

Research Article

Vector Autoregressive Hierarchical Hidden Markov Models for Extracting Finger Movements Using Multichannel Surface EMG Signals

Nebojša Malešević¹,^{ORCID} Dimitrije Marković,² Gunter Kanitz,³ Marco Controzzi,³ Christian Cipriani,³ and Christian Antfolk¹

¹Department of Biomedical Engineering, Lund University, 221 00 Lund, Sweden

²Chair for Neuroimaging, Department of Psychology, Technische Universität Dresden, 01069 Dresden, Germany

³The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy

Correspondence should be addressed to Nebojša Malešević; nebojsa.malesevic@bme.lth.se

Received 9 June 2017; Revised 27 December 2017; Accepted 14 January 2018; Published 18 February 2018

Academic Editor: Zoran Obradović

Copyright © 2018 Nebojša Malešević et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a novel computational technique intended for the robust and adaptable control of a multifunctional prosthetic hand using multichannel surface electromyography. The initial processing of the input data was oriented towards extracting relevant time domain features of the EMG signal. Following the feature calculation, a piecewise modeling of the multidimensional EMG feature dynamics using vector autoregressive models was performed. The next step included the implementation of hierarchical hidden semi-Markov models to capture transitions between piecewise segments of movements and between different movements. Lastly, inversion of the model using an approximate Bayesian inference scheme served as the classifier. The effectiveness of the novel algorithms was assessed versus methods commonly used for real-time classification of EMGs in a prosthesis control application. The obtained results show that using hidden semi-Markov models as the top layer, instead of the hidden Markov models, ranks top in all the relevant metrics among the tested combinations. The choice of the presented methodology for the control of prosthetic hand is also supported by the equal or lower computational complexity required, compared to other algorithms, which enables the implementation on low-power microcontrollers, and the ability to adapt to user preferences of executing individual movements during activities of daily living.

1. Introduction

The methods for estimating the intention of an amputee to control the movement of a prosthetic hand often relies on the interpretation/decoding of the electrical activity of muscles (electromyograms, EMG) recorded on the skin-surface of the residual limb [1, 2]. Even though information regarding the muscle activity could be obtained in various ways, commercially available prosthetic hands commonly use only a few surface EMG channels. Furthermore, the classification algorithms implemented in such prostheses generally comprise only calculation of an amplitude based EMG feature (e.g., root mean square) that is thresholded to obtain binary control of a single hand function [3]. Even in the case of multifunction prostheses with a plurality of independent joints,

the same control paradigm is still employed where switching between different grasps can depend on non-EMG inputs [4], such as smartphone interfaces, specific movements measured using inertial sensors, object recognition algorithms based on camera systems [5], or EMG inputs such as cocontractions (contractions of multiple muscles). Relatively low percentage of users is capable of fully exploiting the capabilities of multifunction prosthetic hands. This suggests that the control strategies need to be improved in order to achieve an intuitive link between intention and the resulting artificial hand function.

With the ongoing development of dexterous prosthetic hands [6–10], the gap between the potential dexterity offered by the new hands and the actual capabilities of human-machine interfaces is even more evident. In contrast to the

commercial devices, a variety of EMG recording techniques [11, 12], signal preprocessing [13], and classifiers have been proposed in the academia [14]. One of the approaches for increasing selectivity and sensitivity of the EMG signals includes surgical intervention in order to place recording electrodes on top of or inside targeted muscles [15]. This technique was shown to provide superior inputs for any type of EMG based controller; however, it is limited by need of surgical intervention and may not be widely accessible. As an alternative, EMG picked up at the surface of the skin provides a safe, noninvasive interface, at the cost of selectivity. Considering that the underlying mechanism of a classification algorithm is to localize the source of the electrical activity (active muscle), the methodology employed for similar problems is often based on multiple recording sites. Therefore, using a multichannel EMG approach provides an interesting perspective for improving the performance of surface EMG controllers. Increasing the number of the input signals also enables advanced data processing and classification techniques. Indeed, computational methods such as support vector regression [16], tree-structured neural network [17], Bayesian inference [18], ICA clustering [19], hidden Markov models [20], nonnegative matrix factorization [21], and various pattern recognition approaches [22–25], demonstrated promising classification results while using multiple discrete EMG channels or high-density surface EMG electrodes.

As one of the approaches for decoding finger movements using only multiple EMG signals, in this study we propose a novel classification algorithms based on the combination of EMG feature extraction and piecewise modeling of the feature temporal dynamics, incorporated in hierarchical hidden semi-Markov models (HHMM) [26–28] and an online Bayesian classifier implemented through model inversion. We developed and assessed two variations of the aforementioned method and included special cases of these algorithms with reduced computational requirements. In order to assess our algorithm but also to provide an extensive evaluation of the impact of different EMG features on the classification accuracy, we compared its performance with some of the states of the art classifiers. This work is an extension of the study presented in [29].

2. Materials and Methods

2.1. Subjects. Five able bodied subjects participated in the experiment. All the subjects provided informed consent, and the study was approved by the Regional Ethical Review Board in Lund, Sweden.

2.2. Protocol. The methodology presented in this research relies on multielectrode EMG signals for movement classification. During the measurement procedure, a subject was comfortably seated with the right hand resting in a neutral position. The subject was asked to perform a movement to match a hand image shown on the screen in front of him/her. The requested movements were flexion/extension of all five fingers and adduction/abduction of the thumb (12 different movements). Movement and resting periods between movements were of equal length (5 seconds) and were timed by

a LabVIEW (National Instruments, Austin, TX) custom application. The participant was visually prompted by the program, which synchronously acquired the EMG signal and the current cue annotation. Two different datasets (one for training and one for testing) each consisting of five repetitions of each movement, totaling 60 movements, and the rest states, were stored with the intended class information, for offline analysis.

2.3. EMG Recording. The dataset used in this paper was previously recorded and used in the publication by Huang et al. [30]. The EMG signals were acquired using a modified version of the amplification and acquisition system [31] and a LabVIEW program. The system acquired 16 channels of EMG, sampled at 1.6 kHz per channel and with a band-pass filter between 0.5 Hz and 800 Hz with 16-bit resolution and a gain of 56 dB per channel. The Red Dot Ag/AgCl (3M Health Care, Germany) electrodes that were used in the study were placed on the forearm of the participants as shown in Figure 1. For the 16 channel recordings, twelve electrodes were placed on the superficial flexor muscles on the volar side of the forearm and four electrodes were placed on the superficial extensor muscles on the dorsal side of the forearm. Electrodes were placed in order to cover as many independent muscles as possible. A sample of recorded EMG signals is provided in Figure 2.

2.4. Feature Extraction. A set of commonly used EMG features in the time domain was selected as an additional comparison criterion in this study. The choice of time domain features, versus spectral or time-frequency domain features, is in line with the tendency of deriving a control chain that could be implemented in an embedded system with limited processing power/speed. Among the relatively high number of reported features, we chose three-amplitude-related and three-frequency-related time-based EMG features. We implemented functions for the following EMG features as reported in [13, 32, 33].

2.4.1. Mean Absolute Value. Mean Absolute Value (MAV) is the favored EMG feature in many myoelectric control applications. It is calculated within a moving average window of the rectified EMG signal:

$$\text{MAV} = \frac{1}{N} \sum_{n=1}^N |x_n|, \quad (1)$$

where N denotes window length and x_n the n th EMG sample within current window.

2.4.2. Root Mean Square. Root Mean Square (RMS) calculation is, similar to MAV, also windowed function:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2}. \quad (2)$$

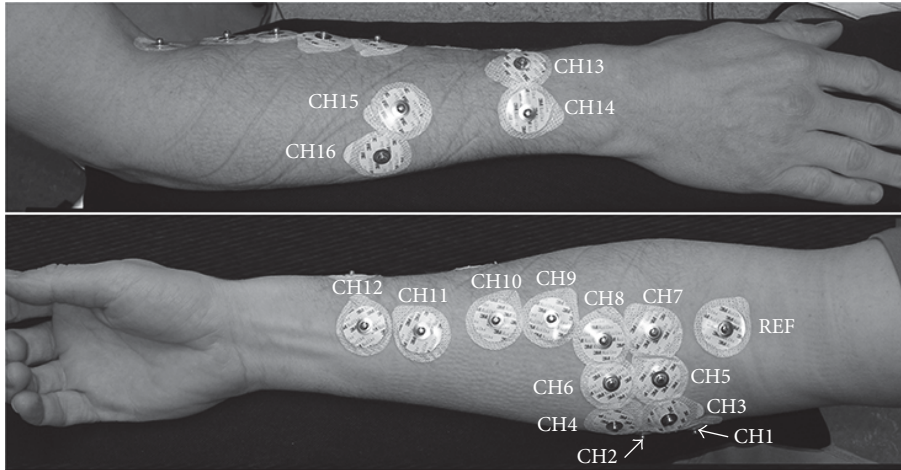


FIGURE 1: Measurement setup as in [25]. 16 monopolar channels were positioned in accordance with underlying hand muscles. The set of targeted muscles was chosen based on hand movements performed during the measurement protocol.

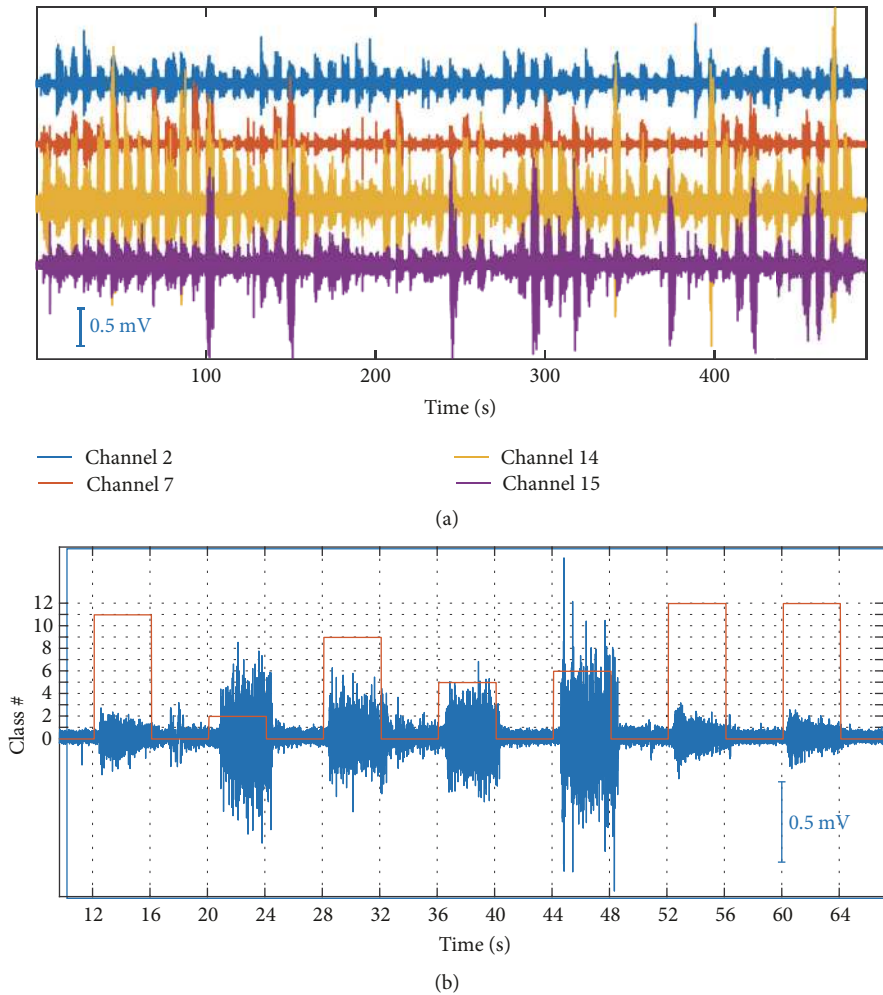


FIGURE 2: (a) shows sample EMG channels (2, 7, 15, and 15). (b) shows only channel 7 raw EMG signal (blue) with the visual cues/classes (red). This figure is an example of the latencies between visual cue and the EMG onset. This is also noticeable for the movement endings which are delayed with respect to the visual cue. Another important piece of information presented in this figure is similarity between EMG waveforms for the same movements (i.e., class 12).

In the case of the signal with mean value close to zero, such as surface EMG (sEMG) recorded with relatively large transcutaneous electrodes, the RMS becomes standard deviation of the input time series.

2.4.3. Variance. Variance (VAR) in the case of sEMG signal could be simplified enabling fast implementation in an embedded system. The resulting formula is the following:

$$\text{Var} = \frac{1}{N-1} \sum_{n=1}^N x_n^2. \quad (3)$$

2.4.4. Slope Sign Change. Slope Sign Change (SSC) is a time domain method used to estimate frequency feature if the EMG signal. The calculation of the SSC relies on counting changes between positive and negative slope among three consecutive samples. To limit SSC calculation only to periods with EMG activity, the threshold function is imposed in the feature extraction method:

$$\text{SSC} = \sum_{n=2}^{N-1} f [(x_n - x_{n-1}) \times (x_n - x_{n+1})] \quad (4)$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases}$$

2.4.5. Zero Crossing. Zero Crossing (ZC) is the function that counts the number of consecutive EMG samples that change sign within the sliding window. Similar to SSC calculation, the threshold function is imposed to remove calculation of the ZC during periods without pronounced EMG activity:

$$\text{ZC} = \sum_{n=2}^{N-1} [\text{sgn}(x_n \times x_{n+1}) \times f[x_n - x_{n+1}]] \quad (5)$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases}$$

2.4.6. Willison Amplitude. Willison amplitude (WA) is a measure related to superimposed action potentials that make the EMG signal. The WA is the number of consecutive differences between consecutive samples that exceed set threshold:

$$\text{WA} = \sum_{n=1}^{N-1} f[|x_n - x_{n+1}|] \quad (6)$$

$$f(x) = \begin{cases} 1, & \text{if } x \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases}$$

To produce comparable input conditions, the sliding window for all feature calculations was the same (250 ms width and 25 ms overlapping).

In the case of SSC, ZC, and WA calculations, thresholds were set based on the estimated white noise level during recording. As each recording session started with a rest

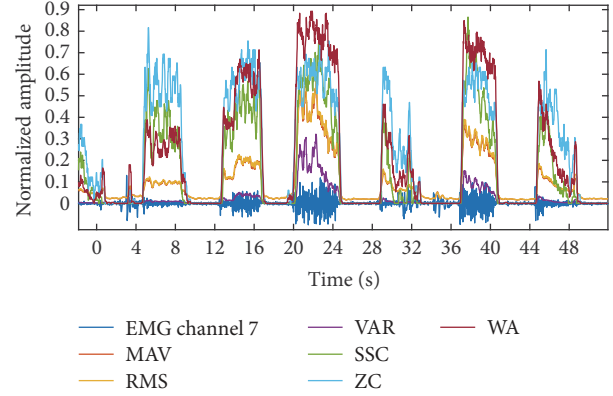


FIGURE 3: Example of the extracted features calculated from the EMG signal recorded on a single channel. The figure reveals that MAV and RMS features, in this case, almost overlap, but other features contain complementary information regarding EMG signal.

period, it was convenient to use first 0.5 s to calculate thresholds that are used throughout the same recording. A sample of calculated features is shown in Figure 3.

All feature extraction calculations were conducted using MATLAB, with moving window size of 250 ms and a displacement of 25 ms.

2.5. Classification Algorithm. Here, we will extend our previous work [29] in which we derived Bayesian online classifier using vector autoregressive hierarchical hidden Markov models (VARHHMM), with a classifier based on vector autoregressive hierarchical hidden semi-Markov models (VARHHSMM) [26–28]. The flowchart for the classifier's components is shown in Figure 4. The basic presumption of this approach is that the feature dynamics in time domain during a movement could be estimated by a number of VAR(1) segments defined via a unique set of parameters. A movement denotes a finger movement represented in the EMG feature space.

From the basic flowchart, it can be noted that the main difference between the two algorithms is in the top layer of the hierarchy. In the case of the VARHHMM, this layer only takes signal likelihood as the input to for the layer that identifies the current movement. In the case of the VARHHSMM, the current movement label will also depend on the expected duration of individual movements when identifying the active moment label. In what follows, we will refer to these two classifiers as hierarchical hidden Markov model (HMM) and hierarchical hidden semi-Markov model (HSMM), where we dropped the VAR and hierarchical labels for readability.

We assumed here that the selected time domain EMG features could be represented by a number of segments. To accommodate signal dynamics in real-time and enable creating a classifier suitable for embedded implementation, we defined individual segments as vector autoregressive (VAR) models of the first-order VAR(1). Using the VAR process for approximating the dynamics of a segment of an EMG

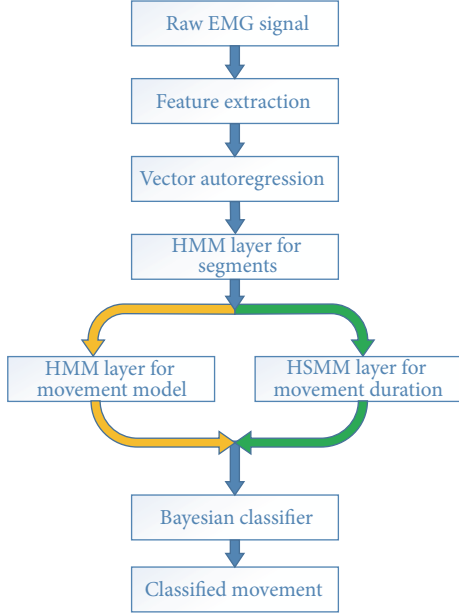


FIGURE 4: VARHMM flowchart. At the hierarchically higher HMM layer, the two algorithms are split before the Bayesian inference layer. While one algorithm for HMM (orange path) is the same as the HMM for segments, the other algorithm is based on hidden semi-Markov model (HSMM) paradigm that incorporates movement duration as a free parameter.

time series allows for efficient prediction of feature dynamics and implicitly accounts for the noise presented in the recorded EMG signals. Formally, the VAR(1) process is defined as

$$\vec{y}_t = \vec{\mu}^{(i,m)} + A_1^{(i,m)} \vec{y}_{t-1} + \vec{\psi}_t^{(i,m)}, \quad (7)$$

where \vec{y}_t is the latest acquired signal feature in vector form, $\vec{\mu}^{(i,m)}$ is the mean value of the signal, i denotes the segment within the m th finger movement, $A_1^{(i,m)}$ is the transition matrix of the VAR(1) model, and $\vec{\psi}_t^{(i,m)}$ is the random normal variable with zero mean. Note that the dynamics within each segment i and movement m are completely defined by the following set of parameters $(\vec{\mu}^{(i,m)}, A_1^{(i,m)}, \Sigma^{(i,m)})$, where $\Sigma^{(i,m)}$ denotes covariance of $\vec{\psi}_t^{(i,m)}$.

Based on predicted and measured sample vectors we can calculate which of the possible (fitted) VAR(1) models provides the best description of the recorded time series. The transition between individual segments (between different VAR(1) models) is considered as stochastic; thus, in our realization it is modeled as a HMM (as in [26]). The transitions between segments of a single movement are constrained by additional hidden states which define the currently active movement. For this reason, we introduced another HMM chain that acts as the second level of the hierarchy. The transitions between possible VAR(1) models of a single movement and different movements are defined through the transition

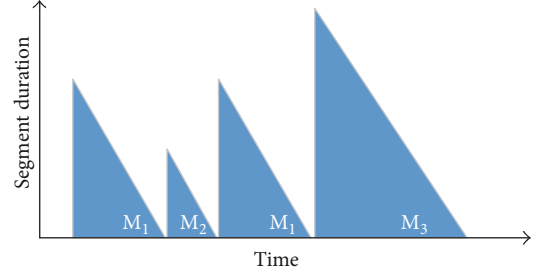


FIGURE 5: HSMM movement models. M1–M3 denote different dynamic movements. The HSMM algorithm uses distributions of individual movements durations to constrain transitions between them.

matrices for each HMM chain. The transition matrices on both levels of the hierarchy are estimated during the training procedure that relies on Viterbi expectation maximization algorithm [34].

By applying an approximate Bayesian inference procedure to the hierarchically arranged HMM and the emission distributions defined with VAR(1) models, we can classify each of the measured data points in a time series. The Bayesian inference for real-time signal classification relies on estimating the posterior probability for each movement. The posterior probability is calculated by combining expectations over movement probabilities with the evidence provided by the emission distributions (observation likelihoods). The expectations (predictions) are estimated using the posterior distribution at previous time sample and the transition matrices of movements and associated segments. Using the values of the posterior probability $p(M_t = m | y_{t:0})$ at the latest acquired signal y_t and the current movement $M_t = m$, we can classify the current sample as

$$l_t = \underset{m}{\operatorname{argmax}} p(M_t = m | y_{t:0}), \quad (8)$$

where l_t is the classification label obtained by our algorithm.

The HSMM algorithm is similar to the HMM algorithm with the only difference being that each movement includes a sequence of durations associated with it. This approach increases the number of free parameters with an additional vector comprising prior duration probabilities of individual movements. In other words, the probability of switching between movements is also influenced by the model of movement duration. An illustration of a specific realization of the movement switching sequence of the HSMM is shown in Figure 5.

The implementation of the HSMM algorithm relies on a timer, or counter (Cnt) in the case of equidistant sampling. Upon entering a state/movement, the counter is set to predefined value related to that particular state duration and with every new sample is decremented. When the counter reaches 0, the state/movement is free to change. Upon switching to a new state, the counter is reset to zero. This additional condition is defined as

$$\begin{aligned}
& p(M_t = i \mid M_{t-1} = j, \text{Cnt} = d) \\
&= \begin{cases} \delta(i, j) & \text{if } d > 0 \text{ (remain in the movement)} \\ \Upsilon_{i,j} & \text{if } d = 0 \text{ (free to change movement)}, \end{cases} \quad (9)
\end{aligned}$$

where M_t denotes movement at time sample t and Υ denotes transition matrix between movements.

In the case of HSMM, the duration of a movement is provided in form of a prior distribution. As the measurement protocol was executed in automated manner, all performed movements are of the same duration (~ 5 s) with some variability related to subject's reaction times to a visual cue. In order to simulate more realistic end-user scenario, we defined prior distribution to have a duration range with 2-second variability (5 ± 1 s). As the simplest scenario derived a priori, the imposed distribution was defined as uniform within the whole range. To estimate influence of the range of the imposed distribution, the accuracy score was also calculated for ranges from 5 ± 0.05 s to 5 ± 5 s.

As the number of segments per movement is the free parameter of both algorithms, it was of specific interest to evaluate performance of the classification with respect to a movement division by the automated optimization procedure. Specifically, the case of a single segment per movement was analyzed independently. The rationale for this lies in the extracted features dynamics in the time domain. When visualizing the recorded signals, a movement in a feature space reassembles a step response of a dynamic system, with a relatively short transient compared to the constant part. When ignoring the noise part of the vector autoregressive model (see above) we can rewrite the update equation as

$$\vec{y}_t = \vec{y}_{t-1} + B_1 (\vec{\theta} - \vec{y}_{t-1}), \quad (10)$$

where the current sample y_t is a function of previous sample and the distance of the previous value from the terminal value $\vec{\theta}$ in the $t \rightarrow \infty$ limit. The constant matrix B_1 defines the rate of change along each dimension of the recorded signal. This form of the basic VAR representation is convenient for illustrating the signal trajectory after the onset of a movement (see Figure 6). The advantages of using single segments per movements area significantly shorter optimization time (explained in the following section) and a simplified real-time processing implementation as the lower HMM layer is removed.

2.6. Optimization. The estimation of the free parameters of all models presents the only computationally demanding procedure of the presented methodology. During the optimization, the free parameters of the VAR models (number of segments per movement, mean signal values of individual segments $\vec{\mu}^{(i,m)}$, covariance matrix $\Sigma^{(i,m)}$, transition matrix $A_1^{(i,m)}$, and between segment's transition matrix $\Upsilon^{(m)}$ which defines the HMM structure) are evaluated using a Viterbi algorithm coupled with an expectation maximization algorithm (EM) [35]. The flowchart of the optimization procedure is shown in Figure 7.

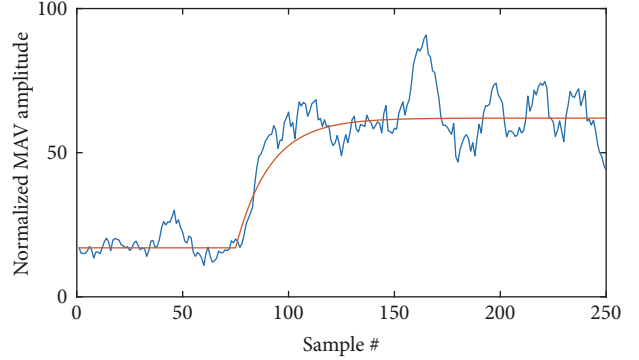


FIGURE 6: Example of the segment switching with AR(1) model (a one-dimensional VAR(1) model) with the forced one segment per movement. Blue line represents MAV feature dynamics when switching from a rest state to a movement, and the red line represents example of a single AR segment optimized for that movement.

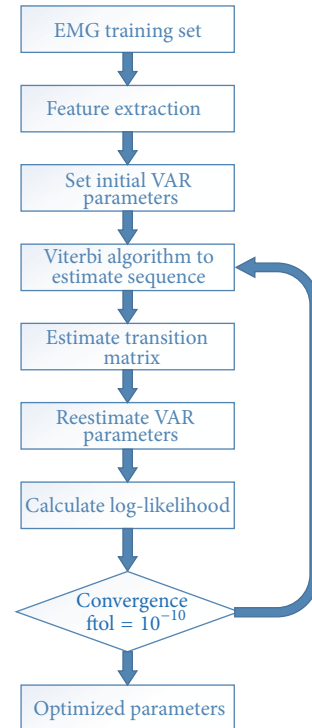


FIGURE 7: Optimization procedure flowchart. The main load of the optimization procedure is division of movements to segments. With a higher number of segments per movement, the Viterbi algorithm needs a large number of initial conditions for the model to converge to the global minimum instead of converging to a local minimum. This movement dividing method is not deterministic, resulting in slightly different models with every optimization run.

The optimization procedure is an iterative process where from the initial values of VAR parameters the Viterbi algorithm determines the most likely sequence of segments and estimates transition matrices. The following step includes reestimation of VAR parameters based on a maximum likelihood calculation and the most likely separation of training

data suggested by the Viterbi algorithm. The procedure shown in Figure 7 is iterated with the log-likelihood set as the convergence criterion. When the convergence criterion reaches a value of 10^{-10} , the iterations are stopped. The whole optimization method was based on the implementation of the Viterbi EM algorithm provided in the `pyhsmm-auto-regressive` Python library [36].

Bearing in mind that the optimization method could converge towards a local optimum, we repeated the whole optimization procedure for 1000 different initial values of free parameters and only the solution with the lowest value of Bayesian Information Criterion (BIC) [37] was kept as final. This iterative extension of the Viterbi EM algorithm helped us to improve the convergence of the parameter estimation to optimal parameters values. The Python 3.6 codes for the parameters optimization and classification simulation are available upon request sent to the corresponding author.

2.7. Classifiers Used for Benchmarking. To evaluate the effectiveness of our algorithm, we used some of the most commonly used classification methods including

- (i) Linear Discriminant Analysis (LDA),
- (ii) Quadratic Discriminant Analysis (QDA),
- (iii) K -nearest neighbors (knn), with $k = 15$,
- (iv) Support Vector Machine with the Error-Correcting Output Codes (SVM ECOC),
- (v) Linear Classifier with the Error-Correcting Output Codes (LC ECOC),
- (vi) Linear Discriminant Analysis with the Error-Correcting Output Codes (LDA ECOC),
- (vii) Naive Bayes (NB),
- (viii) Random Forest (RF),
- (ix) Decision Tree (DT).

The implementation of selected classifiers was done using `fitc` function in the MATLAB 2016b (The MathWorks Inc., Natick, MA). Similar to the VARHHMM algorithms, the classifiers were trained on only the training data set and tested on the test set and no postprocessing of class-annotations was performed.

3. Results

The ground truth for classification in this paper is set to be the visual cue presented to the subject. Although there is the noticeable latency between the visual cue and the muscle activity in both movement onset and cessation as shown in Figure 2, we decided to use this as a reference to provide the same basis for all of the classification methods. This was done to force classifiers to adapt to the transient processes between consecutive states. As expected, this approach results in lower overall classification accuracy but provides interesting insights into classifiers performances during dynamic, continuous changes of input parameters.

The main metric for comparing features and performance of the classifiers was accuracy. For the accuracy calculation,

each feature sample (one observation every 25 ms) was treated individually. Moreover, as the rest state is directly involved in active motor driving by prosthesis control, it was also considered equal to the other movement classes. With this evaluation paradigm, each classification point was categorized as a true positive (TP) if the predicted class matches the visual cue label, or as a false positive (FP) if there is mismatch between predicted class and the visual cue. This approach generates a large number of data points, which was in our case approximately 20000 per dataset that could be used to investigate the performances of various features and classifiers.

The cumulative results for all subjects are presented in Table 1. As the accuracy results for the subjects do not fit into normal distribution, the accuracy values presented in the table are median values among all subjects. The differences between classifier and features are emphasized by ranking top three features and classifiers overall but also top three results for each EMG feature. From Table 1, it could be noted that HSHMM outperforms other classifiers for the majority of tested feature extraction algorithms. It is also interesting that the HSMM with only one segment per movement closely follows performance of the HSMM with multiple segments per movement. Among feature extraction algorithms, the MAV features resulted in the best performance across different classifiers and subjects; RMS and WA lead to slightly lower results compared to MAV, while using VAR, SSC and ZC failed to produce similar results. Among classifiers, only NB underperformed, while other classifiers produced mutually comparable results.

As the average accuracy across all subjects and all classifiers was the greatest in the case of MAV feature, the results obtained using this feature were analyzed in more detail. Table 2 shows accuracy results for each classifier and each subject when the MAV feature was used. The results show that all VARHHMM variations are top-ranked, with the HSMM algorithm having the highest median accuracy score. Although at the top ranks, the difference of median values is not prominent.

With the focus on VARHHMM methodology, to illustrate ability to separate individual finger movements and resting periods, the confusion matrix is shown in Figure 8. For this example, we selected the best performing feature method among all classifiers (MAV) and the subject with the median results (Subject 4). This was done to analyze in more detail the case close to the end-user scenario. The common indicator of the classification effectiveness is the confusion matrix which is used to reveal weak points of a classifier. Figure 8 is showing the confusion matrix for the HSMM algorithm. It could be noted that majority of movements (2, 4, 5, 6, 7, 11, 12, and 13) are classified with just sporadic misclassifications. For these movements, most of the errors are the consequence of the latency between the visual cue (used as the golden standard in this paper) and the actual muscle activity of the targeted movement. This effect is responsible for the movement-to-rest and rest-to-movement misclassifications. The rest class, except the errors related to delayed transitions, is generally well classified. The movements (3, 8, 9, and 10) have more samples that are labeled incorrectly as some other movement.

TABLE 1: Classifier accuracy in % for LDA, QDA, knn (15), SVM ECOC, LC ECOC, LDA ECOC, KNN ECOC, NB, HHMM, HSMM, and HHMM with only one segment per movement and HSMM with one segment per movement. The top ranked classifiers are marked with * (top ranked), ** (second ranked), and *** (third ranked).

| | LDA | QDA | KNN | SVM ECOC | LC ECOC | LDA ECOC | KNN ECOC | NB | RF | DT | HHMM | HSMM | HHMM S1 | HSMM S1 |
|--------|-------|-------|-------|----------|----------|----------|----------|-------|-------|-------|----------|----------|---------|---------|
| MAV* | 80.28 | 81.94 | 79.10 | 82.43 | 78.06 | 82.02 | 79.19 | 69.36 | 79.37 | 64.86 | 83.00*** | 84.30* | 82.59 | 83.35** |
| RMS** | 80.16 | 81.40 | 78.89 | 82.11* | 76.92 | 81.37 | 79.74 | 69.81 | 79.44 | 63.68 | 81.58** | 81.52*** | 69.95 | 81.16 |
| VAR | 71.77 | 38.12 | 78.22 | 79.76** | 70.22 | 76.23 | 78.40 | 51.91 | 79.44 | 64.61 | 78.91*** | 82.36* | 42.08 | 70.85 |
| SSC | 74.80 | 69.29 | 73.73 | 75.78** | 75.30*** | 75.90* | 73.99 | 62.56 | 73.83 | 62.70 | 71.41 | 75.56 | 71.41 | 75.56 |
| ZC | 74.75 | 73.03 | 72.64 | 75.90* | 75.68** | 75.42 | 73.08 | 60.17 | 72.34 | 59.72 | 74.26 | 75.44*** | 74.26 | 75.44 |
| WA*** | 79.20 | 75.54 | 76.73 | 79.50 | 77.96 | 80.37*** | 76.92 | 63.06 | 77.74 | 64.93 | 78.39 | 84.09* | 78.39 | 82.05** |
| Median | 77.00 | 74.29 | 77.47 | 79.63** | 76.30 | 78.30 | 77.66 | 62.81 | 78.59 | 64.15 | 78.65*** | 81.94* | 72.83 | 78.36 |

TABLE 2: Classifier accuracy in % for the different subjects and MAV feature.

| SUB | LDA | QDA | KNN | SVM ECOC | L ECOC | LDA ECOC | KNN ECOC | NB | RF | DT | HMM | HSMM | HMM S1 | HSMM S1 |
|--------|-------|-------|-------|----------|--------|----------|----------|-------|-------|-------|-------|-------|--------|---------|
| 1 | 82.07 | 86.39 | 79.10 | 82.43 | 80.60 | 82.65 | 79.19 | 65.97 | 79.65 | 69.70 | 87.50 | 91.18 | 86.70 | 90.94 |
| 2 | 80.28 | 81.94 | 79.96 | 84.83 | 80.85 | 83.55 | 80.45 | 69.36 | 83.01 | 65.68 | 83.51 | 85.61 | 83.51 | 85.42 |
| 3 | 79.69 | 56.82 | 77.19 | 81.26 | 72.77 | 80.31 | 77.98 | 63.79 | 79.48 | 57.75 | 76.28 | 75.76 | 71.51 | 73.43 |
| 4 | 81.64 | 82.31 | 82.95 | 82.83 | 77.92 | 82.02 | 83.12 | 75.33 | 80.56 | 70.08 | 83.00 | 84.30 | 82.59 | 83.35 |
| 5 | 80.21 | 66.88 | 78.54 | 80.97 | 78.06 | 78.45 | 78.78 | 70.25 | 74.62 | 63.94 | 77.20 | 82.96 | 62.86 | 78.39 |
| Median | 80.28 | 81.94 | 79.10 | 82.43 | 78.06 | 82.02 | 79.19 | 69.36 | 79.65 | 65.68 | 83.00 | 84.30 | 82.59 | 83.35 |
| Rank | 8 | 7 | 11 | 5 | 12 | 6 | 10 | 13 | 9 | 14 | 3 | 1 | 4 | 2 |

| | | | | | | | | | | | | | |
|----|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 9267 | 71 | 78 | 42 | 39 | 71 | 30 | 87 | 65 | 69 | 35 | 22 | 27 |
| 2 | 68 | 716 | 0 | 1 | 0 | 0 | 8 | 0 | 2 | 0 | 0 | 0 | 0 |
| 3 | 40 | 0 | 561 | 3 | 3 | 2 | 0 | 1 | 326 | 0 | 0 | 0 | 0 |
| 4 | 93 | 0 | 148 | 720 | 44 | 89 | 38 | 215 | 63 | 15 | 7 | 0 | 0 |
| 5 | 26 | 0 | 0 | 0 | 702 | 36 | 0 | 69 | 0 | 66 | 0 | 148 | 0 |
| 6 | 19 | 0 | 0 | 0 | 0 | 601 | 0 | 0 | 0 | 0 | 17 | 0 | 0 |
| 7 | 169 | 1 | 1 | 2 | 0 | 0 | 721 | 2 | 0 | 0 | 1 | 0 | 1 |
| 8 | 14 | 2 | 0 | 6 | 0 | 0 | 0 | 409 | 0 | 1 | 0 | 0 | 0 |
| 9 | 15 | 0 | 4 | 0 | 0 | 0 | 0 | 1 | 345 | 0 | 1 | 0 | 0 |
| 10 | 96 | 0 | 1 | 18 | 11 | 0 | 2 | 2 | 0 | 465 | 0 | 0 | 0 |
| 11 | 83 | 0 | 0 | 0 | 1 | 1 | 1 | 2 | 0 | 184 | 739 | 0 | 0 |
| 12 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 629 |
| 13 | 94 | 10 | 7 | 8 | 0 | 0 | 0 | 13 | 0 | 0 | 0 | 0 | 773 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

FIGURE 8: Confusion matrix, HSMM, MAV, Subject 4. There are around 800 observations per movement and 10100 in the rest class. The algorithm underperformed when discriminating classes 3, 8, 9, and 10, with class 9 being most problematic as the majority of observations were misclassified as class 3.

These errors, if consistent, could significantly impede control of a prosthesis. Thus, to examine the behavior of the VARHHMM implementations in a real-time application, the classification outputs were further analyzed.

Figure 9 contains some of the most severe examples of the misclassification errors produced by the HSMM algorithm.

Figure 9(c) is an illustration of the proper movement classification where only errors occur during transition between rest and the movement. Compared to this sample, Figure 9(a) depicts an error of the labeling and keeping the incorrect movement active almost from the beginning of the movement onset, Figure 9(b) depicts misclassification at the middle of the movement period, and Figure 9(d) depicts the indecisiveness of the algorithm to label samples at the transient period. Out of these examples, the most severe in terms of real-time application is the first case that could lead to the execution of the false actuation. The other cases are relatively easy to correct using a state machine and/or majority voting algorithm that prevents situations that are less likely to occur by the user's intent.

Other metrics that were implemented to calculate classifier behavior in the real-time are motion selection (MS) and motion completion (MC) times. The calculation of MS and MC is derived from the papers of Li et al. and Ortiz-Catalan et al. [38, 39]. MS was computed as the delay between cue onset and the first correctly classified sample. Due to the difference in features frequency, the condition for the MC was slightly modified compared to other papers. In our algorithm, a new EMG feature was produced every 25 ms, while in [38] every 100 ms, so instead of 10 correct classifications, we waited for 40 correct classifications until the MC condition was fulfilled. This change was done in order to have comparable results with the relevant publications. The same metric for MS and MC was employed for all subjects and all features. Figure 10 is showing MS and MC times mean values across features and classifiers. It could be noted that HSMM has

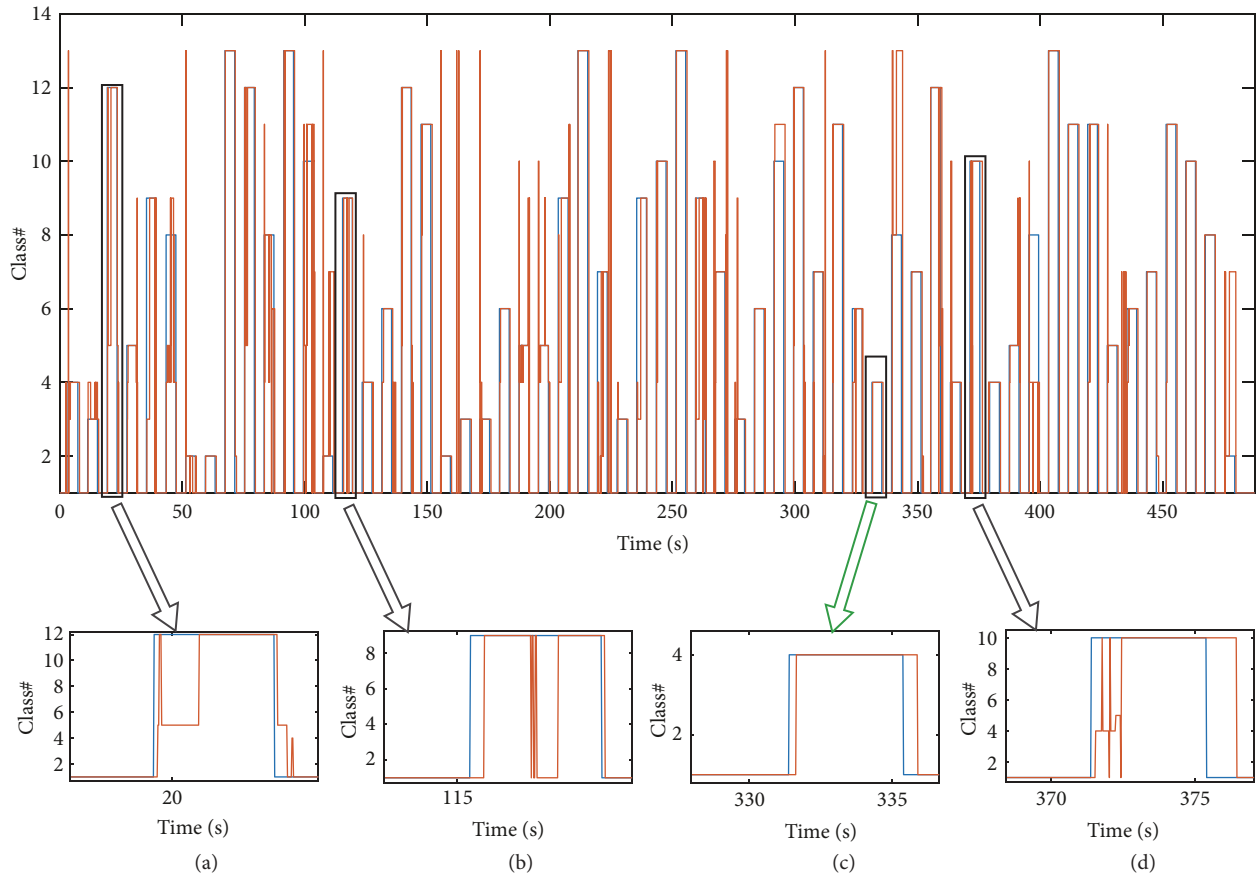


FIGURE 9: Sample classifications of the HSHMM. MAV (feature with the highest accuracies across all classifiers) and Subject 4 (median results).

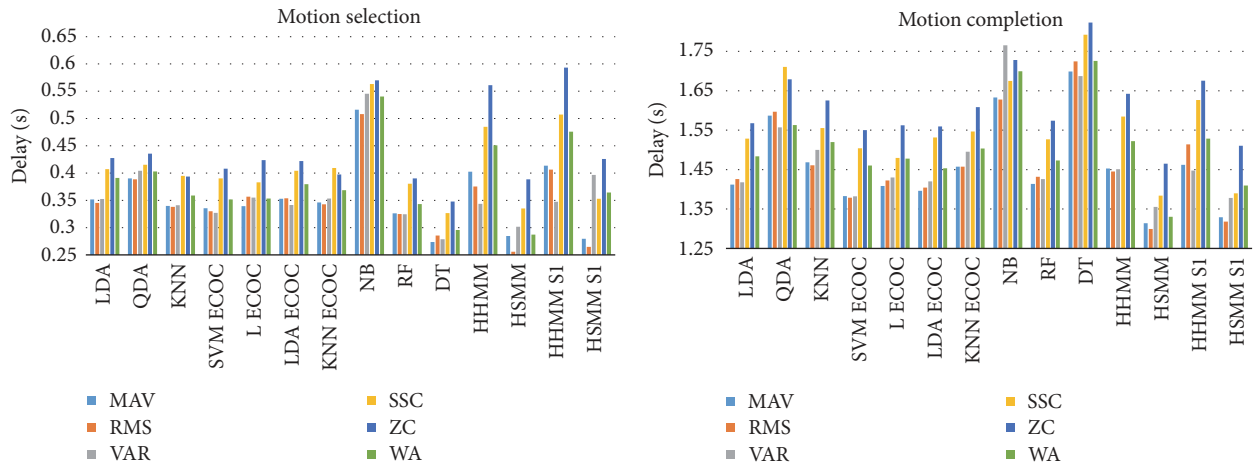


FIGURE 10: Motion selection (MS) and motion completion (MC) time. Mean values for MS time ranged from 0.25 s (HSM-RMS) to 0.6 s (HHMM S1-ZC) and for MC time ranged from 1.3 s (HSM-RMS) to 1.8 s (DT-SSC).

the lowest delay between cue and the first correctly classified sample, indicating that the method has a relatively low detection threshold compared to the other classifiers. It is also clear that the HSM works the best with amplitude based features (MAV, RMS, and WA). Another interesting

outcome is that HSM with only one segment per movement achieves similar results as HSM with multiple segments per movement when fed with MAV or RMS features of the EMG signal. On the other hand, the regular HHMM performed slightly worse on average than the best classifiers, such as

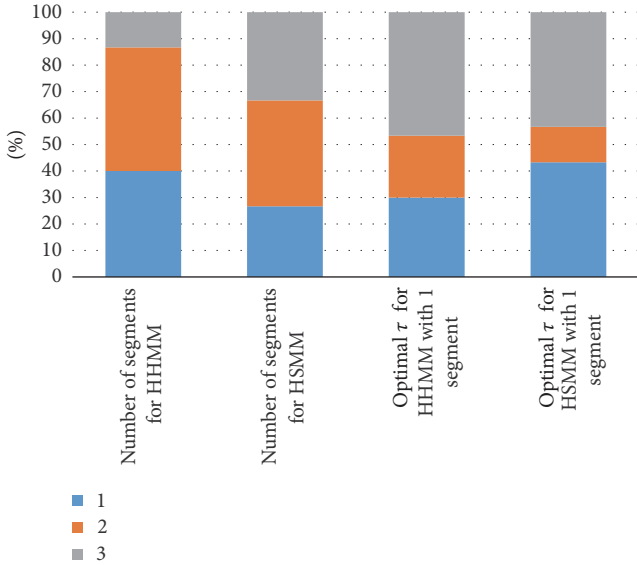


FIGURE 11: Optimization of the parameters. The chart shows that there is no clear advantage of using high number of segments per movement as in lot of cases even one segment is enough to achieve the highest classification accuracy. The same conclusion could be made for the number of AR samples as having only one previous sample available for making prediction could lead to high accuracy.

LDA, SVM ECOC, L ECOC, and LDA ECOC. Similar to HSMM, reducing number of segments per movement for the HHMM does not result in significant drop in performance related to MS and MC times.

We also analyzed another parameter, namely, the optimal number of segments per movement that results from the optimization procedure. For both HHMM and HSMM algorithms, two segments per movement most frequently produced the highest accuracy score (Figure 11). Another interesting parameter that was optimized and analyzed is the order of the vector autoregressive process (τ) in the HHMM and HSMM models with one segment per movement. The highest accuracy was achieved with 3 successive points, but what was not expected was relatively good accuracy with only one autoregressive sample (prediction relies only on the previous observation).

As the performance of HSMM algorithm depends on the predefined distribution of possible movement durations, analyzing this parameter was of great importance. During the measurements, the duration of movements was governed by the visual cues that were presented in automated manner. This way, the resulting movements have roughly the same duration of 5 s. To artificially introduce expected movement duration variability, we expanded the uniform duration distribution around central point of 5 s. With the extension of the expected movement duration, we evaluated classification accuracy. As presented in Figure 12, the dependency between classification accuracy and the range of permitted movement duration is almost linear in the mid-range. This finding is valuable for defining expected movement duration, while taking into account the tradeoff between accuracy and flexibility of the classifier to adapt to free movements by a user.

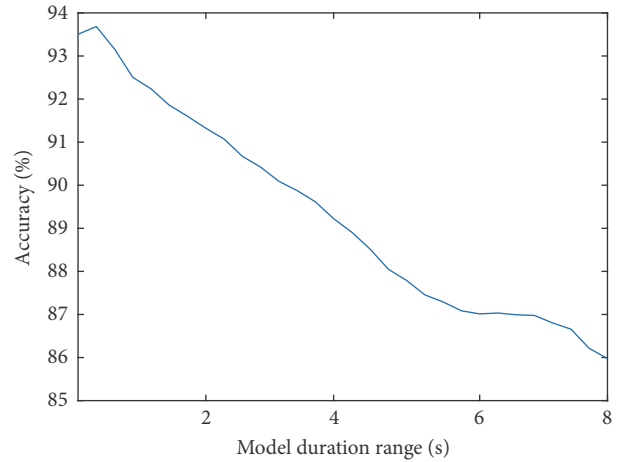


FIGURE 12: Dependence of the distribution range to the accuracy of the HSHMM classifier. The graph is showing that the accuracy decreases almost linearly with the increase of the movement duration range.

4. Discussion

In this paper, we presented novel algorithms for classifying features from surface EMG signals. Both of the proposed algorithms (HHMM and HSMM) are variations of the VARHHMM algorithm which combines vector autoregression, hidden Markov models, and Bayesian inference. The main focus of the presented results is the comparison between different EMG classifiers, including some of the most commonly used ones and VARHHMM variants. The results presented in Table 1 indicate that the HSMM algorithm with a priori fixed movement duration outperforms other classifiers. It should be also noted that the results of HSMM algorithm presented in this table are with the ad hoc uniform movement durations distribution between 3 s and 5 s. It is envisioned that, in the prolonged prosthesis use with this control algorithm, the movement duration distribution would be the parameter updated with each correctly executed movement. Thus, as the time of use increases, the distribution should become optimal and result in increased movement classification accuracy. The other derived algorithm (HHMM) was ranked overall just behind the SVM ECOC, mostly because of low accuracy when frequency related time-based features were used as inputs.

The results also reveal that reducing number of segments per movement to one does not result in considerable drop in accuracy scores for the derived algorithms (with the exception of VAR feature). This fact has significant impact on the possible implementation of the algorithms as the optimization, even in the case of large time series, for example, tens of minutes, could be executed in a matter of seconds. As the illustration, optimization of algorithm variants with free parameter of segments per movement takes up to 10 minutes per movement (Python 3.6 and Intel i7 processor). Depending on the embedded platform, our estimate is that optimization with one segment per movement could be still done in less than a minute. The other fact that contributes to

low processing during optimization and real-time application is that algorithm does not need high order autoregressive models. As presented in Figure 11, using even only one previous sample point could result in the highest accuracy score.

As this study was also aiming at finding out which feature extraction method works the best when coupled with the proposed algorithms and commonly used ones, we carried out a systematic test that included all features and subjects. With the EMG signals that we acquired, using MAV feature as the first step in the classification chain produced the highest accuracy scores among classifiers. The performance of MAV was closely matched by RMS, while WA produced the best results among the frequency related time-based features.

When compared with the reported results of Li et al. [38], the obtained MS and MC times calculated with our data and algorithms show some deviations. This is mostly present in MS values that represent the delay of the first properly classified sample. In the case of Li et al., MS values were in range 0.16–0.33 s, while in our simulations, only HSMM with MAV, RMS, and WA achieved similar performance. This behavior is the result of the difference in the ground truth choice: Li et al. used actual onset of the EMG activity, while we selected visual cue onset. With this in mind, it is expected that the MS times in our case have the superimposed lag of approximately 200–400 ms resulting from the latency between the cue and the muscle activation. That practically means that the HSMM would outperform reported classifiers if used in similar conditions. The same rationale goes for the MC times. Although classifiers tested in this study produced longer MC times on average, if reduced by the visual cue to muscle activity offset, the results would show that HSMM manages to correctly detect 40 samples (which correspond to 10 samples in paper of Li et al.) faster than the classifiers reported by Li et al.

5. Conclusion

The study performed on our set of multichannel surface EMG indicates that using MAV feature coupled with HSMM algorithm leads to movement decoding accuracy higher than other combinations of features and classifiers. This combination also guarantees the shortest MS and MC times, influencing a prosthetic hand to a user intent.

The main advantages inherent to our algorithms compared to the existing methods are the following. (1) Low computation complexity for the execution of the algorithm: following the optimization procedure, which is computationally demanding but executed only once per model, the implementation of our algorithm comes down to the basic algebraic operations that could be implemented even on low-end microcontrollers. This feature also permits true real-time execution with the delay only related to EMG feature extraction and AR depth. Additionally, we tested even faster variations of developed algorithms that have only one segment per movement. With the small drop in performances, these algorithms, especially HSMM with one segment per movement, significantly decreased optimization time and further decreased execution load. (2) Easy expansion: once optimized, models could be stored, and in the case of introducing

a new movement class, the optimization could be performed only on the newly added movement class. (3) Noise resilient: in comparison with some weak classifiers and threshold centered rule-based algorithms, the VARHHMM approach implicitly takes into account possible sources of stochastic noise.

The method proposed in this paper is intended for decoding individual finger movements for directly controlling actuated fingers of a prosthetic hand. Although desirable, this approach is not common as the prosthesis use during activities of daily living mostly consists of synergistic movements (grasps). Thus, as the alternative control paradigm, the proposed method could be implemented as the state machine classifier-controller. In this approach, the separable classes might be used to initiate fixed grasps (pinch/power/lateral/open) as this is what most commercial hands support.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the EU-Funded DeTOP Project (EIT-ICT-24-2015, GA no. 687905) and the Swedish Research Council (637-2013-444).

References

- [1] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews in Biomedical Engineering*, vol. 30, no. 4–6, pp. 459–485, 2002.
- [2] N. Jiang, S. Dosen, K.-R. Muller, and D. Farina, "Myoelectric control of artificial limbs—is there a need to change focus? [In the Spotlight]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 148–152, 2012.
- [3] A. Fougner, O. Stavadahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, "Control of upper limb prostheses: Terminology and proportional myoelectric control review," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 5, pp. 663–677, 2012.
- [4] C. Connolly, "Prosthetic hands from touch bionics," *Industrial Robot*, vol. 35, no. 4, pp. 290–293, 2008.
- [5] S. Došen, C. Cipriani, M. Kostić, M. Controzzi, M. C. Carrozza, and D. B. Popović, "Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation," *Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 1, article 42, 2010.
- [6] M. Controzzi, F. Clemente, D. Barone, A. Ghionzoli, and C. Cipriani, "The SSSA-MyHand: A dexterous lightweight myoelectric hand prosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 5, pp. 459–468, 2017.
- [7] *Touchbionics*, <http://www.touchbionics.com/>.
- [8] *bebionic*, <http://bebionic.com/>.
- [9] *Michelangelo*, <http://www.ottobockus.com/prosthetics/upperlimb-prosthetics/solution-overview/michelangelo-prosthetic-hand/>.

- [10] *Vincent hand*, <http://vincentsystems.de/en/>.
- [11] M. Ortiz-Catalan, R. Brånemark, B. Håkansson, and J. Delbeke, "On the viability of implantable electrodes for the natural control of artificial limbs: review and discussion," *Biomedical Engineering Online*, vol. 11, article 33, 2012.
- [12] S. Muceli and D. Farina, "Simultaneous and proportional estimation of hand kinematics from EMG during mirrored movements at multiple degrees-of-freedom," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 3, pp. 371–378, 2012.
- [13] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "A novel feature extraction for robust EMG pattern recognition," *Journal of Computing*, vol. 1, no. 1, pp. 71–80, 2009.
- [14] M. A. Oskoei and H. Hu, "Myoelectric control systems—a survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.
- [15] M. Ortiz-Catalan, B. Håkansson, and R. Brånemark, "An osseo-integrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs," *Science Translational Medicine*, vol. 6, no. 257, Article ID 257re6, 2014.
- [16] A. Ameri, E. N. Kamavuako, E. J. Scheme, K. B. Englehart, and P. A. Parker, "Support vector regression for improved real-time, simultaneous myoelectric control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 6, pp. 1198–1209, 2014.
- [17] F. Sebelius, L. Eriksson, C. Balkenius, and T. Laurell, "Myoelectric control of a computer animated hand: a new concept based on the combined use of a tree-structured artificial neural network and a data glove," *Journal of Medical Engineering & Technology*, vol. 30, no. 1, pp. 2–10, 2006.
- [18] S. Bitzer and P. Van Der Smagt, "Learning EMG control of a robotic hand: towards active prostheses," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 2819–2823, IEEE, May 2006.
- [19] G. R. Naik, A. H. Al-Timemy, and H. T. Nguyen, "Transradial amputee gesture classification using an optimal number of sEMG sensors: an approach using ICA clustering," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 8, pp. 837–846, 2016.
- [20] A. D. C. Chan and K. B. Englehart, "Continuous myoelectric control for powered prostheses using hidden Markov models," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 121–124, 2005.
- [21] C. Huang, X. Chen, S. Cao, B. Qiu, and X. Zhang, "An isometric muscle force estimation framework based on a high-density surface EMG array and an NMF algorithm," *Journal of Neural Engineering*, vol. 14, no. 4, Article ID 046005, 2017.
- [22] 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," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1821–1831, 2017.
- [23] A. A. Adewuyi, L. J. Hargrove, and T. A. Kuiken, "An analysis of intrinsic and extrinsic hand muscle EMG for improved pattern recognition control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 485–494, 2016.
- [24] N. Celadon, S. Došen, I. Binder, P. Ariano, and D. Farina, "Proportional estimation of finger movements from high-density surface electromyography," *Journal of NeuroEngineering and Rehabilitation*, vol. 13, no. 1, article 73, 2016.
- [25] C. Cipriani, C. Antfolk, M. Controzzi et al., "Online myoelectric control of a dexterous hand prosthesis by transradial amputees," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 3, pp. 260–270, 2011.
- [26] M. Yang, "Some properties of vector autoregressive processes with Markov-switching coefficients," *Econometric Theory*, vol. 16, no. 1, pp. 23–43, 2000.
- [27] M. Dong and D. He, "Hidden semi-Markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," *European Journal of Operational Research*, vol. 178, no. 3, pp. 858–878, 2007.
- [28] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.
- [29] N. Malešević, D. Marković, G. Kanitz, M. Controzzi, C. Cipriani, and C. Antfolk, "Decoding of individual finger movements from surface EMG signals using vector autoregressive hierarchical hidden Markov models (VARHHMM)," in *Proceedings of the International Conference on Rehabilitation Robotics (ICORR '17)*, pp. 1518–1523, IEEE, July 2017.
- [30] H. Huang, T. Li, C. Bruschini et al., "EMG pattern recognition using decomposition techniques for constructing multiclass classifiers," in *Proceedings of the 6th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob '16)*, pp. 1296–1301, IEEE, June 2016.
- [31] C. Antfolk, C. Cipriani, M. Controzzi et al., "Using EMG for real-time prediction of joint angles to control a prosthetic hand equipped with a sensory feedback system," *Journal of Medical and Biological Engineering*, vol. 30, no. 6, pp. 399–406, 2010.
- [32] N. Alves, E. Sejdjić, B. Sahota, and T. Chau, "The effect of accelerometer location on the classification of single-site forearm mechanomyograms," *Biomedical Engineering Online*, vol. 9, article 23, 2010.
- [33] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multi-function myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
- [34] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [35] A. Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2139–2156, 1999.
- [36] M. Johnson, *pyhsmm-autoregressive*, <https://github.com/mattjij/pyhsmm-autoregressive>.
- [37] B. North and A. Blake, "Learning dynamical models using expectation-maximisation," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 384–389, IEEE, 1998.
- [38] G. Li, A. E. Schultz, and T. A. Kuiken, "Quantifying pattern recognition—based myoelectric control of multifunctional transradial prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 185–192, 2010.
- [39] M. Ortiz-Catalan, B. Håkansson, and R. Brånemark, "Real-time and simultaneous control of artificial limbs based on pattern recognition algorithms," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 756–764, 2014.

