

Learning EMG control of a robotic hand: Towards Active Prostheses

Sebastian Bitzer

Department of Computer Science
University College London
Email: s.bitzer@ucl.ac.uk

Patrick van der Smagt

Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
P.O. Box 1116, 82230 Wessling, Germany
Email: smagt@dlr.de
(corresponding author)

Abstract—We introduce a method based on support vector machines which can detect opening and closing actions of the human thumb, index finger, and other fingers recorded via surface EMG only. The method is shown to be robust across sessions and can be used independently of the position of the arm. With these stability criteria, the method is ideally suited for the control of active prosthesis with a high number of active degrees of freedom. The method is successfully demonstrated on a robotic four-finger hand, and can be used to grasp objects.

I. INTRODUCTION

The focus on the properties of prosthetic robotic hands is currently undergoing a slow but steady shift. Whereas the cosmetic aspect traditionally was and still remains an important issue for most patients, regained dexterity becomes more prominent as advances in mechatronic systems offer such possibilities. Consequently, most prosthesis manufacturers (e.g., Otto Bock, Motion Control, Liberating Technologies), besides their line of cosmetic hands, also offer a line of active hands. Such hands are controlled over myoelectric (EMG) interfaces, measuring the activity of the patient's muscles (viz. their motor units) in the lower arm.

Such active hands however suffer from low acceptance among patients with amputated hands. One reason for this is the limited dexterity that such hands offer to the patient: with only one active degree of freedom, the hand can only open and close on command, and can thus only be used for very coarse grasping and handling.

Whereas already many highly integrated prosthetic hands are under development or at production level (e.g., the DLR-HIT hand, the Cyberhand, the Fluid hand), control of such hands using noninvasive interfaces is still problematic and cannot be used on patients. Limitations lie with the number of fingers that can be controlled, session independence without retraining, and insensitivity to other arm muscle use, e.g., w.r.t. arm pronation or supination.

In this article we present a highly integrated approach to the use of EMG recording of the human lower arm in order to control the opening and closing of three fingers of the hand. Although user independence is problematic, we do attain optimal results with respect to session independence, allowing a patient to wear use the prosthetic device in the morning without having to retrain it. Also we demonstrate the ability

of using the device independent of the position of the arm. The method is demonstrated on a robotic four-finger hand, and can be used for grasping objects.

II. EMG DATA ACQUISITION/SETTING/EMG SETUP

A. EMG Hardware

The EMG data are recorded using ten commercial Otto Bock® active double differential electrodes 13E200=50 and a custom-built, low data-rate USB AD-converter for EMG data acquisition. Each electrode includes an amplifier with a factor of up to 100,000 changeable on a scale from 1 to 7 as well as a rectifier; in the experiments, the amplification is set to 4.

B. Properties and Problems of Recorded Signals

The surface EMG signals are influenced by several factors not related to finger movements [1]. All of these factors may influence the measured EMG signal and thus disturb the classification. We investigated most of these disturbances and discuss how we handle them.

Depending on the size of the muscle that is measured and its position relative to the electrode, the EMG signal will be weaker or stronger. Thus with different electrode placements the EMG amplitudes at maximum force production of underlying muscles may be very different.

On top of these between-electrode differences one can often see global changes in which the signal amplitude in all electrodes increases or decreases. Such changes can be due to muscle fatigue, a subjective change in the perception of the force produced during a finger action, or due to external influences related to electromagnetic pollution.

However, probably the most challenging property of the surface EMG signal with respect to a robust classification is that it changes with arm posture, because underlying muscles especially towards the wrist change their position relative to the electrodes on the skin when the hand is rotated.

C. Finger Movements and Corresponding Electrode Positions

In the following we describe the automatic detection of extension and flexion of the thumb, the index finger, and the middle, ring and little fingers together (hereafter “remaining”). We will refer to these six movements as “finger actions” to highlight their binary character.

For each finger action, a separate muscle or set of muscles is responsible. We distribute our ten electrodes over these muscles as follows. One electrode is placed above the extensor pollicis brevis and longus for thumb extension, two above the flexor pollicis longus for thumb flexion, one above the extensor indicis for index extension, two above the distal part of the flexor digitorum superficialis for index flexion, two above the extensor digitorum for remaining extension, and two above the proximal part of the flexor digitorum superficialis for remaining flexion.

III. LEARNING HUMAN FINGER ACTIONS

In our approach to the control of active prostheses we try to minimize the effort for the patient. We consider it of key importance that the system adapts itself to the user, not the other way around. Consequently, we apply state-of-the-art machine learning techniques to discriminate user-specific finger actions. In this section we present our classification system which mainly consists of a set of Support Vector Machines.

A. Preprocessing

The Otto Bock electrodes output an amplified, filtered, and rectified signal. The AD converter we use sends new data whenever the value of at least one electrode changes. With this setup we get a sampling rate of about 35Hz on average, with a maximum of up to 125Hz when values in many electrodes are rapidly changing.

Nevertheless it is still hard to classify single samples and therefore we always concatenate five samples to a sequence, representing one data point. Thus we are classifying data points extending over the last 140ms on average.

Furthermore we normalize training data points to have mean zero and a standard deviation of one in each dimension or electrode, respectively. This largely eliminates the between-electrode differences mentioned above in the training data set, but because we use the so estimated normalization constants for testing, too, some of the differences still might recur in the normalized test set and have to be taken care of by the classifier.

B. Support Vector Machines for Finger Actions

Support Vector Machines (SVMs) [2] have proven to be powerful classification tools [3]. They linearly separate two data sets by placing a hyperplane between them such that the distance of both data sets to the hyperplane is maximally large. By the clever usage of kernels the algorithm can be extended to non-linear separation without significantly increasing calculation costs.

In their original formulation SVMs can separate only two data sets which does not suffice for our application. Adapted methods have been proposed to the combination of several SVMs to a multi-class classifier [2]. We decided to use a one-versus-rest approach in which one SVM per class is trained to separate that class against all others. Input data is classified by that class giving the highest classification result.

C. Posterior Probabilities

In the standard formulation of SVMs the classification result is interpreted as binary, i.e., an SVM either assigns the one class (1) to a data point or the other (-1). This is not suitable in the multi-class case when two one-versus-rest SVMs signal a positive classification, because then it is not possible to discriminate between them.

Fortunately, the true classification result of an SVM is proportional to the distance of the data point to the separating hyperplane and it can be assumed that, with increasing distance, the affiliation to the corresponding class is more secure. In [4] a method is proposed with which such SVM classification results are mapped to the posterior probabilities

$$P(\text{class} = c | \text{SVM} = x) \quad c \in \{-1, 1\}, x \in \mathbb{R},$$

computing the likeliness that a data point belongs to a certain class. Accordingly, we apply this method to all our SVMs and thus obtain a better and particularly more distinctive ground for deciding for one of six finger actions.

D. Accepting a Classification

If the maximum of the posterior probabilities for the six finger classes is above a certain threshold (in our case, we used 0.95 which worked well in practice) we accept this class as classification result. Any classification below that threshold is discarded and the previously obtained class is used.

This fallback mechanism has turned out to be very helpful to increase the robustness of the system, because it is capable of bridging subsequences of data points for which classification is difficult and thus prevents false classifications which is especially important within a sequence of data points corresponding to a single finger action.

IV. CLASSIFICATION RESULTS

With the described classification system we achieve very good results with respect to robustness in several aspects. In the following we will show that the six finger actions can be discriminated with different arm postures and across sessions between which the electrodes were replaced on the arm, even after several days.

Our results are reported in terms of performance p . We define performance as the relative frequency of correctly classified data points

$$p = \frac{\# \text{ correct data points}}{\# \text{ data points}}.$$

Thus a performance of 1 would be perfect. Nevertheless it is never achieved which is also due to inaccuracies in the registration of classes for training and test data. Such inaccuracies occur as slight shifts between the end of a finger action and the beginning of a new finger action. Accordingly we define a tolerable error as an error which lies in a window of 300ms from the borders of a finger action to another one and where the error consists of wrongly classifying one finger action as its neighbour. For our system we find that a performance of over 0.9 with a relative frequency of tolerable

errors to overall errors (short: tolerable error ε_t) exceeding 0.3 is acceptable for our application. Note also that errors are expressed in terms of single data points and that they often occur only as short sequences with 10 to 20 data points corresponding to an error length of not more than 300ms.

A. Relax

We first classify the six finger actions in EMG data which was recorded while the lower arm was in a relaxed position (relax). We learn¹ on one such data set consisting of 3,000 data points and test on another data set of the same size which was recorded during the same session.

In 15 such tests our classifier gives results with $0.89 \leq p \leq 0.97$. The mean performance is $\bar{p} = 0.94$ with a standard deviation of 0.025. For the tolerable error these values are similarly good with $\bar{\varepsilon}_t = 0.62 \pm 0.17$ (mean \pm std).

We present an example classification result in figure 1. With $p = 0.92$ and $\varepsilon_t = 0.51$ this represents a moderately good example, but it still has the very nice property that a sequence of data points corresponding to a single finger action is not interrupted by an error. All errors occur either at the beginning or the end of a finger action. This classification continuity is of great importance, because a user of the system, for example, would not want to lose hold of a glass when drinking. Also note that in this example all finger actions have eventually been correctly recognised while the subject had no feedback to adjust his actual movement.

B. Pronation

Rotation movements of the lower arm or hand, respectively, pose a challenge for robust classification of finger actions. Especially downward rotations of the palm (pronation), e.g. for typing tasks, are a problem, because these movements have strong effects on amplitude characteristics of some electrodes.

To quantify the problems with pronation we learned² the classifier on a relax data set and tested it on a pronation data set from the same session in accordance with above. The average performance for 8 such tests is $\bar{p} = 0.74 \pm 0.1$ with $\bar{\varepsilon}_t = 0.16 \pm 0.11$. Accordingly the resulting classifiers are almost useless, because single finger actions are interrupted or some finger actions are even never recognised.

By combining relax and pronation data for training it should be possible to learn pronation patterns, too. We once recorded 16 data sets within one session consisting of 12 relax sets and 4 pronation sets. We trained the system on all combinations of one relax and one pronation data set and tested its resulting performance on the remaining data sets.

Figure 2 shows these performances separately for relax and pronation test data for each pronation training data set averaged over the different relax training sets. It can be seen that the performance on relax data ($\bar{p} > 0.94$, $\bar{\varepsilon}_t > 0.45$) is excellent independent of which pronation set is used for training, but although training with three out of four pronation sets results in usable to very good classifiers ($0.90 \leq \bar{p} \leq 0.94$,

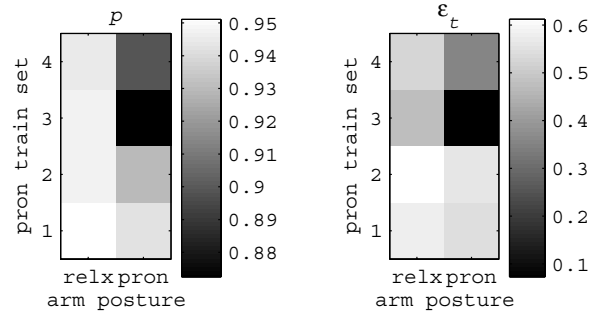


Fig. 2. Performances for learning a combination of one relax and one pronation data set from a single session averaged over different relax training sets. Y-axis: pronation data sets used for training. X-axis: test sets with different arm posture, relx – relax, pron – pronation. For relax test data $p > 0.94$ and $\varepsilon_t > 0.45$. For pronation test data pronation set 3 has $p = 0.87$ with $\varepsilon_t = 0.07$, but learning with other pronation sets is better with $p > 0.9$ and $\varepsilon_t > 0.35$

$0.36 \leq \bar{\varepsilon}_t \leq 0.56$), the system learned with pronation set 3 still has problems on unseen pronation test data ($\bar{p} = 0.87$, $\bar{\varepsilon}_t = 0.07$).

The session from which we report here was one of the first sessions we recorded. After its recording we made adjustments to electrode positions which improved classifiability of the data. Nevertheless, one can see that even for this session it is possible to achieve good classification results while the lower arm is pronated.

C. Session Independence

From session to session, it is impossible to place electrodes at the exact same position. In our data we see that electrodes which are lying exactly next to each other, c.q. record data from positions on the arm not more than 1.5cm apart, can have very different amplitude characteristics. Thus, even moderate displacements of electrodes can lead to sensitive changes in amplitude patterns.

Under the assumption that a large body of training data contains the most important amplitude patterns, we trained³ our classifier on one relax and one pronation data set from four sessions, and tested its performance on a remaining session. We did this in turn for all of five test sessions and so implemented a cross-validation procedure.

While we get excellent performance on almost every trained session (mean \pm std, relax: $\bar{p} = 0.95 \pm 0.03$, $\bar{\varepsilon}_t = 0.55 \pm 0.21$, pronation: $\bar{p} = 0.94 \pm 0.04$, $\bar{\varepsilon}_t = 0.41 \pm 0.20$) we also achieve high performances on the untrained data which can be seen in figure 3. There the performances on almost all data sets are partly even clear above 0.9. However some problems occur in the two remaining data sets. The fact that in these two cases the performances are acceptable when the arm is in a different posture suggests that at least one electrode has been shifted too far away from its designated position which is at least partly corrected by a rotation of the arm.

¹SVM parameters: linear Kernel, $C = 0.1$

²linear kernel, $C = 0.1$

³rbf kernel, $\sigma = 7$, $C = 20$

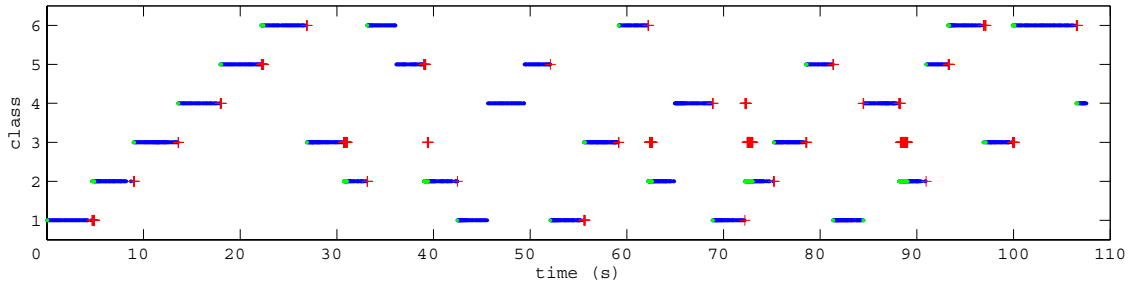


Fig. 1. Example of classification result for one data set. Single data points are shown. Blue dots: correctly classified data points. Green dots: correct class of data points falsely classified as indicated by red pluses. $p = 0.92$, $\varepsilon_t = 0.51$ Longest error: 760ms. No ongoing finger action is interrupted by errors. Example would be perfectly usable in an online test.

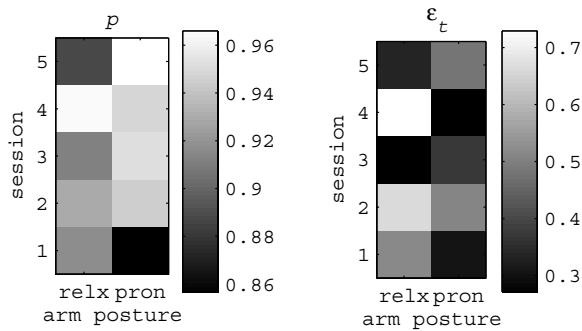


Fig. 3. Cross-validation performance for multiple session learning. Y-axis: session left out for testing, x-axis: data sets with different arm posture, relx – relax, pron – pronation. All pronation data sets have $p > 0.94$ except those in session 1 ($p = 0.86$). For session 5 $p = 0.89$ on relax data sets, all others have $p > 0.91$. Lowest ε_t is 0.27 (session 3 relx, session 4 pron). For all other data sets $\varepsilon_t > 0.3$.

This result shows that session independence is achieved, but if the differences between trained and untrained sessions are too large, for example because electrode displacements are too large, the system has to be retrained to adapt to these changes.

V. APPLICATION TO A ROBOTIC 4-FINGER HAND

The method described above has been used for the control of the DLR four-finger hand II [5]. Since we are only interested in the control of three fingers, the fourth finger has been configured to move parallel with the third finger.

The DLR Hand II is a four-finger hand with thirteen active DoF: three per finger, and one for opposition of the thumb. In the four identical fingers, the motion of the third phalanx is coupled to that of the second. The actuation system consists of brushless DC motors, tooth belts, harmonic drive gears and differential bevel gears in the base joint. The differential joint allows the use of full power of the two actuators for flexion or extension, thus allowing the use of optimally small motors, obtaining 30N force at the finger tip. The high joint speed of over $360^\circ/s$ allows for high finger speed, important for, e.g., ball catching. Besides force and position sensors, we use joint torque sensors and specially designed potentiometers in each

joint. Furthermore, a tiny six-dimensional force/torque sensor is used in the finger tip.

A simplified version of this hand, targeted at the prosthetic market, is available as the DLR/HIT hand [6], [7].

In the EMG experiment, all used phalanges of each robotic finger move with equal speed and displacement, so as to obtain natural motion patterns when opening or closing the fingers or the whole hand. Measured and classified muscle activity is transferred to the robot hand, which then follows a predefined trajectory to open or close the finger.

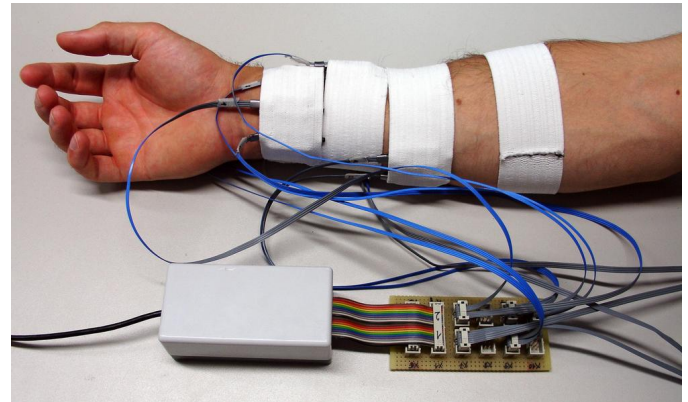


Fig. 4. Placing the electrodes on the arm.

Figure 4 shows the placement of the electrodes on the arm. A total of ten electrodes is used in the experiments, in order to obtain optimal results even when the arm is in supination.

Example use of the interface is demonstrated in figure 5.

The EMG interface can be well used to grasp objects. Since no force feedback is possible using this interface, the patient can use her visual feedback to interact with the object via the prosthetic hand (see figure 6).

VI. CONCLUSIONS

We have obtained unparalleled results with the classification of finger movements using surface EMG on the human lower arm. Apart from obtaining session independence, thus making the system optimally suited for every-day use by amputees, the system is insensitive to pronation and supination of the

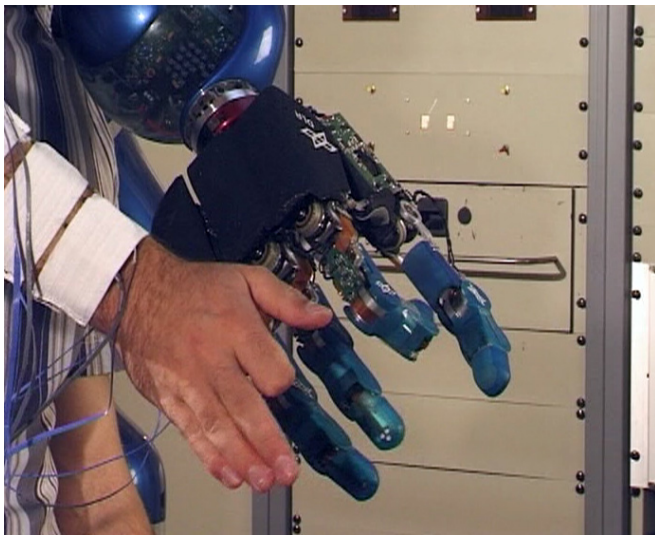


Fig. 5. The robotic four-finger hand controlled with EMG interface.

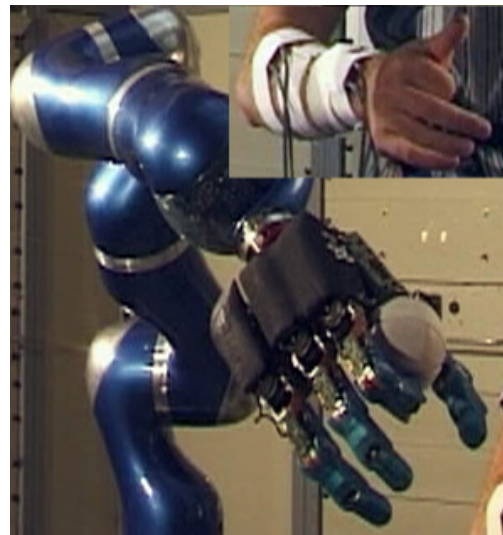


Fig. 6. Grasping an object using the EMG interface.

lower arm, allowing the opening and closing of three fingers (thumb, index finger, and other fingers).

Crawford et al. [8] also use an SVM-based approach for EMG classification with 7 electrodes, distinguishing 8 classes. Even though their immediate results are very good, the robustness of the system between sessions without retraining is not demonstrated and half of their classes depend on the arm being pronated or supinated, thus making the system less useful for a prosthetic application.

Corresponding with the position of the relevant muscles in the lower arm, correct placement of the electrodes on the arm remains a key ingredient to obtaining high success rates. As is common practice in the fitting of commercially available active prosthesis, a professional is required to perform this step, including the training of the system.

Future investigation of the system may include the use of a cuff fitted with a large array of electrodes, worn on the lower arm. Whereas the system proposed in this article uses 10 active electrodes with a market value of several hundred euros each, an array of electrodes would be differently constructed, where the signal amplifications would not occur on-site. Such a cuff would be universally fittable, so that only training sessions have to be done before the system can be used.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Goekhan Bakir whose support on vector machines was key to this paper.

REFERENCES

- [1] C. De Luca, "The Use of Electromyography in Biomechanics," *Journal of Applied Biomechanics*, vol. 13, no. 2, pp. 135–163, 1997.
- [2] B. Schoelkopf and A. J. Smola, *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [3] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
- [4] J. C. Platt, *Probabilistic outputs for support vector machines and comparison to regularized likelihood methods*. MIT Press, 2000, pp. 61–74.
- [5] J. Butterfass, M. Fischer, and M. Grebenstein, "Design and Experiences with DLR Hand II," in *Proceedings of the World Automation Congress*, 2004.
- [6] P. He, M. Jin, L. Yang, R. Wei, Y. Liu, H. Cai, H. Liu, N. Seitz, J. Butterfass, and G. Hirzinger, "High Performance DSP/FPGA Controller for Implementation of HIT/DLR Dexterous Robot Hand," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 3397–3402.
- [7] R. Wei, X. Gao, M. Jin, Y. Liu, N. Liu, N. Seitz, R. Gruber, and G. Hirzinger, "FPGA based Hardware Architecture for HIT/DLR Hand," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3233–3238.
- [8] B. Crawford, K. Miller, P. Shenoy, and R. Rao, "Real-Time Classification of Electromyographic Signals for Robotic Control," in *Proceedings of AAAI*, 2005, pp. 523–528.