

Surface EMG in advanced hand prosthetics

Claudio Castellini · Patrick van der Smagt

Received: 22 December 2007 / Accepted: 16 October 2008 / Published online: 18 November 2008
© Springer-Verlag 2008

Abstract One of the major problems when dealing with highly dexterous, active hand prostheses is their control by the patient wearing them. With the advances in mechatronics, building prosthetic hands with multiple active degrees of freedom is realisable, but actively controlling the position and especially the exerted force of each finger cannot yet be done naturally. This paper deals with advanced robotic hand control via surface electromyography. Building upon recent results, we show that machine learning, together with a simple downsampling algorithm, can be effectively used to control on-line, in real time, finger position as well as finger force of a highly dexterous robotic hand. The system determines the type of grasp a human subject is willing to use, and the required amount of force involved, with a high degree of accuracy. This represents a remarkable improvement with respect to the state-of-the-art of feed-forward control of dexterous mechanical hands, and opens up a scenario in which amputees will be able to control hand prostheses in a much finer way than it has so far been possible.

Keywords Learning and adaptive systems · Rehabilitation robotics · Physical human-robot interaction

This work is partially supported by the project NEURObotics, FP6-IST-001917.

C. Castellini (✉)
LIRA-Lab, University of Genova, viale F. Causa, 13,
16145 Genova, Italy
e-mail: claudio.castellini@unige.it

P. van der Smagt
Institute of Robotics and Mechatronics, German
Aerospace Center/DLR, Oberpfaffenhofen, Germany
e-mail: smagt@dlr.de

1 Introduction

In the framework of hand prosthetics, one usually operates the distinction between passive and active hand prostheses. As the name suggests, an active hand prosthesis (AHP) is a hand prosthesis which can be voluntarily actuated, to some degree, by the patient wearing it. Besides being cheap, visually appealing, lightweight and long-running, the ideal AHP is highly dexterous and easily controlled. Sensorial feedback completes the picture, making the prosthesis a sensible replacement of the lost hand.

At the time of writing, however, the state-of-the-art of AHPs is far from this scenario. The best known commercially available AHPs are Otto Bock's SensorHand [Otto Bock \(2008b\)](#) and Touch Bionics's i-LIMB [i-Limb \(2007\)](#). The SensorHand is a classical one-DOF "claw", proportionally controlled typically by one or two electromyography electrodes; the i-Limb has five independently moving fingers plus a passively opposable thumb, but only uses two electrodes and, as far as one can understand, offers no fine control over single fingers or over the required amount of force. Nevertheless, one can see a definite move forward as far as the mechatronics is concerned—a drive which mainly comes from miniaturised electronics and humanoid robotics; examples of this are, e.g. the DLR prosthetic hand ([Huang et al. 2006](#), see Fig. 1) and the [CyberHand \(2007\)](#).

It seems then that the problem of *control* by the patient is going to be a major issue in the next years. As the prosthetic hand becomes more and more flexible, how is the patient supposed to precisely command the prosthesis what to do? Operating a hand requires a fine control, possibly down to the level of the single fingers: first of all, presented with a certain task such as turning a door handle or grabbing a car key, the patient must be able to enforce the correct grasping type; this involves the activation of some joints only, and in

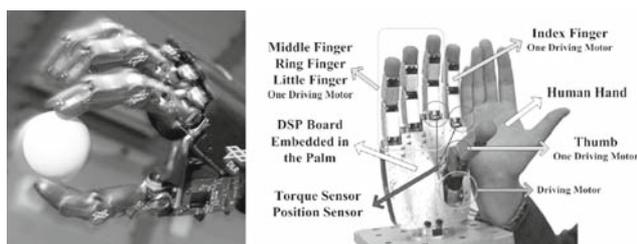


Fig. 1 Left the DLR Hand II. Right the DLR prosthetic hand

particular positions. Second, the amount of force involved in the grasp must be controlled, so that it is possible to grab, e.g. both a hammer without letting it slip and an egg without breaking it. Third, feedback to the patient is paramount.

As far as the feed-forward path is concerned, that is, sending commands to the prosthesis, two types of interfaces between the patient and the prosthesis have been developed or are being studied: *invasive* and *non-invasive*. The former gather control signals directly from the user's nervous system, either via brain implants or surgical use of electrodes. Quite obviously, invasive interfaces are supposed to deliver a high signal quality, since the signals can be gathered exactly in the right spots; but they involve surgery and all related sterility (and psychological) issues. On the other hand, non-invasive interfaces are easier to handle and maintain, but require a much better signal conditioning, since they usually work with surface (skin) signals or vision and gaze tracking.

In the context of non-invasive interfaces for controlling mechanical hands, a concrete possibility arises from *forearm surface electromyography* (EMG), a technique by which muscle activation potentials are gathered by electrodes placed on the patient's forearm skin; these potentials can be used to track which muscles the patient is willing to activate, and with what force. Surface EMG is therefore, in principle, a cheap and easy way of detecting what the patient wants the prosthesis to do.

Using the EMG to feed-forward control an AHP requires adaptivity, accuracy and speed: each single patient must be able to control the prosthesis accurately in real time. These characteristics belong to *machine learning*, which has in fact already been applied to EMG in the classification of isotonic hand postures of healthy subjects [Ekvall and Kragić \(2005\)](#); [Bitzer and van der Smagt \(2006\)](#); but so far, no indication about the amount of force involved in the grasping act is detected.

In this paper we go one step beyond by applying machine learning techniques to the forearm EMG signal generated by a healthy subject in controlled conditions. Three machine learning approaches have been compared: (a) a simple feed-forward neural network with one hidden layer, (b) a Support Vector Machine with radial basis function kernel [Boser et al. \(1992\)](#), and (c) Locally Weighted Projection Regression+

[Vijayakumar et al. \(2005\)](#). Our analysis benefits from a simple but effective procedure for selecting a subset of the samples on-the-fly, called *Online Uniformisation* (OU); and in the end it shows that there is no clear winner among the tested approaches, but that, as a whole, the idea is viable. The resulting system, in fact, guesses *both* (a) what kind of grasp the subject is applying, and (b) how much force the subject is exerting. The system attains remarkable accuracy: the type of grasp can be reconstructed with an average accuracy of $89.67 \pm 1.53\%$, and the applied force can be predicted with an average percentage error of $7.89 \pm 0.09\%$, meaning 4.5 N over a range of about 57 N.

Last, the system has been really tested on-line on the DLR-II Hand (see [Butterfass et al. 2004](#)), which is a dexterous, multifingered robotic hand, although not a prosthetic one. The system was able to control it in impedance, to the point of grasping an egg without breaking it, and then to actually break it, when the subject would tighten the grip.

All in all, the results in this paper lay the basis for the feed-forward control of the next generation AHPs. The last big step is to check whether an amputee's forearm still contains enough muscular potential activity to obtain the same results. This is the main subject of future research, along with feedback, to better close the sensorimotor loop between the patient and the prosthesis.

The paper is structured as follows: after a brief review of relevant literature, we describe in detail the experiment and the methods used to tackle it (Sect. 2); then we show and comment on the experimental results (Sect. 3); last, discussion and conclusions are presented in Sect. 4.

1.1 Related work

Machine learning has already been used for hand posture classification using the EMG signal, at least in [Ferguson and Dunlop \(2002\)](#); [Tsuji et al. \(2003\)](#); [Bitzer and van der Smagt \(2006\)](#). In the latter work, in particular, as many as *nine* different postures could be classified to a remarkable degree of accuracy; it is deemed that this was possible since the hand postures would correspond to isotonic and isometric muscular configurations, i.e. precise force configurations. The EMG signal is known to be related to the *force* a muscle is applying (see, e.g. [De Luca 1997](#)); therefore, if one wants to reconstruct the hand position one must resort to classification performed in controlled force conditions. On the other hand, as far as we know, nobody has ever attempted to build a map from the EMG to the force the fingers apply—rather than classification, a *regression* task.

OU is based upon the concept of sparsification of a function, meaning that only a subset of the samples in the training set are used to build an approximation to a target function. This is required in an online setting, since there is no guarantee that the flow of data potentially usable for training will

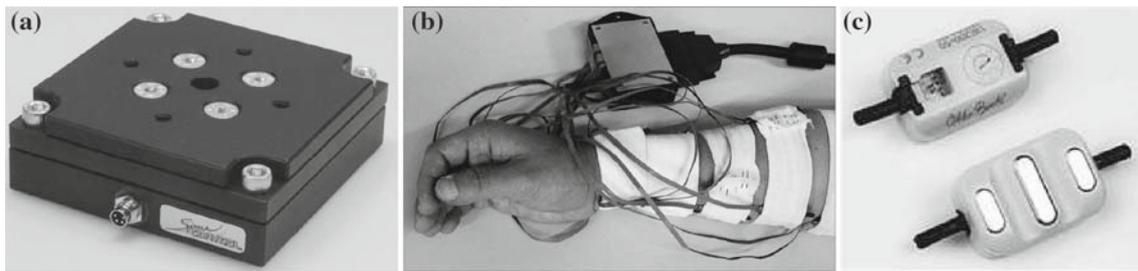


Fig. 2 The experimental setup. **a** The SpaceControl OFTS force/torque sensor, large face up. **b** The arm of the subject with the EMG electrodes fitted and held in place by elastic bands. Electrode cables are wired in a box and then directed to a National Instruments

PCI-6023E analogic/digital conversion card (not shown). **c** An Otto Bock 13E200=50 surface EMG electrode, with the amplification gauge (*upper part* of the figure) and the three metallic contacts (*lower part*)

ever cease. Other ways of reducing the number of support vectors have been tried, e.g. in Orabona et al. (2007), in which sparseness is achieved without losing any accuracy, or, e.g. Lee and Mangasarian (2001); Keerthi et al. (2006) where a subset of the support vectors is heuristically selected and fewer support vectors are traded for larger classification/regression errors. Indeed, the idea of sparsifying a solution is not new and has been used, e.g. in a Bayesian framework Figueiredo (2003); as well, the sparseness of solutions in the framework of Support Vector Machines has been exploited and improved, e.g. with Relevance Vector Machines Tipping (2000).

2 Materials and methods

In this section we describe the experiment we have conducted and the methods we have employed to gather the data, filter and analyse them.

2.1 Experimental setup and design

2.1.1 General setup description

The experiment consisted of freely, and repeatedly grasping a SpaceControl OFTS force/torque sensor OFTS (2007) orthogonally to its large face (see Fig. 2a). Four different ways of pressing were allowed: opposing the thumb and index, the thumb and middle, the thumb and ring or the thumb and all other fingers. The speed and force were intentionally left to the subject's will. Four force-sensing resistors (FSRs) were applied on the subject's hand fingertips (thumb, index, middle, and ring), in order to be able to detect which grasp type was used at each instant of time. At the same time, *ten* forearm surface EMG electrodes were applied to the subject's forearm, held in place by elastic bands, in order to gather information about the muscle activation (see Fig. 2b).

Numerical data from the EMG electrodes, FSRs and OFTS were gathered at the fastest sampling rate we could obtain, that is, 256 Hz, using a National Instruments DAQ PCI-6023E analog/digital conversion card NIDAQ (2007), mounted on a fast PC equipped with Windows XP.

2.1.2 EMG signal and electrode placement

The *ten* EMG electrodes were applied to the subject's right forearm, held in place by elastic bands. The electrodes were double-differential Otto Bock 13E200=50 models (OttoBock 2008a; see Fig. 2c), each one gifted with an amplification gauge ranging from 2000 to 100,000 times. Initial qualitative experiments revealed that a safe setting for the amplification gauge was in the middle of the range, corresponding to about 14,000 times. This is in agreement with the EMG signal amplitude predicted in the related literature (see, e.g. De Luca 1997), that is about $100\mu\text{V}$ on average: the voltages our DAQ card read ranged from 0 to about 2.5 V.

Six of the electrodes were placed in pairs along the lower face of the forearm, whereas four of them were applied in pairs on the upper face. The initial positioning of the electrodes was chosen in order for them to lie approximately on top of the muscles which elicit finger movements; the precise placement was done following the description in Bitzer and van der Smagt (2006), which proved to be optimal for Support Vector Machine classification of hand postures.

As far as the EMG signal is concerned, it must be remarked that it is subject to remarkable changes depending on, at least, four orders of factors:

1. *Inter-subject variability*. All forearms are different from one another in shape, size and power.
2. *Arm posture*. Besides finger movements and grasping, the forearm muscles are also involved in the motion of the arm. The EMG signal is therefore likely to change if the forearm is moved during signal acquisition, for example when switching from pronation to supination, or simply

while walking around. Even raising the shoulder to lift the forearm from the table will result in remarkable signal changes.

3. *Electrode displacement.* The intensity and quality of the EMG signal depends upon a correct placement of the electrode over a muscle. In principle, each electrode should be placed over a single muscle, precisely on top of the muscle belly, halfway the length of the muscle, and always exactly in the same place. Displacing the electrodes will alter the signal, and beside that, a precise placement is essentially impossible when dealing with surface forearm EMG.
4. *Muscle fatigue.* As the muscles are used more and more, continually, fatigue changes the RMS of the EMG signal, calling for continual adaptation, at least over a reasonable set of different fatigue conditions.

Problems 1 and 2 have been for now neglected by concentrating on one subject only: male, aged 35 and fully able-bodied, instructed to keep the arm still and relaxed on a table in a comfortable position, with the palm orthogonal to the plane of the table. See the discussion section for more about these issues.

As far as muscle fatigue and electrode displacement are concerned, electrodes *cannot* be expected to exactly lie in the very same position every time the prosthesis is used; moreover, in a preliminary round of experiments, muscle fatigue was clearly perceived by the subject during the experiment. In this framework, the only possibility to overcome these problems is to explicitly take them into account, gathering enough data to be able to train the machine under different conditions of electrode displacement and muscular fatigue.

We then organised the experiment as follows: the subject was instructed to continuously grasp the sensor over a period of time of 3–4 min; then he was allowed to rest for about 2 min. This was called a *session*. It was expected that muscle fatigue would appear already during one session. Three sessions were gathered without taking the elastic bands off the subject's forearm, in order *not* to have electrode displacement within such a set of sessions, that we called a *group*. After each group, the electrodes and bands were removed and the subject was allowed for a much longer period of rest, ranging from half an hour to 1 hour. During resting in-between groups, the subject could get back to his normal muscular activity.

Five groups were then gathered during one day; and this entire procedure was repeated during another day. This procedure would allow us to examine a relevant amount of data, gathered along a relatively long period of time and under different conditions of muscle fatigue (within one session) and electrode displacement (between groups). Spectral analysis of the EMG signal revealed that all relevant information is limited to 10 Hz (damping of -30 dB at that frequency),

therefore sampling at 256 Hz proved to be a large overshoot. We will employ this fact later on.

2.1.3 Force applied during the grasp

The OFTS force/torque sensor would output a (negative) integer numerical value ranging from 0 to about $-5,000$, expressed in (negative) fiftieths of a Newton. After normalisation, the range of the applied force would then be between 0 and 100 N, with a resolution of 0.02 N. Linearity of the sensor is guaranteed, and was anyway manually verified.

2.1.4 Type of grasp

The voltage values output by the four FSRs applied onto the subject's fingertips were monitored in order to understand which kind of grasp the subject was applying to the sensor. A threshold was experimentally decided, above which the finger would be defined *in contact* with the sensor. Using this technique, for each instant in time one of five possible categories was established: 0, no action; 1, grasp by opposing the thumb and index finger; 2, opposing thumb and middle; 3 thumb and ring; and last 4, grasp by opposing the thumb and all other fingers.

It must be remarked here that the EMG signal would be altered immediately at the onset of finger movement, which our setup was unable to detect. This would result in potentially wrong EMG values for category 0. Moreover, the FSRs have been experimentally determined to suffer from a remarkable hysteresis effect, that is, they will indicate slightly different voltage while pressing and releasing; this is due to small rubber ends glued on top of the sensor surfaces, which aid grasping by raising the static friction coefficient. Hysteresis is also supposed to somehow degrade the quality of the learning. Because of these factors we would never expect a close-to-100% classification accuracy, nor a perfect reconstruction of the applied force. A better setup is currently being studied, which would avoid these effects. Again, see the discussion for more about this issue.

2.2 Learning methods

The gathered data were analysed both for classification and regression. *Classification* is the process by which one wants to assign a label to each sample in the input space, whereas in *regression* the target is a real-valued function of the values of the input samples. All machine learning methods we examined need to be first trained on a set of points in the input space for which the target (label or force value) is known; this set will be called *training set*. Then, in order to verify that the obtained models are good, they are tested on a separate set of points, the *testing set*. In all our experiments, the training and

testing sets were disjoint, and fivefold cross-validation was employed to check the generalisation power of the methods.

Taking into account the considerations of the previous section, we set the input space to be \mathbb{R}^{10} , that is, one coordinate for each EMG electrode; therefore, $\mathbf{x}_i \in \mathbb{R}^{10}$, $i = 1, \dots, l$. In the case of classification, each category representing a grasping type would be represented as an integer value, that is, $y_i \in \{0, \dots, 4\} \subset \mathbb{N}$, $i = 1, \dots, l$. In the case of regression, the force value would be directly encoded as a real number, that is, $y_i \in \mathbb{R}$, $i = 1, \dots, l$. Before training, all samples in the training set were normalised, as is customary, by subtracting the mean values and dividing by the standard deviation, for each input space dimension. Testing was done by normalising with the mean and standard deviations found during training, as one should not assume to know any a-priori statistics whatsoever about the testing samples. No filtering whatsoever was applied to the input signal, in order to have a more realistic, delay-free result.

2.2.1 Neural networks

Feed-Forward Artificial Neural Networks (FFN; see, e.g. Bishop 1995 for a comprehensive introduction) are probably the most popular machine learning algorithm nowadays available for both classification and regression. For our experiment we strived to keep the FFN as simple as possible. With ten inputs, we chose a feed-forward neural network with one hidden layer with *ten* units with hyperbolic tangent (sigmoidal) activation function. *Five* Outputs are used for classification, each unit representing one category, and one unit in the output layer for regression, the unit representing the target force value. The network was trained using gradient descent, with backpropagation being enforced via the quasi-Newton algorithm (for classification) and the Levenberg-Marquardt algorithm (for regression). The training phase was stopped either after 100 epochs or when the validation set would show an increasing error. For each experiment, we repeated the training phase 10 times and then gathered the best model found, in order to overcome the well-known problem of local minima. Validation was done on a disjoint subset of the training samples.

2.2.2 Support vector machines

Support Vector Machines (SVMs; see, e.g. Boser et al. 1992; Burges 1998; Cristianini and Shawe-Taylor 2000) are machine learning methods able to determine the best candidate function for a classification or regression problem, drawn from a functional space induced by the choice of a binary function between points in the input space, $K(\mathbf{x}_1, \mathbf{x}_2)$, with $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{10}$ in this case. K is called *kernel*. In the most general setting, the function found is

$$f(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \quad (1)$$

where $b \in \mathbb{R}$, whereas the $\alpha_i \in \mathbb{R}$ are Lagrangian coefficients obtained by solving a minimisation problem whose cost functional is guaranteed to be convex. Because of this, SVMs do not suffer from the problem of local minima. We employed a well-known freely available SVM package, *libsvm* v2.83 Chang and Lin (2001), in the Matlab wrapped flavour.

As is quite standard in the related literature, we employed a Gaussian kernel, so that in general

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$$

We used fivefold cross-validation to determine the generalisation error of each model, and then grid-search to find the optimal values of the hyperparameters σ and C (a constant involved in the cost functional minimisation problem).

2.2.3 Locally weighted projection regression

Locally Weighted Projection Regression (LWPR, see Vijayakumar et al. 2005) is a regression method especially targeted for high-dimensional spaces with redundant and irrelevant input dimensions. It employs locally linear models, each of which performs univariate regressions in selected directions in the input space. It has a computational complexity that is linear in the number of inputs, but due to its incrementality it can take long time to train (as we verified it was the case). Therefore we used it for regression only, and trained it with the uniformisation procedure.

We used the latest stable C version of LWPR, kindly made available by Stefan Klanke, wrapped in a Matlab command interface. We chose to use the Radial Basis Function kernel and meta-learning, and then performed fivefold cross-validation and found the initial values of the distance metric for receptive fields by grid search.

3 Experimental evaluation

In this section we describe the experimental analysis. In particular, we first describe some preliminary results obtained in batch fashion, which have led to the development of OU. We then show a detailed comparative analysis of the three approaches selected, first on the classification problem, and then on the regression problem, in which OU is used.

3.1 Batch experiments

For a set of preliminary experiments, we first focus on SVMs applied to the grasp type classification problem. The simplest idea is that of checking whether data obtained during

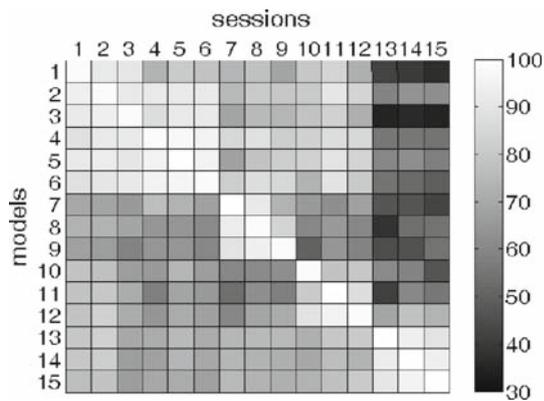


Fig. 3 Accuracy matrix for day 1. Accuracy for non-diagonal elements is $73.23 \pm 14.29\%$

one session can be used to build a model able to generalise over other sessions during the same day. Such a result would indeed mean that less than 3 min of data gathering are potentially able to control the prosthesis for a long time—at least one day, if not more.

In order to check this hypothesis, we trained a SVM for classification over each single session, for the first day. Sessions are numbered chronologically during the day, sessions 1, 2, 3 forming group 1, sessions 4, 5, 6 forming group 2, and so on (with a slight abuse of language, we will call the model obtained by training a machine on session i , *model i*). We then tested each model on all sessions of the first day, obtaining therefore an *accuracy matrix* A in which A_{ij} would be a percentage denoting the correctly guessed labels when testing model i on session j . This is a *cross-session analysis*. The accuracy matrix A for day 1 is visible in Fig. 3.

As one can see, the hypothesis fails: the average accuracy attained on non-diagonal elements is about 74%, dropping down to about 30% in the worst cases (of course, we are not interested in diagonal elements of the matrix, which represent accuracies obtained by testing on data on which the models have been trained). One cannot expect to correctly drive a prosthesis if one sample in four is misclassified. Notice that, however, a significant “group accuracy” phenomenon is present: in the matrix, good accuracy values are obtained on 3×3 sub-matrices located on the diagonal, corresponding to cross-session accuracy for sessions *belonging to the same group*, that is, when the elastic bands were not removed and no electrode displacement was present. A simple explanation of this is that electrode displacement shifts the samples in the input space, causing a decrease in performance. This explanation will be tested later on.

Another problem is that there are simply too many samples to train upon. The total time of data gathering was about 100 min; at 256 Hz, that means about 1.6 millions samples, an unfeasibly large training set for any of the examined methods, not to mention SVM classification. Even restricting

a training set to a single session, this would result in about 53, 000 samples, which not only is still too large, but probably contains redundant and irrelevant information. Moreover, in the real setting, that is on-line, this number is doomed to grow continually and cannot therefore be used as it is for periodical re-training. A smart sample reduction strategy is needed, in order both to overcome this problem, and also to gain insight into the EMG signal in general.

3.1.1 Batch uniformisation

The simple idea behind batch uniformisation is that, in a real-life set-up such as ours, there can be many input samples located in the very same region of the input space, with very similar target values. One obvious case is that of label 0, indicating no ongoing grasping: it is intuitively expected that a large number of samples will be taken in that region of the input space, since the subject will be in the 0 condition for a longer time than all other labels.

Since all functions involved in the experiment are due to human motion, it seems reasonable to assume that they are continuous and derivable up to any arbitrary order; therefore, it makes little sense to consider samples obtained in a non-uniform way such as that described above: if samples are too close to each other (according to a suitable notion of distance in the input space) then the value of our target functions should be similar, and non-parametric learning systems such as SVMs should be able to take it into account.

Batch uniformisation consists of removing, from a training set, those samples which are *too close to each other*, according to a suitable notion of inter-sample distance. In order to take into account the variance of each single electrode, and since in batch data analysis all data are available beforehand and therefore all possible statistics can be gathered a-priori, we have decided to adopt Mahalanobis’s distance as the inter-sample distance. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{10}$; then the Mahalanobis distance between \mathbf{x}_1 and \mathbf{x}_2 is defined as

$$MD(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2)}$$

where Σ is the 10×10 covariance matrix, evaluated on the training set. $MD(\mathbf{x}_1, \mathbf{x}_2)$ is a measure of distance independent of the (co)variance of the electrodes (notice that if Σ is replaced by the identity matrix, $MD(\mathbf{x}_1, \mathbf{x}_2)$ is reduced to the usual notion of Euclidean distance).

Since checking the inter-sample distance on a batch of samples obviously takes a quadratic time with respect to the number of samples, which was infeasible, we adopt an approximated method which removes most, but not necessarily all, samples which are too Mahalanobis-close to each other. After a few initial experiments, the threshold distance was set at 1. Obviously, no *testing* set is uniformised. Notice, further, that applying uniformisation results in training sets which are considerably smaller than the original ones, up to

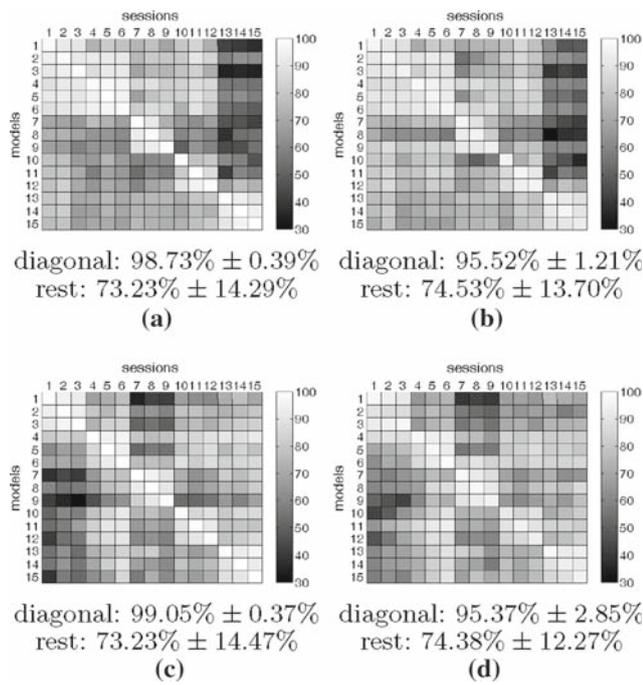


Fig. 4 Cross-session analysis and evaluation of the batch uniformisation procedure. **a** and **b**, accuracy matrices for day 1: **a** full models, **b** uniform models. **c** and **d**, same for day 2

about 100 times smaller, in fact dramatically decreasing the time required to train and test.

Consider now Fig. 4. The Figure presents more cross-session accuracy analysis, but this time both for the first and second day, and both with full and uniformised training sets (if the training set of a model has been uniformised, we will call the model *uniform*).

Consider first panes (a) and (b) of the Figure, pane (a) being the accuracy matrix for full models, day 1, and pane (b) being the accuracy matrix for the same day, but using uniform models [actually, pane (a) is the same matrix seen in Fig. 3]. It is apparent that uniform models attain a slightly better accuracy on non-diagonal elements, if compared to the full models. In fact, the accuracies are $73.23 \pm 14.29\%$ for the full models, and $74.53 \pm 13.70\%$ for the uniform ones. The same analysis for day 2 yields analogous results [consider the same Figure, panes (c) and (d)].

From this we conclude that the uniformisation procedure greatly reduces the training set size (and training and testing times) without degrading the performance. This is apparent from the fact that uniform models are slightly more accurate on testing sets which are disjoint from the training sets. Uniform models generalise slightly better, at least in this case.

3.1.2 Classification

Let us go more in detail, as far as the first day is concerned (Fig. 4 again, pane b): one can see that the first six models

(trained on the first two groups) obtain a quite good accuracy on the first six sessions (first two groups) whereas their accuracy rapidly degrades as more sessions are tested for. This is probably due to the first two groups having been gathered in similar conditions, very similar electrode positions and/or similar movements performed by the subject. On the other hand, sessions in the last group (columns 13, 14 and 15 of the matrix) are particularly hard, except when tested by models obtained from the last group itself—here the effect is probably motivated by the opposite reason: during those sessions, the subject must have explored different parts of the input space. This is corroborated by the fact that models 13, 14, 15 perform rather well on *all* sessions, if compared to other models (check rows 13, 14, 15 of the matrix). In other words, sessions 13, 14, 15 contain more relevant information than the others. Analogous considerations can be made by inspecting the accuracy matrix of the second day, pane (d) of the Figure.

From this analysis we confirm (recall the previous subsection) that *electrode displacement plays a determinant role* in the classification accuracy. Notice that muscle fatigue seems not to enter the picture, but this is reasonable since it is present already within one single session and the machine correctly takes it into account during the training phase. Notice once again that the uniformisation procedure does not hinder the generalisation power of the system. Electrode displacement present between groups (but not within a group) causes the samples in a group to be “shifted” in the input space, so that testing on a different group results in poor performance.

If this claim is correct, then there should be (negative) correlation between distance and accuracy. In fact, cross-session accuracy is highly correlated to the *average minimum inter-sample distance* between sessions. More in detail, let S_i and S_j denote the sets of samples gathered during sessions i and j ; then the *cross-session distance matrix* D is such that

$$D_{ij} = \frac{1}{|S_j|} \sum_{s_j \in S_j} \min_{s_i \in S_i} \|s_j - s_i\|^2$$

Essentially, D_{ij} denotes how far away in the input space the samples in S_j are from the samples in S_i . Note that D is in general not symmetric. The cross-correlation coefficient evaluated between D and the cross-session accuracy matrix is about -0.61 *both* for the first and the second day, indicating a strong negative correlation. Further experiments have revealed that this happens for Neural Networks too (cross-correlation -0.40 for day 1 and -0.51 for day 2); and also, that D is strongly *positively* correlated to the MSE in regression, for all the studied approaches: $0.62/0.78$ (day 1/day 2) for SVMs, $0.64/0.72$ for FFNs and $0.77/0.81$ for LWPR. In other words, the larger the distance of S_i and S_j , the worse the performance of model i tested on session j , *both in classification and in regression, and for all approaches tested.*

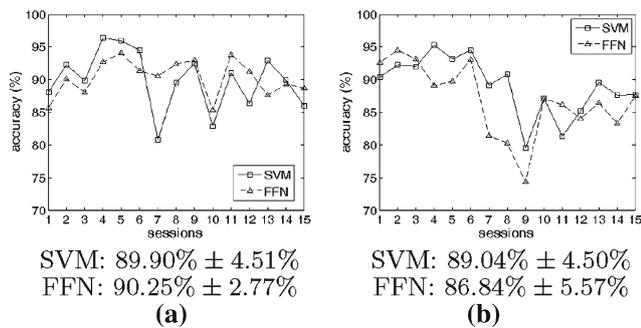


Fig. 5 Classification accuracy of best models, day 1 **a** and day 2 **b**

This tells us that (a) samples of the same group are closer to each other than samples from different groups, therefore electrode displacement causes displacement in the input space too; and (b) that this causes bad inter-group performance. “Samples far away from the training set will be predicted badly.”

The dual consideration is that if we train upon the “right” data, the accuracy should become acceptable. How to find the right data then? In this batch phase, we have decided to adjoin two models per day, which would obtain a good accuracy on all sessions. For instance, consider Fig. 4 again, pane b. It is apparent that model 4 performs well on groups 1 to 4, whereas model 13 does well on group 5 (and is not bad on the others). These two models were used to form a “best” training set which would give good results on the whole day 1. Analogous considerations led us to use also models 4 and 8 of day 2. The obtained model will be called *best* model.

This procedure was repeated for each problem tackled (classification, regression) and approach tested (SVM, FFN, and LWPR). Figure 5 shows the classification accuracy of the best models for classification on all sessions of day 1 and 2, for SVMs and FFNs. The analysis detailed in the previous subsection has been repeated for the Neural Network. In that case, models 8 and 15 of day 1, and models 3 and 10 of day 2 have been used to build the best model.

As one can see, there is no clear winner between SVMs and FFNs. FFNs perform slightly better on day 1 (higher mean, lower standard deviation) but SVMs are analogously better on day 2. All in all, classification accuracy is good, at an overall rate of about 90%. In this case, the training data amounts to four sessions (uniformised in the case of SVMs and full in the case of FFNs), which is about 12–15 min of user activity. But notice, that samples gathered during both days were necessary to have an idea of which sessions to use.

3.1.3 Regression

Last, the most interesting part was how to predict the amount of force applied by the subject by looking at the EMG signal. To do this, we have repeated once again the analysis done in

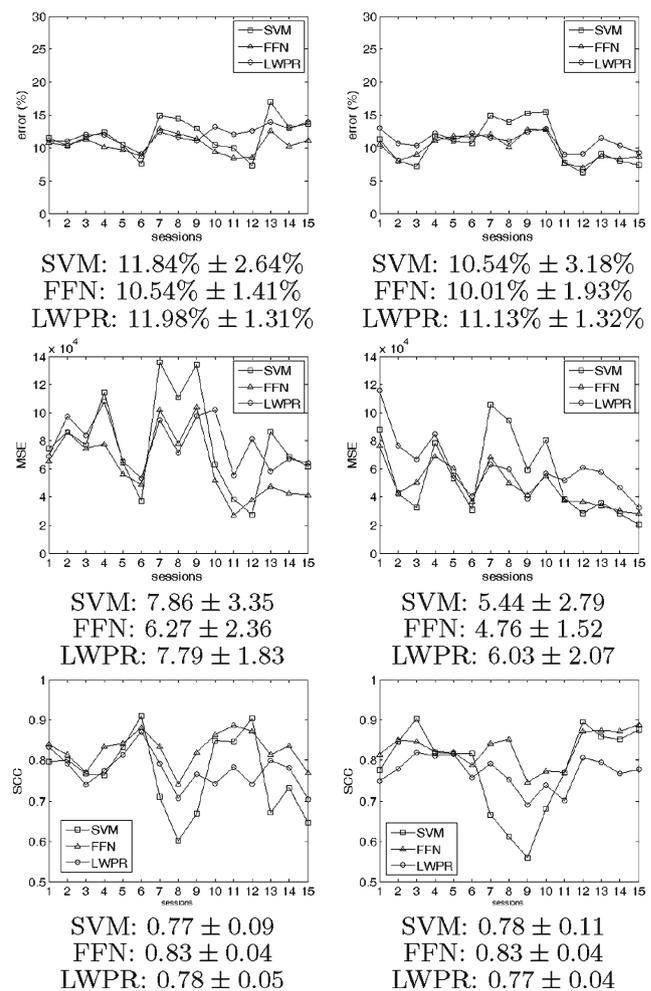


Fig. 6 Regression accuracy of best models, day 1 (left panes) and day 2 (right panes). First row normalised root MSE; second row mean squared error; third row, squared correlation coefficient

Sect. 3.1 for the three approaches selected, and found out that the four sessions involved in the best models were 6, 12, 3, 12 for SVMs, 4, 11, 3, 12 for FFNs and 6, 13, 3, 4 for LWPR. We have considered three indices of performance: the mean squared error (MSE) in its standard definition; the normalised root MSE (NRMSE), ratio of the square root of the MSE and the range of the target values, expressed as a percentage; and the squared correlation coefficient (SCC) between the predicted target and the real target.

Figure 6 shows the results; left panes are for day 1 and right panes for day 2. Consider the first row, plotting the NRMSE for each session: as it is apparent, as it was for classification, there is no clear advantage of one approach over another. FFNs perform slightly better as far as the NRMSE is concerned, which is probably the most interesting measure of performance, when moving to a real setting. Their error is on average $10.54 \pm 1.41\%$ and $10.01 \pm 1.93\%$. But as well, both LWPR and SVM perform quite well, their average errors ranging from 10.54 to 11.98%.

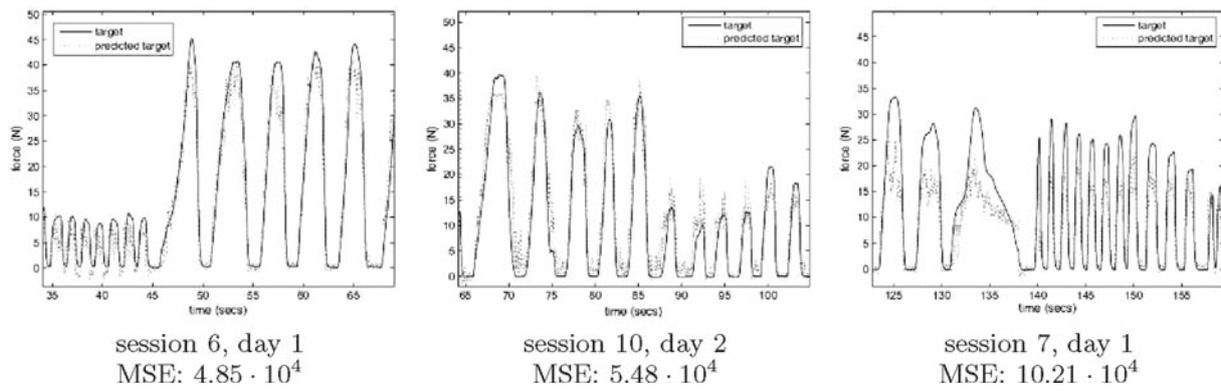


Fig. 7 Examples of real and predicted force target values

Consider now the second and third rows of the Figure. First of all there is a clear inverse correlation between the MSE and the SCC, as expected. Second, it is once again clear that the generalisation performance strongly depends on which data we have used to train the machines: consider for instance the MSE attained by SVM on day 1 (Fig. 6, second row, left pane): the best model was trained upon data coming from sessions 6 and 12, although uniformised, and not surprisingly those are the sessions for which the MSE is minimum; the same effect is present for the other approaches.

Last, in practical terms: the best average MSE obtained by FFNs ($6.27 \times 10^4 \pm 2.36 \times 10^4$ and $4.76 \times 10^4 \pm 1.52 \times 10^4$) corresponds to, in turn, an average error of 5N and 4.36N. Figure 7 shows some samples of the force values obtained from the OFTS, along with the corresponding values predicted by the best approach, that is, FFNs. As one can see, despite the non-perfect correspondence of the two curves, the FFN definitely follows the real target to a remarkable degree of accuracy, for a wide range of frequencies of the pressing/releasing action. The Figure shows data taken from three different sessions, in decreasing order of performance.

A remarkable point in the Figure is the presence of a “plateau” effect, especially in the test with the worst performance, namely session 7 of day 1 (third pane). As is apparent, the predicted targets are mostly wrong in amplitude, being systematically *lower* than the correct values. Again, this is most likely due to insufficient sampling in the region of interest.

3.2 Online experiments

The considerations of the previous Section lead to the reasonable hypothesis that inter-sample distance is the key. Keeping training sets uniform will result in smaller sets which still contain all information required for training. Moreover, bad generalisation is correlated to inter-set distance; therefore uniformisation can be as well used to retain samples which are far away from the current training set. We have then implemented an online version of the uniformisation

procedure, in order to test what would happen in a real setting while the patient is freely moving around.

First of all, since (see Sect. 2.1.2) the bandwidth of the EMG signal is limited in our experiment to about 10Hz, data have been subsampled from 256 to 25Hz, getting to a total training set of about 153, 000 samples. The samples are naturally chronologically ordered, so that they could be fed to the system one by one as it would actually happen during continuous acquisition of data from the patient’s activity.

Furthermore, in an online setting no a-priori statistics about the samples can be assumed. Therefore, as a measure of inter-sample distance, we dropped the Mahalanobis distance (which requires a good estimate of the covariance matrix) and resorted to Euclidean distance, defined in the standard way. Normalisation was still used, as it is essential for most machine learning methods, but the mean value and standard deviation of the training sets were evaluated on-the-fly without keeping the whole sets and re-evaluating them each time. Testing samples were also normalised according to these statistics (we also tested the batch of samples for dimensionality reduction using PCA, but found that no more than 2 or 3 dimensions could be eliminated. Therefore we dropped the idea, also since PCA would require, again, an estimate of the covariance matrix of the data set, which is not available online).

The OU procedure, then, works like this: we initially fix a minimum inter-sample distance d and start with an empty training set S . Then each time a new sample \mathbf{x} is available, we check whether $dist(\mathbf{x}, \mathbf{x}_i) \leq d$, for at least one $\mathbf{x}_i \in S$: if this is the case, then \mathbf{x} is discarded; otherwise, it is added to S .

The first question is: does OU give us an acceptable accuracy *at all times*? That is: does S constitute a good training set, as the patient explores new regions of the input space, and more and more samples are seen? In order to answer this question we considered again the problem of SVM classification, and let S grow according to the OU procedure. Then, every about 1.5min of sampled data, that is every 2, 400 samples, we trained the SVM on S and checked its accuracy

Fig. 8 **a** Classification accuracy of an SVM, as the training set grows according to the OU procedure. **b** Size of the online uniformised training set: number of samples (*upper plot*), fraction of the whole training set (*lower plot*)

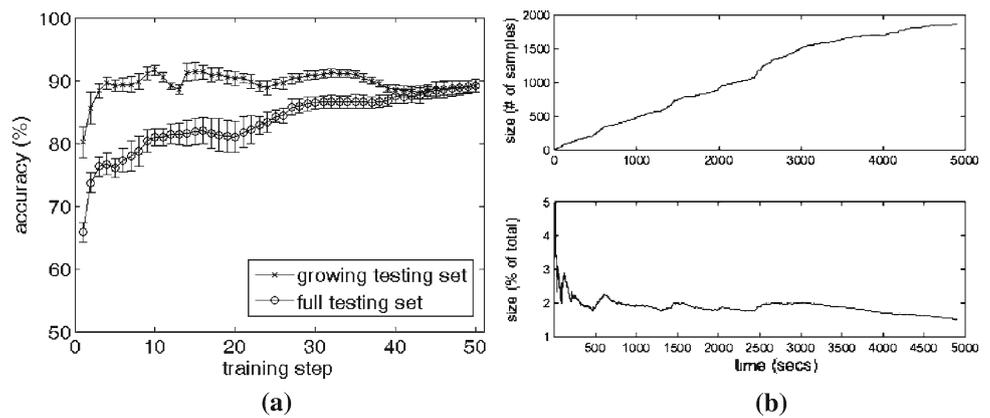
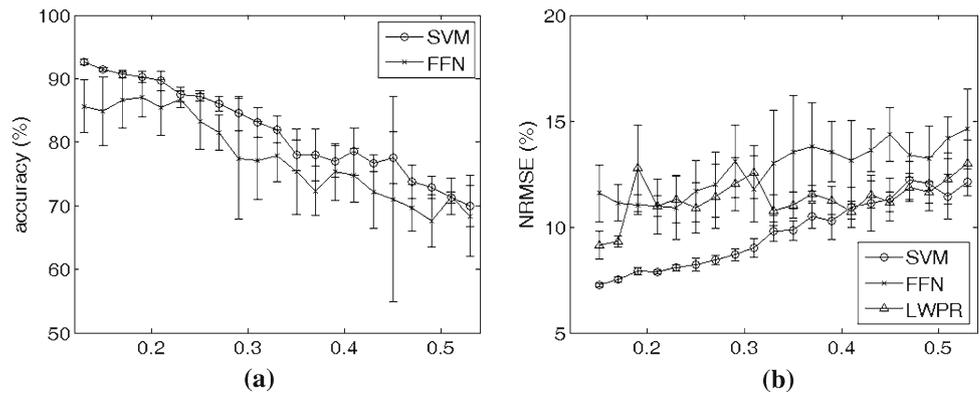


Fig. 9 Classification and regression results using the OU procedure, as d is increased. **a** Classification, **b** regression. Compare with Fig. 10 for the training set sizes



on a testing set drawn from the previously seen samples (but not in S , of course). This was done five times with different splits of S , so to obtain a statistically meaningful measure of accuracy. We chose to use a mid-range value of $d = 0.21$ obtained from initial experiments, which would result in a final training set of about 1,800 samples; moreover, we used hyperparameters $C = 10^{1.5}$ and $\sigma = 10^{0.5}$, found by grid search during the preliminary experiments. Figure 8 shows the results.

Consider first the “growing testing set” curve in pane (a) of the Figure: it is apparent that, already after four training steps, that is after some 6 min, the system can classify with an accuracy of about 90%, as it was the case in the preliminary analysis (accuracy 89.57 ± 0.94). Notwithstanding some oscillations, the accuracy remains substantially constant over the whole test and, at the end, is still 89.14 ± 1.05 . Consider now the “full testing set” curve, representing the accuracy obtained by the same models but on the *whole* testing set: now the system is being tested on samples drawn from zones of the input space it has not yet seen; and, as one would expect, the accuracy steadily increases, and it finally catches up with that obtained by testing on the growing testing set.

Consider now pane (b) of the Figure: the upper plot shows the size of the online uniformised training set as the sample acquisition proceeds; the lower plot shows the same curve as a fraction of the full training set size. As time goes on,

the OU procedure is letting the uniform training set grow less and less; the fraction of the full training set (lower plot) becomes smaller and smaller, being around 1.5% at the end.

From this we conclude that (a) OU is keeping the training set remarkably small in absolute terms, and smaller and smaller percentage-wise, as more and more data are acquired; (b) OU is “letting in” only relevant information, since the accuracy is uniformly high if tested on a growing testing set, and ever growing if tested upon a full testing set. In other words, the OU procedure is effective in building a compact and accurate training set for SVM classification.

As far as other approaches and problems are concerned, Fig. 9 shows an all-inclusive set of results for all problems and approaches considered, and for various values of the minimum distance threshold, d , which was fixed at 0.21 in the previous experiment.

As d is increased from 0.13 to 0.53, all approaches show a decreasing performance, as expected: in classification, the SVM is uniformly better, going from 92.61% for $d = 0.13$ to 70.01% for $d = 0.53$. The standard deviations for the SVM are, also, uniformly smaller. In regression, again, the SVM is uniformly better than the other approaches, ranging from 7.09% NRMSE to 12.12%, and it also shows uniformly smaller standard deviations. In both problems, however, it must be remarked that the error bars largely overlap, at least for $d > 0.4$ for regression. This enables us to conclude that

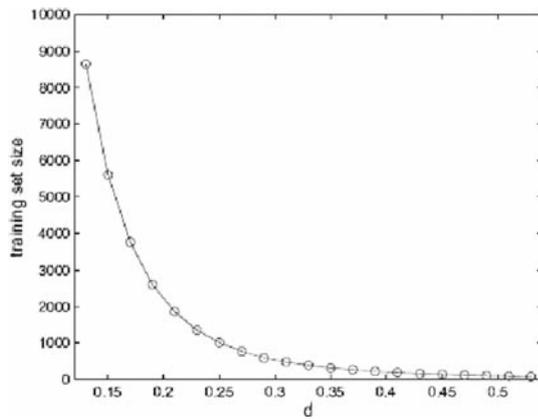


Fig. 10 Online uniformised training set size as d changes

the SVM is the winning approach overall, but that there are cases in which another approach can be better.

One last consideration: as d is increased, the size of the training sets decreases like d^{-10} , since we are building a finite partitioning of a subset of \mathbb{R}^{10} (see also Fig. 10, in which the training set size is plotted as a function of d); whereas, it seems that the accuracy of all approaches, and in both problems, only decreases linearly. This is a remarkable feature of the OU procedure, since it will always be possible to choose a 10-degree polynomially smaller training set and obtain a machine which is only linearly worse.

As a matter of fact, consider once again Fig. 9, pane (b): at the far right end we have a SVM which has a still acceptable error of $12.12 \pm 0.64\%$, but whose training set, averaged over the 5 splits, consists of 77.4 samples out of the original 153, 000!

Last, we compared the results obtained by models trained on uniform training sets with a simple random selection strategy. In this case, rather than employing the OU procedure to reduce the training sets, for each value of d we chose at random the same number of samples obtained by the OU procedure, and then trained on the models so obtained. It was expected that, on full training sets, the random strategy

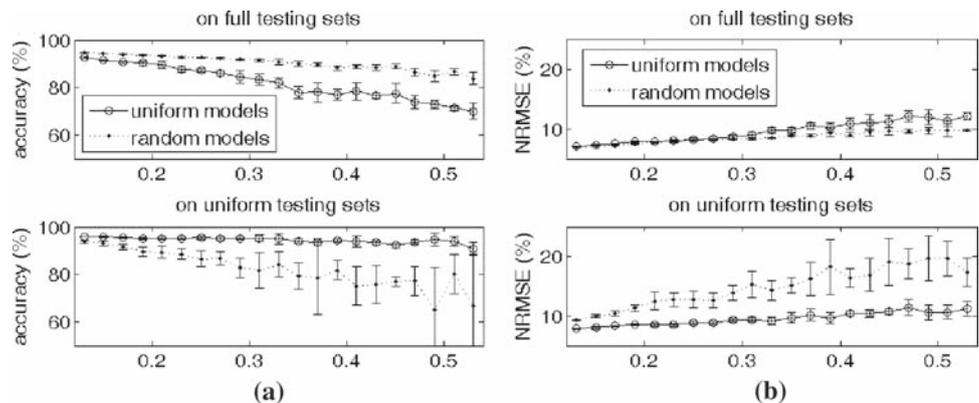
would outperform OU; this is due to the fact that a random strategy will result in training sets which have the same probability distribution as the full testing sets; whereas, the OU procedure produces training sets with a uniform probability distribution. The OU procedure, on the other hand, is expected to perform better on *uniform testing sets*, for the same reason. Figure 11 shows the comparative results for SVM classification and regression, confirming our expectations.

As one can see, both in classification and in regression, an SVM trained on uniform training sets will perform uniformly better than one trained on random training sets, when tested on uniform testing sets; and uniformly worse when tested on the full training sets. But the gap is larger in favour of the uniform training sets. This lets us conclude that uniform training sets will produce models able to perform *uniformly well* in all situations the patient should move. This is very useful in a practical setting: for example, picture a seldom performed movement, such as turning a door handle; if we were employing random training sets, we would have a much worse performance on such a movement. Uniform training sets would make the patient’s life easier.

3.3 Application to a robotic four-finger hand

The method described earlier has been used for the control of the DLR four-finger hand II Butterfass et al. (2004). This is a four-finger hand with 13 active degrees of freedom: 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

Fig. 11 Classification and regression results, comparing the OU procedure with a random sample selection strategy. **a** Classification, **b** regression



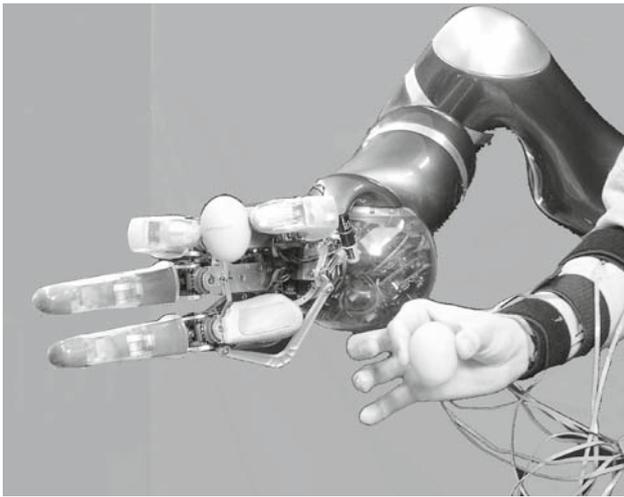


Fig. 12 The DLR four-finger hand II controlled with EMG interface, exerting the right force to hold an egg

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

In the EMG experiment, we used impedance control to exert a force as generated by the EMG system to hold and grasp object. Example use of the interface is demonstrated in Fig. 12. During the experiment, we verified that the correlation coefficient between the force applied by the subject and the robot was never less than 80% over a time window of 10s, which let us say that the force applied by the robot was almost proportional to that applied by the human subject.

4 Discussion and conclusions

Our experimental results, performed on a large data set of about 153,000 samples, clearly show that OU can be used to obtain dramatically smaller training sets with no *qualitative* loss of information; in other words, as more of the input space is sampled, OU keeps the training set up-to-date and small. OU sets will result in small (and therefore, fast) and accurate models of the sought-for EMG-to-hand map. Remarkably, OU works fine for both grasp classification and force regression, and with all three different machine learning approaches tested. Moreover, it is extremely simple, being nothing more than an online check of Euclidean distance in the input space. This check is done so far by considering the new sample's distance from all samples in the current training set, and therefore could become unfeasibly heavy as the set grows; but the same check can be clearly done in constant time using an algorithmic optimisation, such as a hash table.

Moreover, OU will produce models which perform *uniformly* well, enabling a patient to drive the prosthesis with a good accuracy in *all* situations that might arise, no matter how frequently they appear during the training phase.

The choice of the minimum inter-sample distance d is obviously crucial and depends on the required accuracy in classification and/or regression; but as we have seen, as d is increased, the machine's performance degrades only linearly, whereas the training sets become polynomially smaller. Therefore, at the price of having a slightly worse performance, dramatically smaller training sets can be used.

To sum up, in this paper we have presented a machine learning approach to joint classification of grasping and regression on the applied force, using forearm surface electromyography. The approach is totally non-invasive, easy to set up and use and it can be applied from scratch with no previous knowledge of the problem. The OU procedure can be used to incrementally build a training set which will result in small and accurate models of the problem.

Our experiments, carried out using a Support Vector Machine with Gaussian kernel, a Neural Network with sigmoidal activation function and Locally Weighted Projection Regression, indicate that the approach achieves, using a training set of about 1,800 samples on a total of 153,000 (for $d = 0.21$), an average accuracy of around 90% in classification of grasp types and a NRMSE of 7.89% in prediction of the force applied during the grasp. Of the tested approaches, SVM is marginally better than the others, especially when larger training sets are used. The OU procedure is able to find as small a training set as 77.4 samples on average (out of 153,000), which will still result in a SVM having a remarkable NRMSE of 12.12%.

Future work

We believe this is the first step towards the real application of machine learning to an EMG-driven adaptive, dexterous AHP. Let us consider the problems outlined in Sect. 2.1.2: in this paper we have solved problems 3 and 4. Now, since OU lets us obtain good accuracy with extremely small training sets, it is not too far-fetched to say that the solution of problem 2 is at hand—in principle, the changing arm posture can be taken into account simply by sampling more of the input space. As far as problem 1 is concerned, inter-subject usability is really of lesser interest, since one patient only is supposed to ever wear a prosthesis; on the other hand, multi-subject analysis has to be carried out eventually, since it must be possible to obtain good results on *any* subject the method is applied to. We see no reason, however, why this should not be the case, at least as far as able-bodied subjects are concerned.

The ultimate problem is of course that of training the system upon amputees. First of all, the patient must still have a good deal of muscular and nervous plasticity in her arm stump; then, a smart way of collecting training data must be devised—an amputee is obviously not expected to train the machine with a hand. A simple idea is that of gathering EMG

data from the patient's stump and grasping/force data from her healthy hand, while instructing her to imagine doing the same actions with both hands. Second, the issue of (sensorial) feedback will have to be addressed, in order to provide the patient with reliable information about the force actually involved in the grasp. This is likely to be crucial in order to realise a tighter loop between the patient and the prosthesis.

As far as force regression is concerned, the results presented above are, to the best of our knowledge, totally novel. Surprisingly, regression from the forearm surface EMG signal to the force applied by the hand had never been attempted before. Given the good performance obtained by our models, we claim that the relationship between the EMG signal and the force has been captured by the models, under variable conditions of muscle fatigue (within one session) and electrode displacement (within sessions belonging to different groups). In a certain sense, this work is an evolution of the so-called *proportional* feed-forward control of myoelectric prostheses already available on the market; in proportional control, the "claw" of the prosthesis is actuated with a force which is proportional to the amplitude of the EMG signal. In this case, the applied force is proportional as well, but the control is realised in a totally natural way, and is *adaptive* to the user, what has never been done before.

Acknowledgments We thank Giorgio Metta and Giulio Sandini of the Italian Institute of Technology for their support.

References

- Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, New York
- Bitzer S, van der Smagt P (2006) Learning EMG control of a robotic hand: towards active prostheses. In: Proceedings of ICRA, international conference on robotics and automation, Orlando, pp 2819–2823
- Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proceedings of the 5th annual ACM workshop on computational learning theory, ACM Press, New York, pp 144–152
- Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 2(2)
- Butterfass J, Fischer M, Grebenstein M (2004) Design and experiences with DLR hand II. In: Proceedings of the world automation congress
- Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines (and other kernel-based learning methods) CUP
- CyberHand (2007) The CyberHand project. <http://www.cyberhand.org>
- De Luca CJ (1997) The use of surface electromyography in biomechanics. <http://www.health.uottawa.ca/biomech/courses/apa4311/biomec~1.htm>
- Ekvall S, Kragić D (2005) Grasp recognition for programming by demonstration. In: Proceedings of ICRA, international conference on robotics and automation, Barcelona
- Ferguson S, Dunlop GR (2002) Grasp recognition from myoelectric signals. In: Proceedings of the Australasian conference on robotics and automation, Auckland
- Figueiredo M (2003) Adaptive sparseness for supervised learning. IEEE Trans Pattern Anal Mach Intell 25(9):1150–1159
- Huang H, Jiang L, Zhao D, Zhao J, Cai H, Liu H, Meusel P, Willberg B, Hirzinger G (2006) The development on a new biomechatronic prosthetic hand based on under-actuated mechanism. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems, pp 3791–3796
- i-Limb (2007) The i-Limb system. <http://www.touchbionics.com>
- Keerthi SS, Chapelle O, DeCoste D (2006) Building support vector machines with reduced classifier complexity. J Mach Learn Res 8:1–22
- Lee YJ, Mangasarian OL (2001) RSVM: reduced support vector machines. In: Proceedings of the SIAM international conference on data mining
- NIDAQ (2007) National instruments PCI-6023E data sheet. http://www.ni.com/pdf/products/us/4daqsc202-204_ETC_212-213.pdf
- OFTS (2007) SpaceControl GmbH optical force-torque sensor leaflet. <http://spacecontrol.de/>
- Orabona F, Castellini C, Caputo B, Luo J, Sandini G (2007) Indoor place recognition using online independent support vector machines. In: Proceedings of the British machine vision conference (BMVC) (to appear)
- OttoBock (2008a) Otto Bock MYOBOCK 13E200=50 electrodes. http://www.ottobockus.com/products/upper_limb_prosthetics/myoelectric_hands_myobockr.asp
- OttoBock (2008b) Otto Bock SensorHand hand prosthesis. http://www.ottobockus.com/products/upper_limb_prosthetics/myoelectric_hands_sensorhand.asp
- Tipping M (2000) The relevance vector machine. In: Advances in neural information processing systems, San Mateo
- Tsuji OFT, Kaneko M, Otsuka A (2003) A human-assisting manipulator teleoperated by EMG signals and arm motions. IEEE Trans Rob Autom 19(2):210–222
- Vijayakumar S, D'Souza A, Schaal S (2005) Incremental online learning in high dimensions. Neural Comput 17:2602–2634