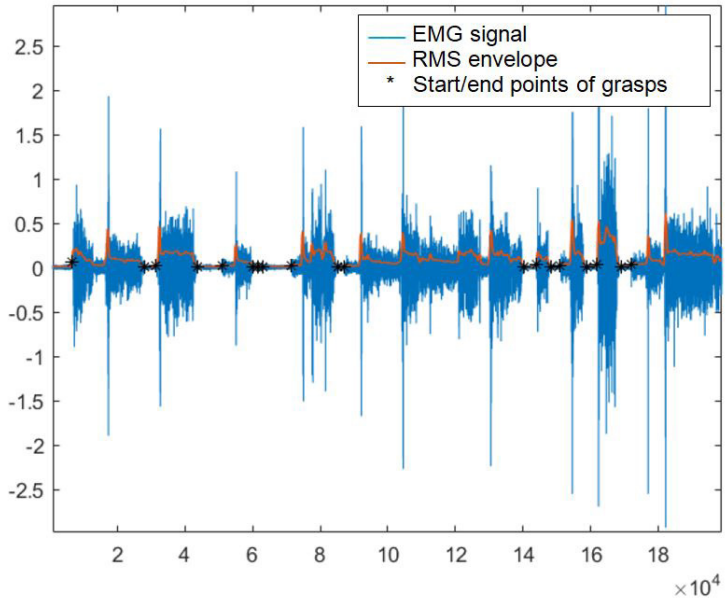

Figure 7. *Illustration of the result of gesture extraction algorithm. Time frames in which the transitions between rest state and other gestures occur are denoted with * symbol (subject 1, instruction set 1, Ch_4).*

At the second phase, these partially segmented signals were further manually analyzed and annotated by hand to determine the remaining time instants of the transitions among the gestures, which could not be detected by the proposed algorithm.

5.2. Data Annotation

The ground truth signal has the same length as the original recorded EMG signal. It is a discrete signal that codes true class labels for each recorded gesture, as well as its duration. It has 4 discrete values that correspond to 4 possible classification labels (4 predefined grasps):

- Class I: power grasp;
- Class II: open hand pinch;
- Class III: lateral pinch;
- Class IV: open hand.

Generation of the ground truth signal actually corresponds to the data annotation (labeling) process. This annotation was performed manually by a human operator, using the following data: (i) the predefined reference set (sequence) of instructions presented to human subjects during recording and time instants when each instruction was actually presented through the GUI, (ii) the time instants of the beginning and end of each performed gesture (instruction) in the recorded EMG signal, and (iii) video recordings of experimental sessions.

By overlapping the reference instruction sequence signals with the recorded and segmented EMG signals, it was possible to assign true class labels to each recorded gesture. Namely, due

to the finite reaction time of human subjects, there is a certain delay in the execution of given instructions. Therefore, the time window of the actual instruction execution in the recorded EMG signal is always shifted and delayed compared to the time window of the corresponding reference instruction. However, these delays were not significant, and they had a small variance, so it was easy for the human annotator to pair the time window of each instruction from the reference signal with the corresponding time window in the recorded signal and to assign the true class. An example of a ground truth signal is shown in Fig. 8.

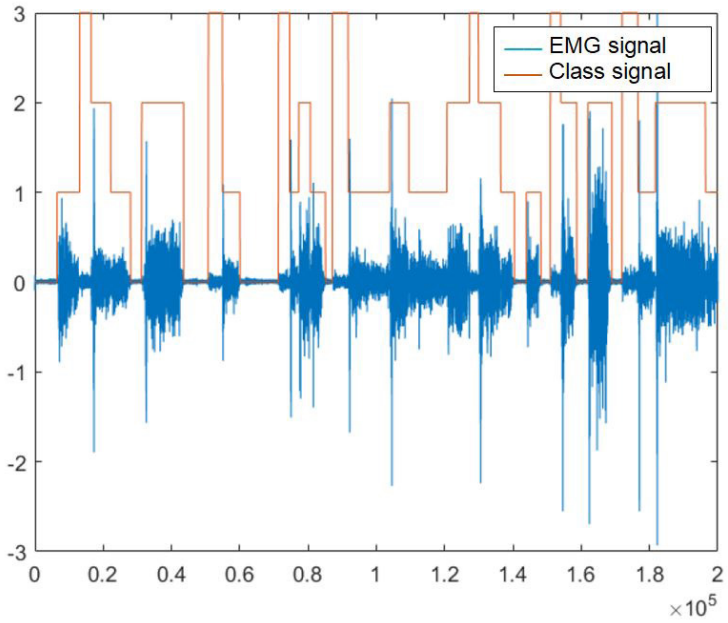

Figure 8. *Ground truth signal with class labels (subject 1, instruction set 1, Ch_4).*

5.3. Outliers Detection and Removal

There have been two sources of outliers in the recorded EMG signals. The first was a possible misinterpretation of the given instruction by the human subject, leading to a realization of a wrong gesture. The second was the occurrence of abrupt changes in the EMG signal amplitude (peaks up to ±120 mV), e.g., due to the accidental cable crossing during the movement execution. An example of such outliers is shown in fig. 9. The outlier removal process was performed manually by a human operator.

The outliers of the first type were removed by comparing the predefined reference instruction set with a video recording of actually performed gestures. Outliers of the second kind were manually removed from the dataset after the visual inspection of the plotted data. Outlier removal assumes removal from all four EMG channels and from the ground truth signal as well, in order to keep the same dimension of recorded and ground truth signals.

Since the number of human subjects that participated in the experimental part of this study is limited, a possible bias in the classifier's training could occur if the recordings related to two mutually exclusive subsets of human participants were used for training and validation. Therefore, in order to avoid such bias, we have randomly mixed the recorded processed data, related to all four human subjects, into one common 5-channel signal (data

set), which can later be divided into subsets for training, validation, and testing. The final signal has 5 channels, including 4 EMG channels and an additional 5th ground truth signal.

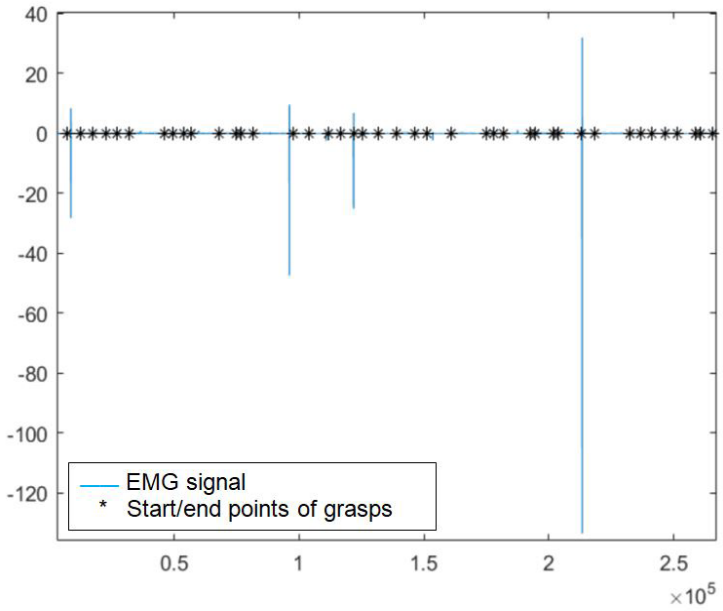

Figure 9. *Illustration of outliers in recorded EMG signals.*

5.4. Signal Formatting for Classification

First, these 5-channel processed recordings of the four human subjects were concatenated into one long recording. After that we have generated a random permutation without repetition of the array of integers in the range [0, *m*], where m is the overall number of given instructions in the joined signal. This permutation was used as a recipe of how to mix the instructions from the initial joined signal. This permutation gives us another array of the same dimension, in which each element represents the index of the corresponding instruction in the initial array.

6. FEATURE EXTRACTION

There are three types of features typically extracted from sEMG signals: (i) time domain, (ii) frequency domain, and (iii) time-frequency domain features. Time domain features are computationally simple operations applied to signal amplitude, and they represent the level and duration of muscle activation. Frequency domain features provide information about signal spectrum as a function of time, thus obtaining characteristic features from the signal's power spectral density. Time-frequency domain features are able to localize signal's energy both in time and spectrum, but usually include computationally more expansive tools like the short-time Fourier transform and wavelet transform. In order to decrease the latency in the robot hand control system, emphasis was given to time and frequency domain features, which are computationally less demanding.

6.1. Time Domain Features

We have used the following time domain features for the classifier design:

Mean Absolute Value – Estimation of a signal's mean absolute value calculated over a rectified signal by summing up all the amplitudes within the time window, divided by the length of the window:

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i|, \tag{2}$$

where N is the length of the time window and x_i is the signal value at i-th sample.

Root mean square – Square root of the signal's mean value within the time window of N samples, which represents an effective value of the EMG signal related to the force exerted by a muscle:

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|X_n|^2}, \tag{3}$$

Waveform length – Cumulative signal length that represents an integrated measure of amplitude, frequency, and sequence duration:

$$WL = \sum_{i=1}^{N}|\Delta x_i|, \tag{4}$$

where $\Delta x_i = x_i - x_{i-1}$.

EMG signal variance – It is related to the force exerted by a muscle and represents the measure of signal power:

$$VAR = \sigma^2 = \frac{1}{N-1}\sum_{k=1}^{N}x_k^2, \tag{5}$$

Zero crossing – Number of intersections of the recorded EMG signal with the zero-value baseline, i.e., measure of how many times the signal value changed its sign. It provides approximate information about signal frequency.

6.2. Frequency Domain Features

We have used the following frequency domain features for the classifier design:

Median frequency – Midpoint of power distribution, i.e., frequency that divides power spectral density into two regions with equal amounts of power:

$$\sum_{k=0}^{f_{median}} P(f_k) = \sum_{k=f_{median}}^{f_s} P(f_k), \tag{6}$$

Mean frequency – Average frequency calculated as the sum of products of the EMG power spectrum and frequency divided by the total sum of the power spectrum:

$$MNF = \frac{\sum_{j=1}^{M} f_j P_j}{\sum_{j=1}^{M} P_j}, \tag{7}$$

The first step in the feature extraction process is data segmentation. Data segmentation assumes defining a time window that contains signal data samples to be used for a single feature extraction. There are two typical approaches to data segmentation using: (i) overlapping windows and (ii) non-overlapping windows. Overlapping windows provide more accurate classification but are computationally more demanding and slower.

Since the shortest recorded gesture was 501 samples long, we have used a 300 samples long time window for data segmentation to enable feature extraction even from the shortest gesture. Also, we have used 50% overlapping to increase the number of the obtained samples. All the samples that included borderline data, i.e., the transition between two different input gestures, were manually excluded from the final dataset.

Since we have calculated 7 different features over 4 channels of EMG signal, the final input feature vector had the dimension 1x28. The complete dataset contained 54,701 samples, and the final input feature matrix had the dimension 54,701x28. The ground truth signal, containing corresponding true output classes for each sample, had the dimension 54701x1.

7. EMG PATTERN RECOGNITION

As mentioned, many different approaches have been proposed so far and proved efficient for the classification problem of EMG signals, including: fuzzy classifiers, different types of neural networks, hidden Markov models, support vector machines (abbr. SVM), k-nearest neighbors (abbr. KNN), etc. In this paper, we have implemented and compared different variations of SVM and KNN approaches.

7.1. Cross-Validation and Normalization

It is usual to divide a dataset using the hold-out method into three distinct subsets: (i) training set, (ii) validation set, and (iii) test set. The training set is used for model selection, the validation set is used for hyper-parameters tuning within the selected model, and the test set is used for the evaluation of the trained model's performance. We have used a combination of cross-validation and hold-out methods. First, we have selected the 20% of data for the test set using the hold-out method, and then we have applied k-fold cross-validation (with value k=5) to the rest of the data (80%) to generate different combinations of training and validation sets. The selected test set had 10,940 samples, while the rest of

the data contained 43,761 samples. Namely, one of the k data partitions has always been selected for validation, and the rest of the k-1 partitions have been used for validation. The process is repeated until each of k=5 partitions gets selected as a validation set.

Classifier training has been performed in two different setups: (i) with original feature vectors and (ii) normalized feature vectors. Normalization (standardization) was performed using Z-normalization, which maps input feature vector set into output feature vector set with approximately zero mean and standard deviation close to 1. Also, we have performed classificatory training with and without dimensionality reduction. The original input feature vector had dimension 1x28. The reduction was performed using Principal Component Analysis, but it did not further improve the obtained classifier's performance, so it is left out from the further discussion.

7.2. K-Nearest Neighbors

KNN is a nonparametric supervised learning approach for estimating data probability density distribution. This is a late learning method that processes data only when the classification request arrives. Classification of the novel data input is based on the class membership of the k nearest neighbors in the feature space of the training dataset, according to the relevant metric, i.e., the novel input is assigned to the most common class in its neighborhood (majority vote principle). We have trained, validated, and tested multiple KNN classifiers with the following variation of free parameters:

- Neighborhood size k: 1, 5, 10;
- Metrics: Euclidian, City-block (Manhattan);
- Ponders (weights associated to each neighbor): equal ponders, inverse proportional to distance.

7.3. Support Vector Machines

SVM is essentially a binary maximal margin classifier that searches for a hyperplane in a high-dimensional feature space that separates the data belonging to different output classes. In the case of linearly separable data, a hyperplane is selected to provide maximal distance (margin) between the plane and the closest members of both classes. These closest data samples on the border line of the margin are called support vectors. In the case of linearly inseparable data, it is possible either to use (i) soft margins or (ii) to apply the so-called "kernel trick." Soft margins allow some data samples to cross the margin or even the hyperplane and enter the wrong class. The kernel trick assumes nonlinear mapping of original feature vectors into another higher-dimensional feature space in which the data becomes linearly separable. A linear decision hyperplane in novel space corresponds to a nonlinear decision boundary in the original feature space. In this paper we have used the kernel trick with several different kernels. For more details about the underlying methodology and mathematical foundations of KNN and SVM algorithms, we refer a reader to some of the many available textbooks on that subject [27, 28]. We have trained, validated, and tested multiple SVM classifiers with the following variation of free parameters:

- Kernel: Gaussian, Linear, Square;
- Regularization parameter: 1, 10, 100;
- Multi-class method: One-vs.-One, One-vs.-All.

8. DISCUSSION AND RESULTS

In order to evaluate the performance of each implemented classifier, a corresponding confusion matrix was generated for each of them. It is a square matrix whose rows represent true (ground truth) class labels and columns represent predicted labels as output from the classifier. Diagonal elements of the confusion matrix represent the numbers of correctly classified data samples. Off-diagonal elements represent the numbers of misclassified samples.

KNN, Non-standard data, Validation set

Confusion matrix

	Power 0	Pinch 1	Lateral 2	Open 3
0	6772	134	30	182
1	226	11291	553	1232
2	141	920	11253	251
3	311	1464	158	8837

Statistics

Precision	0.909	0.818	0.938	0.841
Sensitivity	0.951	0.848	0.896	0.821
Specificity	0.982	0.917	0.976	0.950
Accuracy	0.872	0.872	0.872	0.872
F measure	0.930	0.832	0.916	0.831
F_macroAVG	0.87738			
F_microAVG	0.87185			

KNN, Non-standard data, Test set

Confusion matrix

	Power 0	Pinch 1	Lateral 2	Open 3
0	1702	31	8	38
1	68	2811	168	280
2	24	195	2867	56
3	80	400	43	2169

Statistics

Precision	0.908	0.818	0.930	0.853
Sensitivity	0.957	0.845	0.913	0.806
Specificity	0.981	0.918	0.972	0.955
Accuracy	0.873	0.873	0.873	0.873
F measure	0.932	0.832	0.921	0.829
F_macroAVG	0.87808			
F_microAVG	0.87285			

KNN, Standard data, Validation set

Confusion matrix

	Power 0	Pinch 1	Lateral 2	Open 3
0	6889	87	28	114
1	170	11940	428	770
2	108	501	11710	246
3	237	919	139	9475

Statistics

Precision	0.931	0.888	0.952	0.893
Sensitivity	0.965	0.897	0.932	0.880
Specificity	0.986	0.951	0.981	0.966
Accuracy	0.914	0.914	0.914	0.914
F measure	0.949	0.893	0.942	0.887
F_macroAVG	0.91739			
F_microAVG	0.91438			

KNN, Standard data, Test set

Confusion matrix

	Power 0	Pinch 1	Lateral 2	Open 3
0	1727	16	4	32
1	34	2977	121	195
2	21	122	2953	46
3	59	233	50	2350

Statistics

Precision	0.938	0.889	0.944	0.896
Sensitivity	0.971	0.895	0.940	0.873
Specificity	0.987	0.951	0.977	0.967
Accuracy	0.915	0.915	0.915	0.915
F measure	0.954	0.892	0.942	0.884
F_macroAVG	0.91809			
F_microAVG	0.91472			

Figure 10. *Comparative performance analysis for KNN-based classifier evaluated on validation and test sets, using both non-standard and standardized data.*

Based on the confusion matrix, different statistical performance metrics may be calculated. We have used four statistical measures that are typically used in machine learning to evaluate classification models: (i) precision, (ii) sensitivity (recall), (iii) specificity, and (iv) accuracy. We have also calculated (v) the balanced F-measure (F-score) as the equally weighted harmonic mean of precision and recall. Each of the five measures was calculated over each individual class. Additionally, we have calculated macro and micro average F-measure. The former calculates the F-measure for each class separately and then takes

a mean value over classes, while the latter calculates the F-measure globally over all data samples. Namely, macro averaging treats all classes equally while micro averaging treats all samples equally, and the difference may arise in the case of imbalanced data sets.

For both introduced types of classifiers (KNN-based and SVM-based), we have selected the configuration (parameter values) that achieved the best results and gave the comparative analysis of their performance using the introduced statistical measures. KNN classifier obtained the best results for the following set of parameters: k=10, city block metric, inverse proportional pondering. On the other hand, the SVM classifier obtained the best results for the following set of parameters: Gaussian kernel, regularization value 1, and one-vs.-all method. Both classifiers were evaluated on validation and test sets, using both non-standard and standardized input data. The comparative results for the KNN-based classifier are shown in Fig. 10.

SVM, Non-standard data, Validation set				
Confusion matrix				
	Power 0	Pinch 1	Lateral 2	Open 3
0	6795	124	33	166
1	214	11563	531	1000
2	140	834	11358	233
3	319	1093	172	9186
Statistics				
Precision	0.910	0.849	0.939	0.868
Sensitivity	0.955	0.869	0.904	0.853
Specificity	0.982	0.933	0.976	0.957
Accuracy	0.889	0.889	0.889	0.889
Fmeasure	0.932	0.859	0.921	0.860
F_macroAVG	0.89306			
F_microAVG	0.88897			

SVM, Non-standard data, Test set				
Confusion matrix				
	Power 0	Pinch 1	Lateral 2	Open 3
0	1696	25	8	50
1	61	2836	184	246
2	25	174	2899	44
3	95	281	48	2268
Statistics				
Precision	0.903	0.855	0.923	0.870
Sensitivity	0.953	0.852	0.923	0.842
Specificity	0.980	0.937	0.969	0.959
Accuracy	0.886	0.886	0.886	0.886
Fmeasure	0.928	0.854	0.923	0.856
F_macroAVG	0.89014			
F_microAVG	0.88656			

SVM, Standard data, Validation set				
Confusion matrix				
	Power 0	Pinch 1	Lateral 2	Open 3
0	6907	73	35	103
1	173	12053	417	665
2	119	445	11785	216
3	237	697	161	9675
Statistics				
Precision	0.929	0.908	0.950	0.908
Sensitivity	0.970	0.906	0.938	0.898
Specificity	0.985	0.960	0.980	0.970
Accuracy	0.924	0.924	0.924	0.924
Fmeasure	0.949	0.907	0.944	0.903
F_macroAVG	0.92585			
F_microAVG	0.92365			

SVM, Standard data, Test set				
Confusion matrix				
	Power 0	Pinch 1	Lateral 2	Open 3
0	1726	18	5	30
1	40	3019	103	165
2	23	106	2971	42
3	65	172	50	2405
Statistics				
Precision	0.931	0.912	0.949	0.910
Sensitivity	0.970	0.907	0.945	0.893
Specificity	0.986	0.961	0.980	0.971
Accuracy	0.925	0.925	0.925	0.925
Fmeasure	0.950	0.909	0.947	0.902
F_macroAVG	0.92714			
F_microAVG	0.92514			

Figure 11. *Comparative performance analysis for SVM-based classifier evaluated on validation and test sets, using both non-standard and standardized data.*

It can be seen from Fig. 10 that the KNN classifier obtained similar results on both the validation and test sets. However, significantly better results were obtained using standardized input data compared to the non-standard data. Micro and macro average F-measures are also mutually similar, which is not surprising, since we generated a highly balanced training data set with approximately uniform distribution of classes. Analogously,

the comparative results for the SVM-based classifier are shown in Fig. 11. Finally, we have compared the KNN-based classifier against the SVM-based one using the macro average F-measure. The results are presented as a bar graph in Fig. 12.

The reported results show the error rate in micro and macro F measures around 10% (±2%). We assume that there are several causes of this error. First, a rather small dataset with only four subjects, which was sufficient to validate our concept but probably not representative enough to get the most out of the used classifiers. Second, the high level of noise inherently present in sEMG signals and the variability of sEMG signal characteristics. This variability stems from: (1) placement of electrodes, (2) variability in neural control across users, and (3) time-dependent muscles' states due to fatigue [29]. All these reasons make this classification problem difficult, per se.

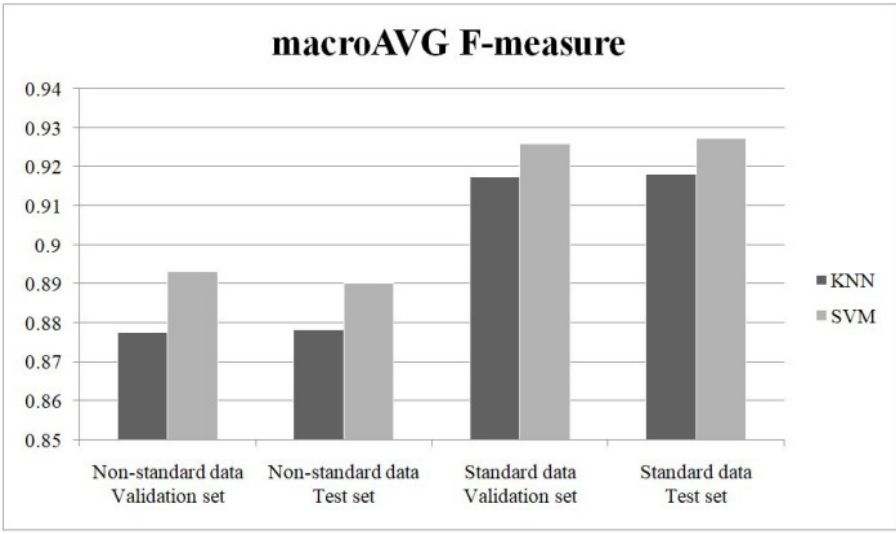

Figure 12. *Comparative reults of KNN-based ad SVM-based classifiers over validation and test sets using non-standard and standardized data.*

In this paper, the focus was on the development of an sEMG-based control interface and its integration with our custom-designed three-finger robotic hand. Hence, two rather classical machine learning approaches were implemented for static gesture recognition. The achieved results may be improved in several ways. First, by generating a larger and more representative dataset with more subjects. Second, using advanced novel feature extraction methods in combination with deep learning algorithms. Novel feature extraction methods include fused time-domain descriptors [30], temporal-spatial descriptors [31], wavelet scattering transform-based features [32], and explainable artificial intelligence (XAI) with a fusion framework for feature selection [33]. Most common deep learning methods used for sEMG feature extraction include convolutional neural networks (abbr. CNNs), long short-term memory neural networks (abbr. LSTMs), and temporal convolutional networks (abbr. TCNs). Several state-of-the-art approaches combine the aforementioned deep learning techniques. E.g., in [29], a multi-stream deep learning module was used utilizing the following architectures in different streams: TCN, LSTM-TCN, and CNN-TCN. In [34], a deep learning classifier based on a CNN–LSTM architecture was combined with the variational mode decomposition (abbr. VMD) technique to achieve hand gesture recognition from a single sEMG channel.

9. INTEGRATION WITH ROBOT HAND CONTROLLER

Based on the results of comparative analysis, we have selected the SVM classifier with standardized features, which achieved the highest F-measure, for final implementation and integration with the robot hand controller. The SVM-based classifier has been deployed as a MATLAB script, while the robot hand control has been implemented as a Python script. Communication between MATLAB and Python scripts is based on TCP/IP client-server architecture, and it has been implemented using sockets. Output from the MATLAB script is an integer from the set {0, 1, 2, 3}, representing the predicted output class for a perceived input hand gesture. We have performed a debouncing technique to provide stable output reading. This class number is sent via TCP/IP communication and further mapped into the corresponding predefined grasping motion of the robotic hand within the Python script. The Python script outputs a desired motion command for the robot controller. The command consists of desired position and imposed velocity and force limits.

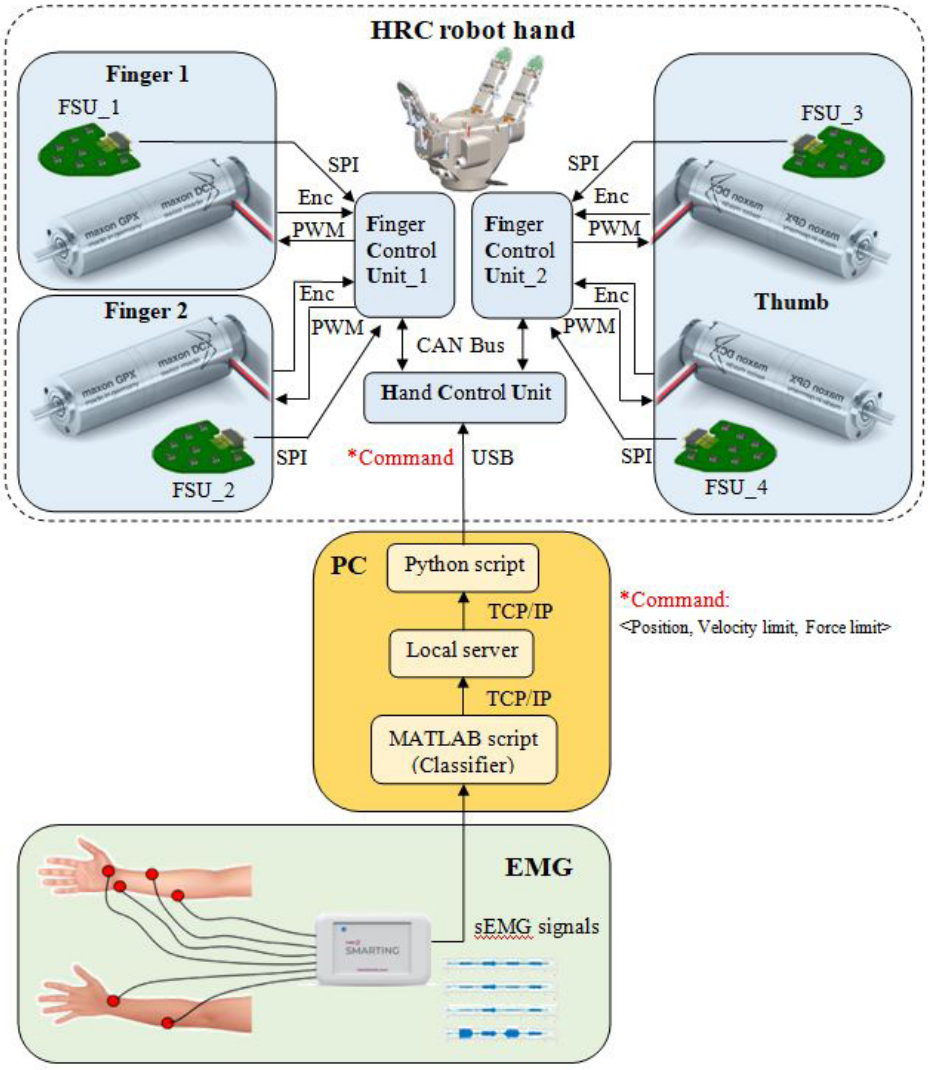

Figure 13. *Block diagram of HCU hand control architecture and HMI interface based on sEMG signals.*

Motion control of the robot hand has been hierarchically organized. The Python script directly communicates with the Hand Control Unit board (abbr. HCU) via USB communication. The HCU controller is a high-level central control unit that receives a command containing information about reference position and predefined velocity and contact force limits for each degree of freedom of the robot hand. The HCU controller passes the message towards two lower-level Finger Control Units (abbr. FCUs). Each FCU board controls two degrees of freedom. The first FCU control flexion/extension of the two robot fingers, while the second controls flexion/extension and rotation of the thumb. The FCU board contains a DC motor driver and collects data from motor encoders and Force Sensor Units (abbr. FSU) placed at the fingertips. FSUs are custom-designed 3-axis force sensors based on magnetic transducer technology. A block diagram of the HRC hand control architecture integrated with the proposed HMI interface based on sEMG signals is given in fig. 13. The motion controller of the HRC hand has been implemented as a cascade PID controller with three closed loops. The inner loop is closed around the measured motor current signal, the middle loop is a velocity control loop, and the outer loop is a position control loop.

10. CONCLUSION

In this paper, we have presented a human-machine interaction interface for teleoperated grasping control of the custom-designed three-finger robot hand. The proposed interface is based on surface electromyography measurement and static gesture recognition based on a support vector machines classifier.

As one of the contributions, a dataset of sEMG signals for hand grasping was created. an experimental study was conducted with four healthy participants, and a user-friendly GUI was designed for instructing the participants. However, the process of electrode placement was subject-dependent and sensitive. Therefore, in future work we will try to replace the *Smarting device* and classic electrodes with the *MYO armband* wearable sensor, which has 8 channels and wireless Bluetooth data streaming.

Current firmware of the HRC robotic hand does not support continuous teleoperation but only realization of discrete predefined grasps. It is one of the reasons why we addressed only the problem of gesture recognition and not continuous hand pose estimation. For future work, we intend to improve the hand firmware and also to address the problem of continuous teleoperation as well, with appropriate kinematic mapping between the human and robot hand. In line with that, we plan to generate a URDF model of the HRC hand and to integrate sEMG-based control with this model within some robotic simulator for underactuated systems, like the Drake simulator [35]. One of the main tasks for future work will be the optimization of all computational steps in order to decrease the latency and reach near real-time teleoperation, which was not required for the current research in this paper. Also, we intend to map the amplitude of sEMG signals to the force limit parameter in the motion command.

In the paper, systematic comparative analysis of two types of classifiers has been conducted and presented: (i) KNN-based classifiers and (ii) SVM-based classifiers. A multi-class classifier based on support vector machines has proven to be very satisfying for static gesture recognition, achieving a high F-measure on the gathered dataset. However, in future

work we will consider using some of the contemporary deep learning-based approaches to deal with dynamic gestures and to avoid feature engineering.

FUNDING

This research has been supported by the Ministry of Science, Technological Development, and Innovation (Contract No. 451-03-65/2024-03/200156) and the Faculty of Technical Sciences, University of Novi Sad, through the project "Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad" (No. 01-3394/1).

ACKNOWLEDGMENTS (OPTIONAL)

The authors would like to thank the participants for the experimental data collection for this work.

AUTHOR CONTRIBUTIONS (OPTIONAL)

Srđan Savić and Dunja Pavlović conceived and designed the study, particularly the protocol for sEMG dataset acquisition and classification algorithm. Dunja Pavlović conducted data gathering and signal processing. Srđan Savić performed statistical analyses. Andrej Čilag performed integration of the classifier with the robot hand controller and validation on the prototype. All of the authors contributed to writing the article.

INSTITUTIONAL REVIEW BOARD STATEMENT

The study was conducted according to the guidelines of the Declaration of Helsinki. The study was approved by the Ethical Committee of the Faculty of Technical Sciences, University of Novi Sad. The identification number of the approval is 01-1443/2.

INFORMED CONSENT STATEMENT

Informed consent was obtained from all subjects involved in the study.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. d. Visser and E. Phillips, "Comparing Virtual Reality Interfaces for the Teleoperation of Robots," 2020 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2020, pp. 1–7, doi: 10.1109/SIEDS49339.2020.9106630.

[2] Lii Neal Y., Pereira Aaron, Dietl Julian, Stillfried Georg, Schmidt Annika, Beik-Mohammadi Hadi, Baker Thomas, Maier Annika, Pleintinger Benedikt, Chen Zhaopeng, Elawad Amal, Mentzer Lauren, Pineault Austin, Reisich Philipp, Albu-Schäffer Alin, "Exodex Adam—A Reconfigurable Dexterous Haptic User Interface for the Whole Hand," *Front. Robot. AI*, Vol 8, March 2022.

[3] S. Li et al., "Vision-based Teleoperation of Shadow Dexterous Hand using End-to-End Deep Neural Network," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 416–422, doi: 10.1109/ICRA.2019.8794277.

[4] A. Handa et al., "DexPilot: Vision-Based Teleoperation of Dexterous Robotic Hand-Arm System," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 9164–9170, doi: 10.1109/ICRA40945.2020.9197124.

[5] B. Fang, D. Guo, F. Sun, H. Liu and Y. Wu, "A robotic hand-arm teleoperation system using human arm/hand with a novel data glove," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 2015, pp. 2483–2488, doi: 10.1109/ROBIO.2015.7419712.

[6] Caeiro-Rodríguez M, Otero-González I, Mikic-Fonte FA, Llamas-Nistal M. A Systematic Review of Commercial Smart Gloves: Current Status and Applications. Sensors (Basel). 2021 Apr 10;21(8):2667. doi: 10.3390/s21082667. PMID: 33920101; PMCID: PMC8070066.

[7] L. Dipietro, A. M. Sabatini and P. Dario, "A Survey of Glove-Based Systems and Their Applications," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 4, pp. 461–482, July 2008, doi: 10.1109/TSMCC.2008.923862.

[8] J. Vogel, C. Castellini and P. van der Smagt, "EMG-based teleoperation and manipulation with the DLR LWR-III," 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011, pp. 672–678, doi: 10.1109/IROS.2011.6094739.

[9] R. Meattini, S. Benatti, U. Scarcia, D. De Gregorio, L. Benini and C. Melchiorri, "An sEMG-Based Human–Robot Interface for Robotic Hands Using Machine Learning and Synergies," in IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 8, no. 7, pp. 1149–1158, July 2018, doi: 10.1109/TCPMT.2018.2799987.

[10] J. Lee, K. Kwon and W.H. Yeo, "Recent advances in wearable exoskeletons for human strength augmentation," Flex. Print. Electron. **7**, 1–31, (2022), DOI 10.1088/2058-8585/ac6a96

[11] Al-Shamayleh, A.S., Ahmad, R., Abushariah, M.A.M. *et al.* A systematic literature review on vision based gesture recognition techniques. *Multimed Tools Appl* **77**, 28121–28184 (2018). https://doi.org/10.1007/s11042-018-5971-z

[12] L. Guo, Z. Lu and L. Yao, "Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," in IEEE Transactions on Human-Machine Systems, vol. 51, no. 4, pp. 300–309, Aug. 2021, doi: 10.1109/THMS.2021.3086003.

[13] H.K. Hameed, W.Z. Hassan, S. Shafie, S.A. Ahmad, and H. Jaafar, "A Review on Surface Electromyography-Controlled Hand Robotic Devices Used for Rehabilitation and Assistance in Activities of Daily Living," JPO Journal of Prosthetics and Orthotic, 32(1), 3–13, (2020), *DOI: 10.1097/JPO.0000000000000277*

[14] R. Meattini, R. Suárez, G. Palli and C. Melchiorri, "Human to Robot Hand Motion Mapping Methods: Review and Classification," in IEEE Transactions on Robotics, vol. 39, no. 2, pp. 842–861, April 2023, doi: 10.1109/TRO.2022.3205510.

[15] R. Li, H. Wang and Z. Liu, "Survey on Mapping Human Hand Motion to Robotic Hands for Teleoperation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 5, pp. 2647–2665, May 2022, doi: 10.1109/TCSVT.2021.3057992.

[16] Bitzer, Sebastian & van der Smagt, Patrick. (2006). Learning EMG control of a robotic hand: Towards Active Prostheses. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006). 2006. 2819–2823. 10.1109/ROBOT.2006.1642128.

[17] Wan-Ting Shi, Zong-Jhe Lyu, Shih-Tsang Tang, Tsorng-Lin Chia, Chia-Yen Yang, "A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study," *Biocybernetics and Biomedical Engineering*, Volume 38, Issue 1, 2018, pp. 126–135,https://doi.org/10.1016/j.bbe.2017.11.001.

[18] Zheng N, Li Y, Zhang W and Du M (2022) User-Independent EMG Gesture Recognition Method Based on Adaptive Learning. *Front. Neurosci.* 16:847180. doi: 10.3389/fnins.2022.847180

[19] Gopal P, Gesta A, Mohebbi A. A Systematic Study on Electromyography-Based Hand Gesture Recognition for Assistive Robots Using Deep Learning and Machine Learning Models. Sensors. 2022; 22(10):3650. https://doi.org/10.3390/s22103650

[20] Parajuli N, Sreenivasan N, Bifulco P, Cesarelli M, Savino S, Niola V, Esposito D, Hamilton TJ, Naik GR, Gunawardana U, et al. "Real-Time EMG Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation", Sensors, Vol. 19, No. 20, 4596, 2019, DOI: https://doi.org/10.3390/ s19204596

[21] A. H. Al-Timemy, G. Bugmann, J. Escudero and N. Outram, "Classification of Finger Movements for the Dexterous Hand Prosthesis Control with Surface Electromyography," in IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 3, pp. 608–618, May 2013, doi: 10.1109/JBHI.2013.2249590.

[22] Rossi, M., Benatti, S., Farella, E. & Benini, L, "Hybrid EMG classifier based on HMM and SVM for hand gesture recognition in prosthetics". 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, pp. 1700–1705, 17–19 March 2015.

[23] Huang, Y., Englehart, K. B., Hudgins, B. & Chan, A. D. C, "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses", *IEEE Trans. Biomed. Eng*, Vol. 52, No. 11, pp. 1801–1811, November 2005.

[24] Kim, EunSu, JaeWook Shin, YongSung Kwon, and BumYong Park, "EMG-Based Dynamic Hand Gesture Recognition Using Edge AI for Human–Robot Interaction", Electronics, Vol. 12, No. 7, pp. 1541, March 2023, DOI: https://doi.org/10.3390/electronics12071541

[25] Yang Z, Jiang D, Sun Y, Tao B, Tong X, Jiang G, Xu M, Yun J, Liu Y, Chen B and Kong J, "Dynamic Gesture Recognition Using Surface EMG Signals Based on Multi-Stream Residual Network", Front. Bioeng. Biotechnol, Vol. 9, Article 779353, pp. 1–13, October 2021, DOI: 10.3389/fbioe.2021.779353, 2021.

[26] Lee, H., Lee, S., Kim, J. *et al,* "Stretchable array electromyography sensor with graph neural network for static and dynamic gestures recognition system", *npj Flex Electron*, Vol. 7, No. 20, pp. 1–13, April 2023, DOI: https://doi.org/10.1038/s41528-023-00246-3

[27] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed, Springer New York, NY, 2006.

[28] V. Kecman, "*Learning and Soft Computing - Support Vector Machines, Neural Networks, and Fuzzy Logic Models*", The MIT Press, 2001.

[29] Shin, J., Miah, A.S.M., Konnai, S., Takahashi I. and Hirooka K, Hand gesture recognition using sEMG signals with a multi-stream time-varying feature enhancement approach. *Sci Rep*, Vol. 14, No. 22061, September 2024, https://doi.org/10.1038/s41598-024-72996-7

[30] R. N. Khushaba, A. Al-Ani, A. Al-Timemy and A. Al-Jumaily, "A fusion of time-domain descriptors for improved myoelectric hand control," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 2016, pp. 1–6, doi: 10.1109/SSCI.2016.7850064.

[31] R. N. Khushaba, A. H. Al-Timemy, A. Al-Ani and A. Al-Jumaily, "A Framework of Temporal-Spatial Descriptors-Based Feature Extraction for Improved Myoelectric Pattern Recognition," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1821–1831, Oct. 2017, doi: 10.1109/TNSRE.2017.2687520.

[32] Al-Taee, A.A., Khushaba, R.N., Zia, T., Al-Jumaily, A. (2022). Feature Extraction Using Wavelet Scattering Transform Coefficients for EMG Pattern Classification. In: Long, G., Yu, X., Wang, S. (eds) AI 2021: Advances in Artificial Intelligence. AI 2022. Lecture Notes in Computer Science(), vol 13151. Springer, Cham. https://doi.org/10.1007/978-3-030-97546-3_15

[33] Naveen Gehlot, Ashutosh Jena, Ankit Vijayvargiya, Rajesh Kumar, Surface electromyography based explainable Artificial Intelligence fusion framework for feature selection of hand gesture recognition, *Engineering Applications of Artificial Intelligence*, Vol. 137, Part A, November 2024, No. 109119, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2024.109119.

[34] Prabhavathy T., Vinodh Kumar Elumalai, Balaji E., Dhanasekaran Sandhiya, A surface electromyography based hand gesture recognition framework leveraging variational mode decomposition technique and deep learning classifier, *Engineering Applications of Artificial Intelligence*, Vol. 130, No. 107669, April 2024, https://doi.org/10.1016/j.engappai.2023.107669.

[35] DRAKE robotic simulator [Internet], [Reviewed 19 June 2023, Cited 19 June 2023], Available at: https://drake.mit.edu/