DOI: 10.2298/CSIS120601007A

Modeling and Visualization of Classification-Based Control Schemes for Upper Limb Prostheses

Andreas Attenberger[1] and Klaus Buchenrieder[2]

[1] Institut für Technische Informatik
Universität der Bundeswehr München
Neubiberg, Germany
Andreas.Attenberger@unibw.de
[2] Institut für Technische Informatik
Universität der Bundeswehr München
Neubiberg, Germany
Klaus.Buchenrieder@unibw.de

Abstract. During the development of control schemes for upper-limb prostheses, the selection of a classification method is the decisive factor on predicting the correct hand movements. This contribution brings forward an approach to validate and visualize the output of a chosen classifier by simulative means. Using features extracted from a collection of recorded myoelectric signals (MES), a training set for different classes of hand movements is produced and validated with additional MES recordings. Using the output of the classifier, the behavior of an actual prosthesis is simulated by controlling the 3D model of a prosthetic hand. For systematic comparison of feature sets and classification methods, a toolbox for MATLAB™ has been developed. Our classification results show, that existing classification schemes based on EMG data can be improved significantly by adding NIR sensor data. Employing only two combined EMG-NIR sensors, five motion classes comprising full movements, including pronation and supination, can be distinguished with 100% accuracy.

Keywords: Classification Algorithms, Decision Trees, Electromyography, Modeling, Prosthetic Hand, Simulation, Support Vector Machines, Visualization.

1. Introduction

Research on the employment of myoeletric signals for prostheses control has been conducted since the 1940s [13]. Myoelectric signals (MES) can be measured by placing electrodes on the skin located over the observed muscles. When a muscle is activated through a neurological impulse, transmitted from the brain, small changes in electrical potential can be detected on the surface of the skin. In order to actuate a prosthesis, these signals are processed. In their work, Englehart, Hudgins, Parker and Stevenson provide a definition for the signal-classification problem and preposition a multi-stage process [4]. In this process, the complexity of recorded data is reduced by the introduction of features like root-mean-square (RMS) values, which denote the average signal strength. After extraction, the selected features are fed into a classifier. Usually, a training-data set, with different classes for various hand movements or hand-positions, is created. Any new electromyographic (EMG) data can then be attributed to one of the given classes. Recently, research about a novel type of sensor using near-infrared (NIR) light, to detect muscle activity, has been disclosed [7] [6]. Near-infrared light is partially absorbed by the hemoglobin in the red blood cells. Due to this effect, different levels of absorption can be recorded using a NIR light source and a photodetector. As a result, the level of muscular activity in the area under the sensor can be observed and hand-positions as well as -movements detected.

In this contribution, we present a model of the classification process for upper-limb prostheses including subsequent simulation, validation and visualization. From the recorded sensor signals, RMS and zero crossing (ZC) features as well as a feature derived from the sensor's NIR component are extracted for five different hand movements. For training and classifier validation two different classification methods are demonstrated and compared. Both, an easy to implement decision tree algorithm as well as a more flexible multi-class support vector machine (SVM) are presented. For the simulation of this process, a 3D model of a hand prosthesis, as shown in Fig. 1, is employed for visualizing the classification results. This modeling and simulation solution is an example of the functionality offered by a custom-built MATLAB™ toolbox allowing the selection of features and the structured comparison of various classification methods for a faster evaluation of prosthesis control models. Furthermore, integrating additional information from NIR sensors leads to improved classification results. Over the years, an important factor in increasing classification accuracy for a higher number of hand movements has been the utilization of additional sensors [11]. However, achieving high accuracy for detecting four or more movement classes with only two sensors placed on the forearm remains challenging. Liu and Luo have built a classifier based on wavelet packet transformation and

Fig. 1. 3D Hand Prosthesis Model.

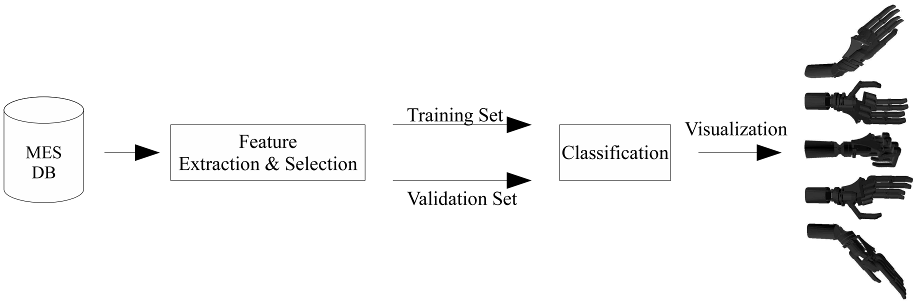


Fig. 2. Modeling, validating and visualizing the classification process.

a neural network (NN) that attains a detection rate of 98% for four hand movements [10]. Arvetti, Gini and Folgheraiter employ wavelet analysis and an NN to reach almost 97% accuracy for five different motion classes [1]. León, Leija and Muñoz identifiy seven different movements through a combination of discrete Fourier transformation and a NN with a success rate of up to 95% [9]. Note, that the last two methods only use either the first or the last part of the signal for identifying a class and not the full, transient movement. Additionally, only León, Leija and Muñoz include pronation and supination motion classes.

2. Method

This section describes our method of modeling, validation and visualization of the prosthesis control scheme. First of all, only employing EMG data, classification of five different hand movements is demonstrated for two different feature combinations. The features extracted from our database of hand movement recordings are used to train a SVM and a decision tree classifier, for which the results are subsequentially validated. Combining EMG and NIR sensor signals offers a significant improvement of the accuracy of a classifier. The final classification results are then used to control the visualization model of the prosthesis embedded in MATLAB™, as shown in Fig. 2.

2.1. Data Acquisition and Feature Extraction

For our investigation, we recorded 100 datasets for five different hand movements each comprising 20 data samples each. For every movement, the signals from two combined EMG and NIR sensors [7] were captured from the forearm of a proband. The sensors were placed over the extensor digitorum and the carpi radialis muscles. The data were recorded with a custom sensor system, integrating both, a single differential EMG sensor as well as a LED and a photo receiver for capturing the amount of near infrared light not absorbed in the underlying tissue.

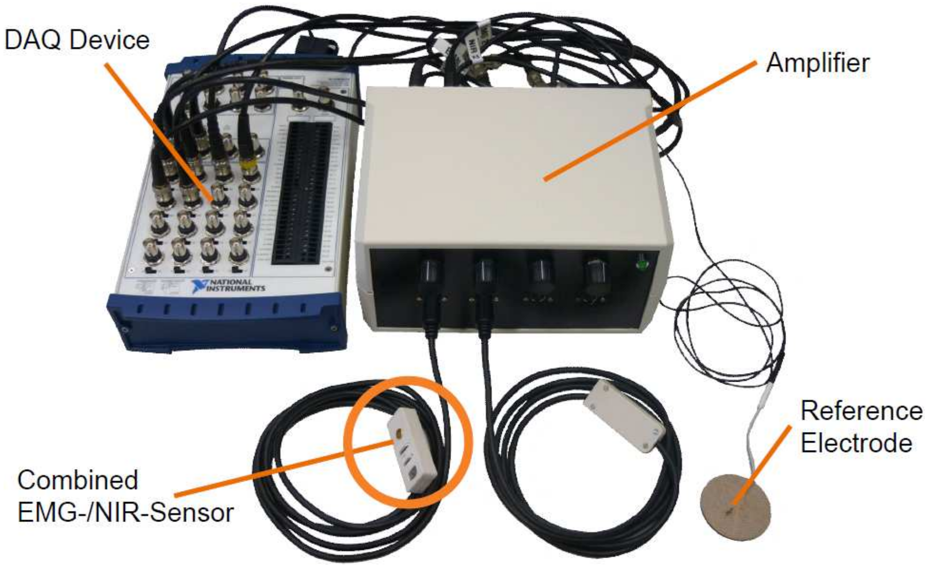


Fig. 3. The hardware used for acquiring EMG and NIR signals, including the DAQ and the sensor system.

The MES were amplified by a factor of about 10 dB. To prevent tissue damage from excessive heat, the NIR light emitted by the diode placed on the sensor is pulse modulated with a pulse rate of 16 Hz and a duty cycle of 2.5%. The rise and fall time of the pulses was 5%. For reducing interference between sensors, an offset of 15% was introduced. The enable signals for the pulses were generated by the MATLAB™ signal generator application displayed in Fig. 4 and output with a NI USB-6229 DAQ device from National Instruments. Fig. 3 shows the hardware setup necessary for acquiring the combined EMG and NIR signals. The EMG/NIR sensor consists of a single differential EMG electrode located between the NIR LED and the photo receiver. The sensors as well as a reference electrode connect to the main signal amplifier. The amplified analog signals are fed into the NI USB Device. Additionally, the enable signal output is also recorded with the DAQ device for further reference. The recordings were conducted with a frontend application created in MATLAB™ and Simulink™ using a sampling rate of 4096 Hz. Each five second data sample is enriched with a time-synchronous video recording of the proband's hand motion. The resulting data was saved in MATLAB™ binary files with the EMG and NIR recordings captured in arrays.

The first step towards creating a training set for the classifiers is the feature extraction from a myoelectric and a near-infrared signal. By this measure, the amount of input data and the complexity is reduced prior to the classification process [2]. Various features like the RMS, ZC or the waveform length can be extracted from EMG signals [12]. In order to gauge the strength of the MES for

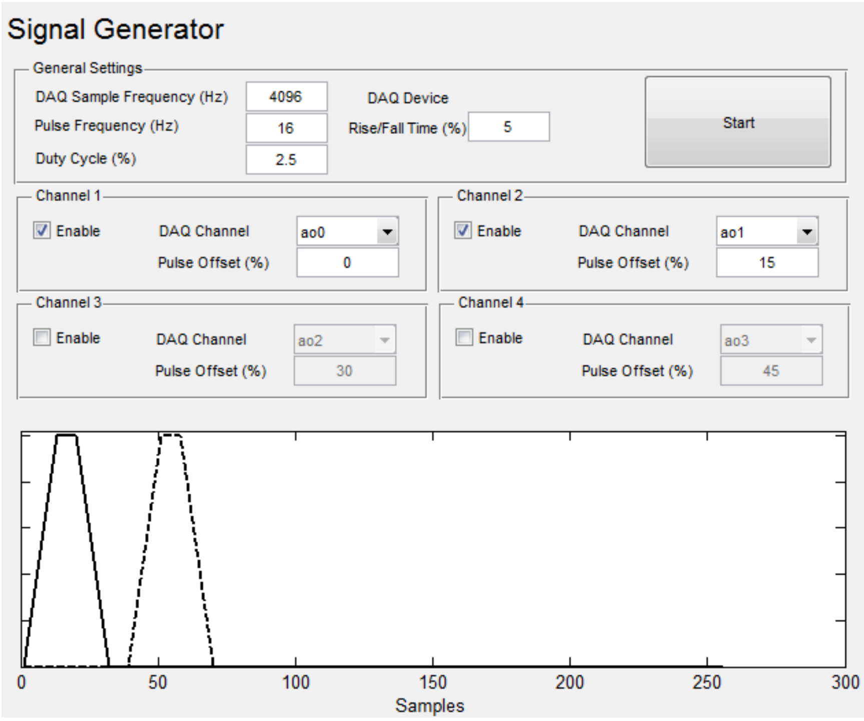


Fig. 4. Generator application for controlling the NIR sensor LEDs.

a set of N samples, the RMS results from:

$$x_{rms} = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^{N} (x_n^2)}\,. \tag{1}$$

The window size was set to 256 samples with an increment of 256. The RMS feature was also used for preprocessing the MES recordings. Several seconds of noise recorded before and after the actual hand movement were removed by amplitude threshold provisory clipping. This was realized by measuring the RMS values of noise recorded with the EMG signal. All recordings contained at least one second of noise before the start of the movement. The first second of each recording was split into four windows of 256 samples and the RMS value of each window was calculated. The maximum out of these four results was then compared to a sliding window of 1024 samples throughout the remaining recording. A window with an RMS value higher than that of the maximum noise RMS sample window was considered to contain the start of the movement. Finally, a window with a resulting RMS equal to or lower than the RMS noise threshold was assumed to mark the end of the movement. The recordings from all sensors for a single recording were trimmed to preserve the integrity of the

EMG signals. Fig. 5 shows the first second of the EMG signal containing noise. Brackets shown beneath the signal denote the actual movement signal as well as the last 1024 sample window discarded due to its RMS value below the noise threshold. The signals from the EMG and NIR feature combination were treated accordingly with the NIRS feature used for determining the beginning and end of a movement as the NIR sensor signal yields a lower noise-to-signal-ratio. Window sizes were adjusted for four values per second, i.e., a 256 window size and increment for both the RMS and the NIRS.

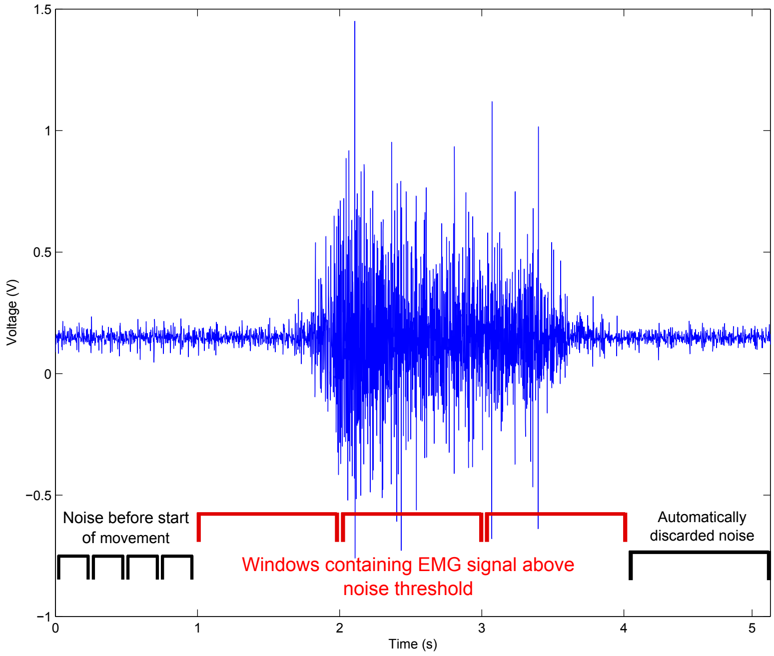


Fig. 5. A raw EMG signal recording with four 256-sample-windows used for determing the noise RMS and 1024-sample-windows for comparing the significant and ineffectual section.

In addition to the RMS, the ZC feature was also extracted from the EMG signal using:

$$x_{zc} = \sum_{n=0}^{N-1} I\{\mathrm{sgn}(n+1) \cdot \mathrm{sgn}(n) < 0\} \,. \tag{2}$$

As the LED is switched on periodically, a NIRS feature taking into account the pulse rate and duty cycle can be derived [6]. For the vectors $\overline{n} = \{n_1, ..., n_k\}$, consisting of the measured NIR signal in the observed time frame, and $\overline{e} = \{e_1, ..., e_k\}$, with $Ena(e_i)$ denoting the state of the enable signal (either 0 or 1 depending on an upper and lower threshold) at each point of time in the signal window, the NIRS feature can be calculated as follows:

$$\text{NIRS} = \text{Signal}(\overline{n}, \overline{e}) - \text{Offset}(\overline{n}, \overline{e})\,. \tag{3}$$

with

$$\text{Signal}(\overline{n}, \overline{e}) = \frac{\sum_{i=1}^{k} n_i \cdot Ena(e_i)}{\sum_{i=1}^{k} Ena(e_i)}\,. \tag{4}$$

$$\text{Offset}(\overline{n}, \overline{e}) = \frac{\sum_{i=1}^{k} n_i \cdot (1 - Ena(e_i))}{\sum_{i=1}^{k} (1 - Ena(e_i))}\,. \tag{5}$$

To produce window sizes of equal length, for which the EMG and the NIRS features can be combined, the NIRS window size and its increment was set to 256 samples as well. For the combined features, the NIRS feature was chosen as the source for the amplitude threshold provisory clipping. This is advantageous because the NIRS feature clearly indicates the beginning of a muscle contraction, revealing the motion of a hand with more precision than RMS alone. Fig. 6 shows the DC corrected EMG signal and the derived RMS and ZC features as well as the NIRS feature from a sensor placed over the extensor digitorum during wrist extension. The RMS, ZC and NIRS features were calculated for a window size and increment of 256 samples.

Besides using a combination of individual features from the different signal types, the combined EMG and NIR sensor also offers the possibility of using a single feature integrating both the EMG as well as the NIR signal. One example is the NIRSRMS feature resulting from combining both the aforementioned NIRS as well as the RMS feature with the myoelectric signal $\overline{m} = \{m_1, ..., m_k\}$:

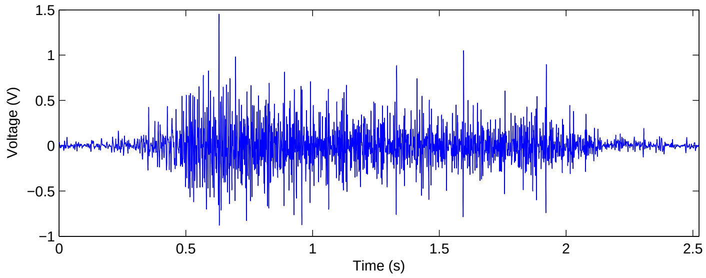
$$\text{NIRSRMS} = \text{RMS}(\overline{m}) \cdot \text{NIRS}(\overline{n}, \overline{e})\,. \tag{6}$$

Apart from the NIRSRMS combination, other features like the DC corrected NIRS signal – useful for realtime control – can be calculated from the NIR sensor data [6].

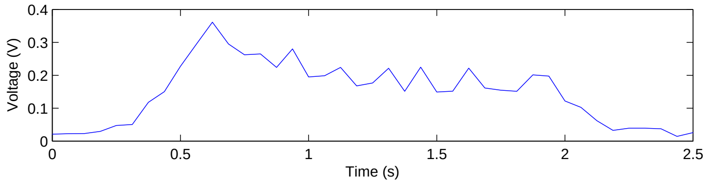
2.2. Training Set Creation

Out of the 20 data samples for each hand movement, 13 are drawn for training the classifier while the remaining 7 are deployed for the validation of the classification method. In the examplary classification process, we distinguish five different hand motions: fist, supination, pronation, wrist extension and wrist flexion.

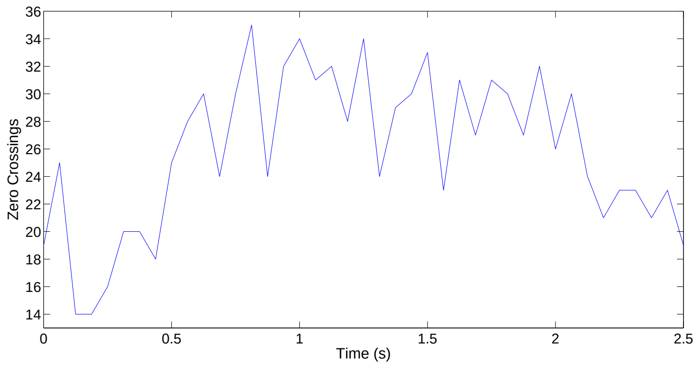
As a first classification method, we have chosen the decision tree algorithm from the MATLAB™ statistics toolbox. For each node t of the tree, a subset X_t



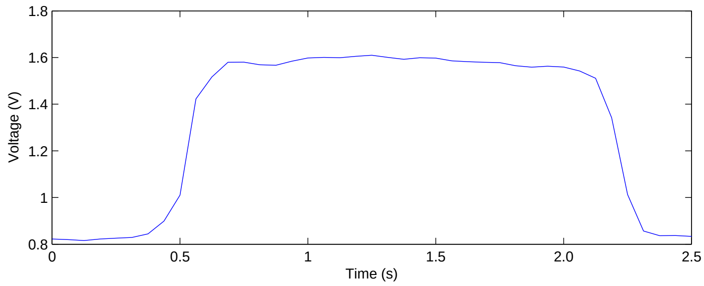
(a) EMG Signal



(b) RMS Feature derived from EMG



(c) ZC Feature derived from EMG



(d) NIRS Feature

Fig. 6. The EMG, EMG-calculated RMS, ZC and NIRS signal values for the extensor digitorum muscle during wrist extension.

is associated with it [14]. The subset is then split into two subsets for the descendant node, containing the 'Yes'-answers X_{tY} and the 'No'-answers $X_{tN}$for the question associated with the current node. The subsets satisfy:

$$X_{tY} \cap X_{tN} = \emptyset \,. \tag{7}$$

$$X_{tY} \cup X_{tN} = X_t \,. \tag{8}$$

The default splitting criterion used in MATLAB™ is the diversity index introduced by Gini for a node τ [8]:

$$i(\tau) = \sum_k p(k \mid \tau)^2 \,. \tag{9}$$

Altering the splitting criterion to other choices, offered by MATLAB™, did not yield a substantial increase in classification accuracy. Decision tree algorithms can quickly be implemented as the parameters are not critical. We have also investigated Support Vector Machines (SVM), which offer more flexibility. SVMs are linear classifiers which separate classes by means of hyperplanes. For a binary SVM, the hyperplane for a set of feature vectors x_i, with $i = 1, 2, ..., n$, which belong to the two classes ω_1 and ω_2, is denoted by [14]:

$$g(x) = \omega^T \cdot x + \omega_0 = 0 \,. \tag{10}$$

Multi-class SVMs can be constructed from binary SVMs by breaking up the original multi-class problem into several binary class problems [14]. The LIBSVM package employs the one-vs-one approach [3]. Depending on the type of training data, kernel choice and regularization constant can have an impact on the classification results of a SVM. Instead of a linear kernel, the authors of LIBSVM recommend the implemented RBF kernel, with:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \,. \tag{11}$$

A five-fold cross-validation was used for finding a suitable value for the regularization and γ parameters. To demonstrate the effects of feature selection and the choice of the classifier, the previously described classification algorithms were applied to different feature sets. Starting with a combination of the RMS values, extracted from the MES recording of the extensor digitorum muscle and the ZC feature derived from the sensor placed over the flexor carpi radialis, the feature values were first subjected to DC correction and noise reduction. All data points below a threshold of 0.15 for the normalized RMS and 0.68 for the normalized ZC feature set were removed before creating the classifier training sets. With these sets, models were generated for the training phase of the SVM and the decision tree classifier. Fig. 8 and Fig. 7 depict the partitioning of the source data into five classes. However, using the RMS-ZC combination, the classifiers cannot unambiguously distinguish between the five movements. While this feature combination is sufficiently distant for most of the classes,

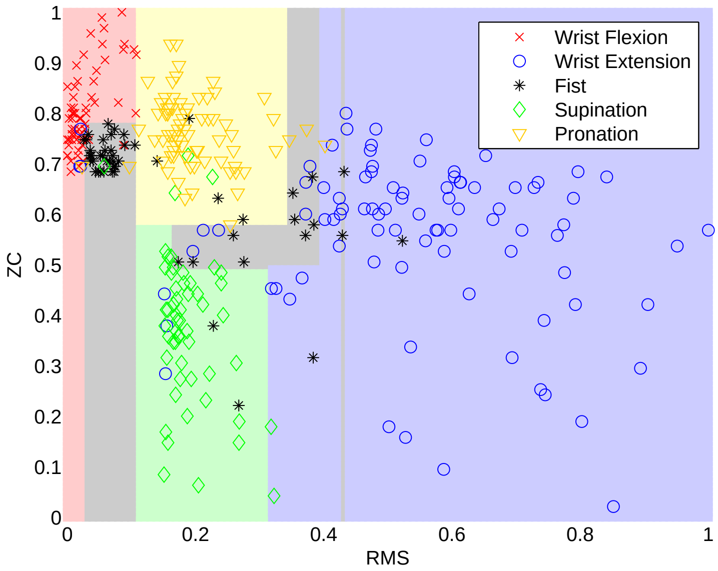


Fig. 7. Decision tree training set with 13 data samples for each class and combination of RMS and ZC features.

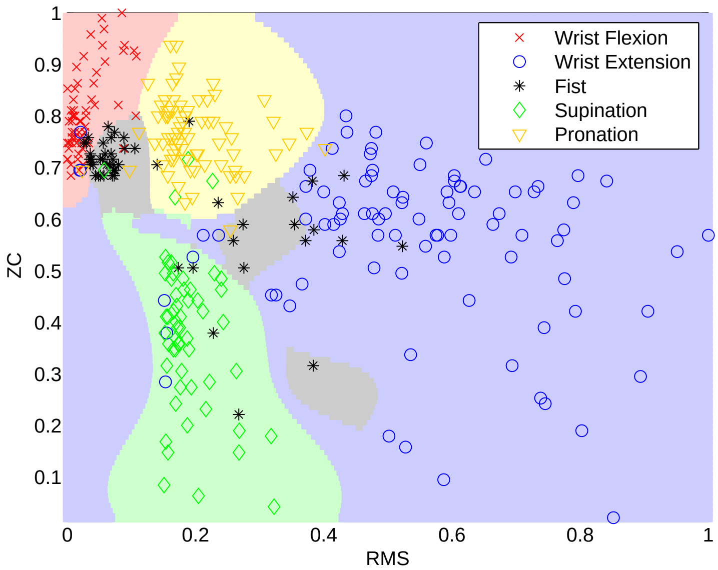


Fig. 8. SVM training set with 13 data samples for each class and a combination of RMS and ZC features.

closing a fist is not as clearly separated from neighboring classes such as wrist flexion or wrist extension.

In order to achieve better results, it is necessary to employ a different combination of features. Applying the same parameters for classifier training, Fig. 9 and Fig. 10 illustrate the results for the five classes with the RMS feature extracted from the data-sets for both the extensor digitorum and the flexor carpi radialis muscles. As evident from the location of the data points, this combination of features yields a clearer separation of the five hand movements. In this case - apart from the DC correction - additional noise reduction did not offer any further improvement of classification results.

Finally, for achieving even better distance between the data points of the motion classes, the RMS-RMS feature set was enriched with the NIRS feature data from both sensors, yielding a four dimensional feature space. Before training, both NIRS and RMS features were DC corrected. Then, the SVM and decision tree classifiers were trained again with this extended feature combination. The training models can now serve as reference for further classification. A validation as well as a visualization of the recording data is presented in the following section.

2.3. Validation and Visualization

In order to validate the classifier and its reference model, seven recordings of each hand movement were fed into the feature extraction process using the EMG and NIRS data. The derived features were then classified using the previously generated decision tree and the SVM models. Based on reference signals of known movements, classifier results were compared and validated. Table 1 contains the percentages of correctly identified hand movements for each class and the overall classification accuracy for the SVM and decision tree training models utilizing a RMS-ZC feature combination. Comparing the result of the RMS-ZC feature combination with the RMS-RMS combination in Table 2, the impact of feature selection prior to classifier training is confirmed. The validation results show an improvement between the RMS-ZC and RMS-RMS. Furthermore, the choice of the classification algorithm can have a substantial effect on accuracy as shown in Tables 1 and 2. Depending on the feature set, the simple decision tree algorithm may produce a variety of results, while the SVM classifier is more consistent. For this, parameters must be determined by cross validation in the training phase and initially set. Apart from classifier selection, our validation data demonstrates the value of the newly developed EMG-NIR sensor. In case of the selected five hand movement classes, 100% classification accuracy can be achieved by combining the recorded EMG and NIR data as presented in Table 3.

Finally, the simulation of a 3D-hand-model of a prosthesis was controlled with the classifier output. The visualization of a prosthesis is based on a model originally created in Autodesk™ 3ds Max [5]. The virtual hand, shown in Fig. 12, consists of components including shaft, wrist and joints for individual fingers as found in typical prostheses. For reduced complexity and better performance

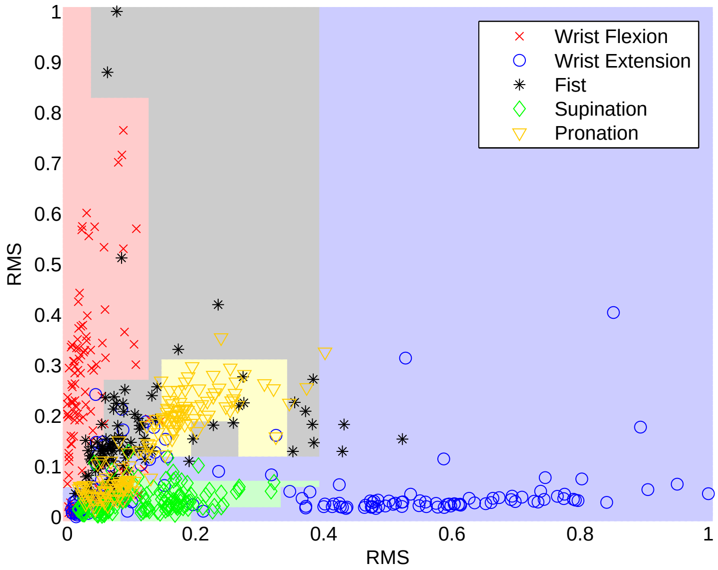


Fig. 9. Decision tree training set with 13 data samples for each class with two RMS features.

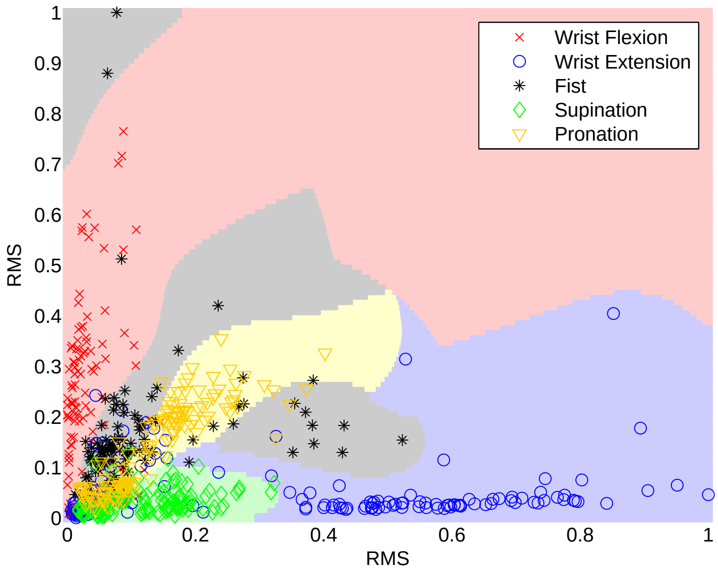


Fig. 10. SVM training set with 13 data samples for each class with two RMS features.

Table 1. Percentages of correct hand movements for the RMS-ZC feature set.

Hand Movement	SVM Model	Decision Tree Model
Wrist Flexion	85.7%	71.4%
Wrist Extension	100.0%	100.0%
Fist	57.1%	14.3%
Supination	100.0%	28.6%
Pronation	100.0%	100.0%
False Positives	2.9%	11.4%
False Negatives	8.6%	25.7%
Overall Accuracy	88.6%	62.9%

Table 2. Percentages of correct hand movements for the RMS-RMS feature set.

Hand Movement	SVM Model	Decision Tree Model
Wrist Flexion	100.0%	85.7%
Wrist Extension	57.1%	100.0%
Fist	100.0%	71.4%
Supination	100.0%	85.7%
Pronation	100.0%	100.0%
False Positives	8.6%	8.6%
False Negatives	0.0%	2.9%
Overall Accuracy	91.4%	88.6%

Table 3. Percentages of correct hand movements for the RMS-RMS-NIRS-NIRS feature set.

Hand movement	SVM Model	Decision Tree Model
Wrist Flexion	100.0%	0.0%
Wrist Extension	100.0%	100.0%
Fist	100.0%	100.0%
Supination	100.0%	100.0%
Pronation	100.0%	100.0%
False Positives	0.0%	20.0%
False Negatives	0.0%	0.0%
Overall Accuracy	100.0%	80.0%

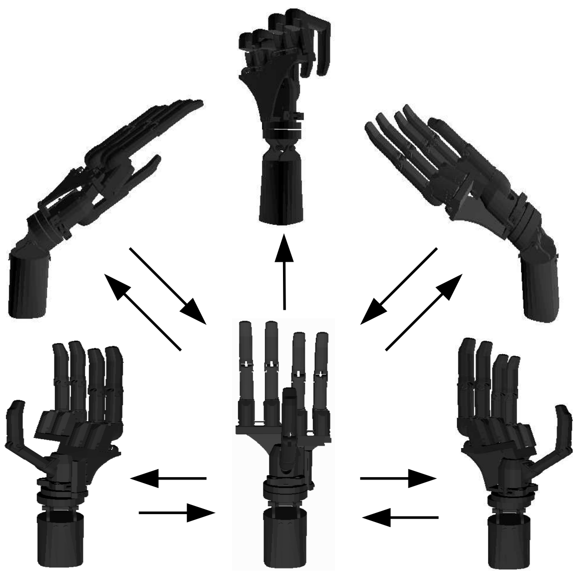


Fig. 11. 3D display of five hand movements (clockwise from bottom left to bottom right) starting and ending in the relaxed hand position (bottom center): pronation, wrist flexion, fist, wrist extension, supination.

during simulation, each finger consists of only two joints connected to a plate mounted on a rotary joint. Extending and flexing the 3D hand is realized with a pivoted joint at the base of the hand.

Fig. 12. The individual components of the 3D prosthesis hand model.

Listing 1.1. Setting hand positions for the 3D prosthesis and working with position files.

```
1  vp = VirtualProsthesis('prosthesis.WRL')
2
3  vp = vp.SetJointPosition('middle01', 90)
4  vp = vp.SetJointPosition('middle02', 90)
5  vp = vp.SetJointPosition('ring01', 90)
6  vp = vp.SetJointPosition('ring02', 90)
7  vp = vp.SetJointPosition('small01', 90)
8  vp = vp.SetJointPosition('small02', 90)
9  vp = vp.SetJointPosition('index01', 90)
10 vp = vp.SetJointPosition('index02', 90)
11 vp = vp.SetJointPosition('thumb01', 90)
12 vp = vp.SetJointPosition('thumb02', 90)
13
14 pose_fist = vp.GetHandPosition('fist')
15 vp = vp.SaveHandPosition(pose_fist)
16 vp = vp.WriteHandPositionsToFile('positions.mat')
17
18 vp = vp.ReadHandPositionsFromFile('positions.mat')
19 vp.GetSavedPositionNames()
20 vp = vp.LoadHandPosition('fist')
```

After conversion to the Virtual Reality Modeling Language (VRML) file format, the resulting file was integrated into the MATLAB™ environment. Several functions are now available for accessing the individual joints of the virtual prototype, allowing for the control of individual fingers. Because of this flexibility, the prosthesis can be used to simulate all hand movements recognized by the classification method. Fig. 11 provides screenshots of the virtual prosthesis displaying the five different hand motions. After instantiating the virtual prosthesis in MATLAB™, the position of the individual phalanges can be changed by entering the name and specifying the angle of the joint. Through combining simultaneous movements of several fingers, different hand-positions can be adopted. Listing 1.1 presents the code to set the position of the individual phalanges to assume a fist position. After setting the various joint angles necessary for simulating the desired hand-position, it is possible to assign a label to the position and save it. Several positions can be stored in a file for later reference. This way, the behavior of various prostheses can be captured and sets for various hand positions stored for quick retrieval. Lines 18 to 20 in Listing 1.1 show the process of accessing individual hand positions stored in a file.

Fig. 2 shows the 3D visualization of the five hand movements used for training the decision tree and SVM classifiers described in this contribution. The output of the classifier serves as control input to assign the desired hand-position to the virtual prosthesis model after a movement change from the initial resting hand position is detected.

2.4. MATLAB™ Movement Classification Toolbox

In order to support and accelerate the decision process for the selection of feature-extraction- and classification-methods, a toolbox for MATLAB has been developed. The aforementioned recordings of hand movements can automatically be subjected to feature extraction, classifier training and classifier validation. Both EMG as well as NIR sensor signals are supported with their corresponding features. Various parameters can be set for the individual steps in the classification process. Fig. 13 shows the main toolbox window containing three tabs. The selected first tab has options and dropdown boxes to choose and calculate the desired features from sensor signals. In the selection process, an arbitrary number of sensors as well as feature combinations can be chosen. Furthermore, it is possible to set the window size for the EMG and NIR features.

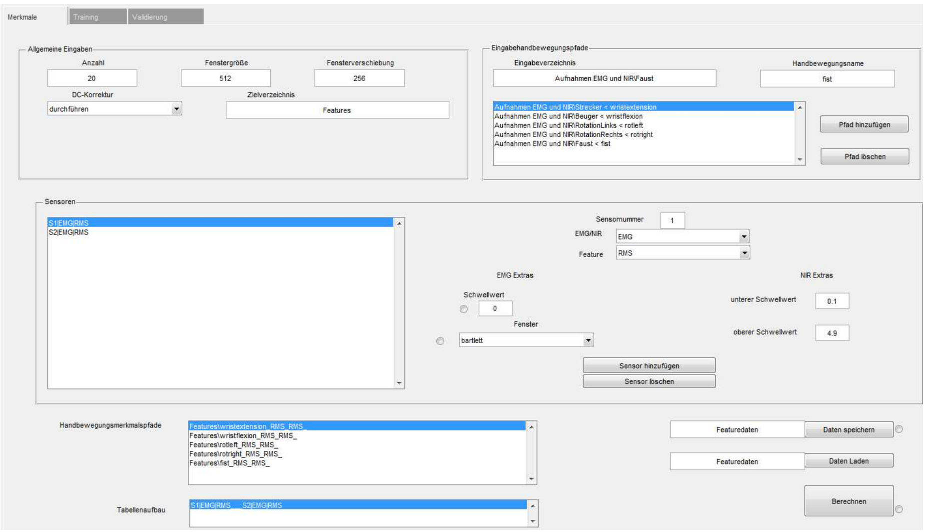


Fig. 13. Main window of the MATLAB toolbox for feature extraction, classifier training and validation.

The program equally offers a high number of parameters for selecting and training the implemented classifiers. The training- and the validation-mode allow to set a threshold to remove noise before feeding the data into a classifier. Furthermore, it is possible to plot selections of training data as well as classification maps. After validation, a detailed report about classification results for individual hand movements can be viewed and saved within MATLAB™. It is also possible to store the parameters used for feature calculation and training model generation. With this combined functionality, feature sets as well as different classifiers can quickly be compared, helping to choose methods and parameters for prosthesis control schemes. The program code has been modularized as far as possible to offer easy integration of new features as well as classifiers. If novel sensor systems become available, the toolbox can be extended to accommodate for new signal source types.

3. Results

This contribution discloses the modeling, validation and visualization of classification-based prosthesis control schemes. As an example for the individual steps necessary during the classification process, five different hand movements were distinguished using decision tree and SVM classifiers. After feature extraction and training set creation, the trained classifier was validated using existing MES and NIR recordings. The impact of feature and classifier selection is shown with four SVM and decision tree classifiers based on two different feature sets. The classification results were further improved by adding the NIR data from combined EMG-NIR sensors. In addition to the classification process, the behavior of a hand prosthesis is demonstrated through the control of a 3D visualization in MATLAB™ version 7.12.0. As a result, the entire process from training to functional validation and visualization can be seamlessly modeled in one application. Due to the considerable amount of feature extraction as well as classification methods, significant differences in classification accuracy mandate further research focusing on a systematic comparison of feature extraction and classification methods. Research efforts at our department so far resulted in the development of a toolbox for MATLAB™ which enables researchers to select, compare and adapt feature-extraction and classification methods. The current version of the toolbox supports several classification methods including decision trees and support vector machines as well as the extraction of various features from both EMG and NIR signals. Future editions will comprise additional feature calculation and classification algorithms. At the moment, only a limited amount of feature algorithms is available for NIR sensor data. Future research will focus on devising new NIR feature calculation methods. Furthermore, initial digital filtering of raw sensor data to remove noise and artifacts before feature extraction is introduced to increase classification results. Besides improvement of sensor signal processing, current and future research targets the extension of sensor capabilities. For example, the NIR sensor allows changing the area of observation by adjusting the distance between the LED and the photo resistor.

Acknowledgments. The authors thank Dr. Stefan Herrmann and Andrej Gehl for the design and the development of the 3D prosthesis for MATLAB™. Furthermore, we are grateful to Manuel Rosenau for creating a first collection of EMG recordings for testing the classifiers. Finally, we thank Marcus Eckert for his effort towards developing the MATLAB™ movement classification toolbox.

References

1. Arvetti, M., Gini, G., Folgheraiter, M.: Classification of EMG signals through wavelet analysis and neural networks for controlling an active hand prosthesis. In: Proc. IEEE 10th International Conference on Rehabilitation Robotics (ICORR 2007). pp. 531–536 (Jun 2007)
2. Buchenrieder, K.: Dimensionality Reduction for the Control of Powered Upper Limb Prostheses. In: Proc. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07). pp. 327–333 (Mar 2007)
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm
4. Englehart, K., Hudgins, B., Parker, P., Stevenson, M.: Classification of the myoelectric signal using time-frequency based representations. Medical Engineering & Physics 21(6-7), 431–438 (1999)
5. Gehl, A.: Modellierung einer Prothesenhand mit Matlab. Bachelor Thesis, Universität der Bundeswehr München, Neubiberg, Germany (Dec 2010)
6. Herrmann, S., Attenberger, A., Buchenrieder, K.: Prostheses Control with Combined Near-Infrared and Myoelectric Signals. In: EUROCAST 2011, Part II. LNCS, vol. 6928, pp. 602–609 (2011)
7. Herrmann, S., Buchenrieder, K.: Fusion of Myoelectric and Near-Infrared Signals for Prostheses Control. In: Proc. 4th International Convention on Rehabilitation Engineering & Assistive Technology iCREATe'10. pp. 54:1–54:4. Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre, Kaki Bukit TechPark II, Singapore (2010)
8. Izenman, A.J.: Modern Multivariate Statistical Techniques. Springer (2008)
9. Leon, M., Leija, L., Muñoz, R.: System for the Identification of Multiple Movements of the Hand. In: Proc. 3rd International Conference on Electrical and Electronics Engineering 2006. pp. 1–3 (Sep 2006)
10. Liu, Z., Luo, Z.: Hand Motion Pattern Classifier Based on EMG Using Wavelet Packet Transform and LVQ Neural Networks. In: Proc. IEEE International Symposium on IT in Medicine and Education (ITME 2008). pp. 28–32 (Dec 2008)
11. Peerdeman, B., Boere, D., Witteveen, H.J.B., Huis in 't Veld, M.H.A., Hermens, H.J., Stramigioli, S., Rietman, J.S., Veltink, P.H., Misra, S.: Myoelectric forearm prostheses: State of the art from a user-centered perspective. Journal of rehabilitation research and development 48(6), 719–738 (Aug 2011)
12. Phinyomark, A., Limsakul, C., Phukpattaranont, P.: A Novel Feature Extraction for Robust EMG Pattern Recognition. Journal of Computing 1(1), 71–80 (Dec 2009)
13. Reiter, R.: Eine neue Elektrokunsthand. Grenzgebiete der Medizin 1(4), 133–135 (1948)
14. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Academic Press, fourth edn. (Aug 2008)

Andreas Attenberger is a PhD student at the Institut für Technische Informatik, Universität der Bundeswehr München under the supervision of Prof. Klaus Buchenrieder. He received his diploma in media informatics from the Ludwig-Maximilians-Universitat Munchen in 2009. The focus of his research

lies on the improvement of natural control schemes for upper limb protheses. His interests include signal acquisition, processing and classification.

Klaus Buchenrieder is a full professor of Informatics at the Universität der Bundeswehr München, Germany. His research and teaching is in the field of embedded systems, reconfigurable system design and design automation. He received a Ph.D. and a M.S. degree in Computer Science from The Ohio State University in Columbus, Ohio. After graduation, he joined the Corporate Research laboratories of Siemens AG in Munich and later the Design Automation Department of Infineon Technologies AG. He holds numerous patents and is a honorary professor of the University of Tubingen, adjunct professor of the Computer and Electrical Engineering department at the University of Arizona and a professor of the Sino-German College of the Tongji University in Shanghai. He is also the founding chair of the Codes/Cashe workshop series on Codesign and of the Consyse workshops on conjoint engineering.

Received: June 1, 2012; Accepted: August 30, 2012.