Surface EMG for Force Control of Mechanical Hands

Claudio Castellini, Patrick van der Smagt, Giulio Sandini, Gerd Hirzinger

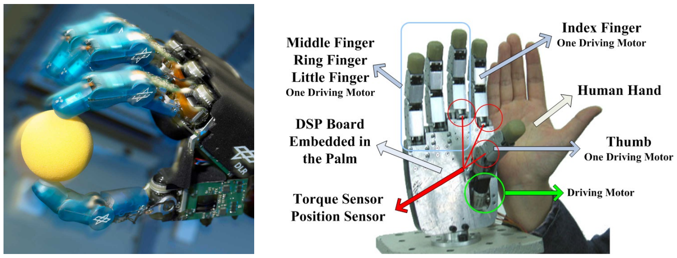


Fig. 1. (left) The DLR Hand II. (right) The DLR prosthetic hand.

Abstract—**The dexterity of active hand prosthetics is limited not only due to the limited availability of dexterous prosthetic hands, but mainly due to limitations in interfaces. How is an amputee supposed to command the prosthesis what to do (i.e., how to grasp an object) and with what force (i.e., holding a hammer or grasping an egg)? So far, in literature, the most interesting results have been achieved by applying machine learning to forearm surface electromyography (EMG) to *classify* finger movements; but this approach lacks, in general, the possibility of quantitatively determining the force applied during the grasping act.**

In this paper we address the issue by applying machine learning to the problem of *regression* from the EMG signal to the force a human subject is applying to a force sensor. A detailed comparative analysis among three different machine learning approaches (Neural Networks, Support Vector Machines and Locally Weighted Projection Regression) reveals that the type of grasp can be reconstructed with an average accuracy of 90%, and the applied force can be predicted with an average error of 10%, corresponding to about 5N over a range of 50N. None of the tested approaches clearly outperforms the others, which seems to indicate that machine learning as a whole is a viable approach.

Index Terms—**Learning and Adaptive Systems; Rehabilitation Robotics; Physical Human-Robot Interaction**

I. Introduction

In the framework of robotics for prosthetics, it is nowadays possible to build mechanically advanced prostheses, such as mechanical hands able to replicate a fair amount of the movements required by the disabled to carry on living in a decent way. Attempts in this sense include, e.g., the DLR prosthetic hand ([1] — see Figure 1), the CyberHand project [2], and the i-LIMB hand by Touch Bionics [3]. Still, a general sense of frustration impends, as far as control is concerned. How is an amputee supposed to command the prosthesis what to do (i.e., how to grasp an object) and with what force?

To this end, two types of interfaces between the patient and the prosthesis have been developed or are being studied: *invasive* and *non-invasive*. In the context of non-invasive interfaces for controlling mechanical hands, a concrete possibility arises from *forearm surface electromyography (EMG)* [4], 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 what muscles the patient is activating, and with what force. Still, the EMG signal suffers from a number of problems, among which the placement of the electrodes, signal drifting and change due to sweat formation and muscular fatigue and cross-talking among deep and superficial muscles. See [5] for an overview of the problems posed by this signal. In this paper, we pursue the approach based upon *machine learning* techniques already attempted in, e.g., [6], [7], [8].

So far, in literature, machine learning applied to surface EMG has been able to *classify* finger movement only. However, control of the exerted force is crucial, for instance for holding a hammer or grasping an egg. We show a detailed comparative analysis of what machine learning can do when applied to such a problem. Over two days, we have gathered forearm surface EMG data while holding a force sensor with different finger combinations. We have then trained three different machine learning systems to compute, from the EMG signal, the finger movements and the finger force. The three approaches we have experimented with are: (a) a simple feed-forward neural network with one hidden layer, (b) a Support Vector Machine with radial basis function kernel [9], (c) Locally Weighted Projection Regression [10]. Our numerical results indicate that, in such a scenario, the type of grasp can be reconstructed with an average accuracy of 90%, and the applied force can be predicted with an average percentage error of 10%, meaning about 5N over a range 50N.

II. Materials and Methods

A. Experimental Setup and Design

The aim of the experiment was to gather forearm surface EMG data in real-time, and to relate it to (a) the type of grasp applied by the subject, (b) the force applied by the subject while grasping.

1) General setup description: The experiment consisted of freely, repeatedly grasping a SpaceControl OFTS force/torque sensor [11] orthogonally to its large face. 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, at different speeds and with varying force. Four force sensing resistors (FSRs) were applied on the subject's hand fingertips (thumb, index, middle and ring),

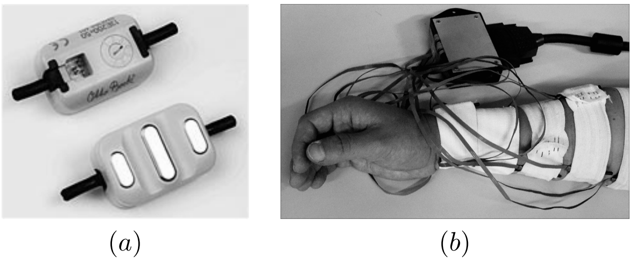


Fig. 2. The experimental setup. (a) An Otto Bock 13E200=50 surface EMG electrode. (b) The arm of the subject with the EMG electrodes fitted and held in place by elastic bands.

in order to be able to detect which grasp type was used at each instant of time. At the same time, 10 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. Figure 2 shows some parts of the setup.

Numerical data from the EMG electrodes and FSRs were gathered at a sampling rate of 256Hz using a National Instruments DAQ PCI-6023E analogic/digital conversion card [12], mounted on a fast PC equipped with Windows XP. Data coming from the OFTS sensors was gathered via the serial port at about 80KHz. The data were subsequently synchronised.

2) EMG signal and electrode placement: The 10 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 [13], used at an amplification factor of about $14,000$. 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. Placement was done following the description in [8], 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) the subject, (2) the arm posture, (3) electrode placement, and (4) muscle fatigue. As far as the first problem is concerned, since in a real setting one person only is expected to train and wear the prosthesis, we have not investigated multi-subject feasibility of the approach, concentrating on one subject only, male, aged 35 and fully able-bodied. In subsequent work, independent multi-subject analysis will have to be carried on.

In order to overcome the second problem, we instructed the subject to keep the arm still and relaxed on a table in a comfortable position, with the palm orthogonal to the plane of the table. Again, a deeper investigation is required in a real setting, since the prosthesis is supposed to be worn all day long in all possible tasks of everyday life.

We then organised the experiment as follows: the subject was instructed to continually grasp the sensor over a period of time of 3 to 4 minutes; then he was allowed to rest for about two minutes. 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 one 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 procedure was entirely 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).

3) Force applied during the grasp: The output of the linear OFTS force/torque sensor lies between 0N and 100N, with a resolution of 0.02N.

4) Type of grasp: The voltage values output by the 4 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 lastly 4, grasp by opposing the thumb and all other fingers.

B. Data Analysis

Let $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$ denote the *training set*, that is, the set of samples gathered for training a machine learning system. 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, $\mathbf{x}_i \in \mathbb{R}^{10}$. In the case of classification, each category representing a grasping type would be represented as an integer value, that is, $y_i \in \{0, 1, \ldots, 4\}$. In the case of regression, the force value would be directly encoded as a real number, that is, $y_i \in \mathbb{R}$. Before any analysis, all samples were normalised, as is customary, by subtracting the mean values and dividing by the standard deviation, for each input space dimension. No filtering whatsoever was applied to the input signals, in order to have a more realistic, delay-free result.[1]

1) Neural Networks: For our experiments we used a feed-forward neural network with 10 inputs, 10 sigmoidal hidden units, and 6 outputs, 5 representing the output class and one the force. Training was done using the Levenberg-Marquardt algorithm [14].

2) Support Vector Machines: Support Vector Machines (see, e.g., [9], [15], [16]) are a machine learning method able to perform both classification and regression. SVMs do not suffer from the problem of local minima, as is the case of NNs, but their training time can be prohibitive on large data-sets. Therefore we have used a *uniformisation* strategy on the training sets: removing those samples which are too

[1] a few experiments have been run with filtered input, revealing non-conclusive results; in a few more we applied a low-pass filter on the output, which improved the results a little.

close to each other, according to the Mahalanobis 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 distance in which each summand is weighted inversely with respect to the variance of the samples along that dimension of the input space: it is therefore a measure of distance independent of the variance of the single electrodes. We adopted an approximated method which was able to remove most, but not all, samples too close to any other sample. After a few initial experiments we set the threshold distance at 1. We also decided to employ the Gaussian kernel. All our experiments with SVMs were then performed on uniformised training sets, using 5-fold cross-validation and grid search to find the optimal values of the standard Gaussian kernel hyperparameters, C and σ. On the other hand, notice that no *testing* set was uniformised, since it would probably be unfeasible to apply the same procedure in an on-line setting. Notice, further, that applying uniformisation resulted in training sets which were considerably smaller than the original ones, up to about 100 times smaller. We employed a well-known freely available SVM package, *libsvm* v2.83 [17], in the Matlab wrapped flavour.

3) Locally Weighted Projection Regression: Locally Weighted Projection Regression (LWPR [10]) 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). Therefore we used it for regression only, and trained it with the uniformisation procedure.

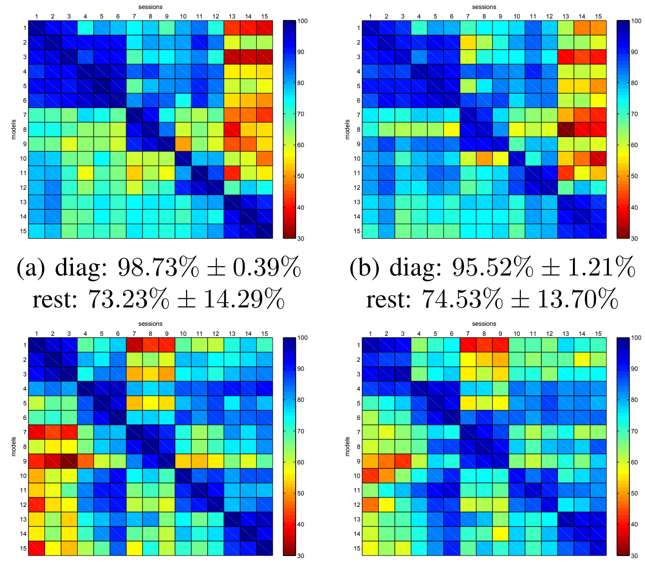
We used the latest stable C version of LWPR, kindly made available by Stefan Klanke, wrapped in a Matlab command interface. We chose to use the Radial Basis Function kernel and meta-learning, and then performed 5-fold cross-validation and found the initial values of the distance metric for receptive fields by grid search.

III. Experimental Evaluation

In this Section we describe the results obtained from the experimental evaluation of our data. In particular, we first describe the general strategy devised to train our machines; we then show a detailed comparison of the three approaches selected, first on the classification problem, and then on the regression problem.

A. Training strategy

1) Uniformisation: First of all we chose one of the selected approaches (NNs, SVMs or LWPR) and one problem (classification or regression) for initial experimentation. We decided to use SVMs for classification. We then decided to test whether data obtained during one session could be used to build a model able to generalise over other sessions during the same day. At the same time, we wanted to check whether the uniformisation procedure (see Section II-B) would be effective in reducing the training set, at the same time without losing relevant information.



(a) diag: 98.73% ± 0.39%
rest: 73.23% ± 14.29%

(b) diag: 95.52% ± 1.21%
rest: 74.53% ± 13.70%

(c) diag: 99.05% ± 0.37%
rest: 73.23% ± 14.47%

(d) diag: 95.37% ± 2.85%
rest: 74.38% ± 12.27%

Fig. 3. Cross-session analysis and evaluation of the uniformisation procedure. (a) and (b), accuracy matrices for day 1: (a) full models, (b) uniform models. (c) and (d), same for day 2.

Therefore, we trained a SVM over each single session, both for the first and second day, and both with full and uniformised training sets. The session were 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. Moreover, in the remainder of this Subsection, if the training set of a model was uniformised, we will call the model *uniform*.

We then tested each model on all sessions of the same day, obtaining an *accuracy matrix* A in which A_{ij} would be a percentage denoting the correctly guessed labels when testing model i (standard or uniform) on session j. Notice once again that, in the case of uniform models, the test is carried out on the *full* session. This is what we would call a *cross-session analysis*. The result is visible in Figure 3.

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. It is apparent from the shades that the full models attain a better accuracy when tested on their own training data, that is, on the diagonal of the matrix, than the uniform models. This is intuitively sensible since, in the case of full models, we are testing on the same data used for training, whereas in the case of uniform models, the training data are a strict subset of the testing data, and a quite smaller subset indeed. But as well, if we consider the remaining elements of the matrices, it is clear that the uniform models attain a better accuracy *overall*, if compared to the full models. The same analysis for day 2 (same Figure, (c) and (d)) yields analogous results.

From this we conclude that the uniformisation procedure is effective in reducing the training set size, without actually degrading the performance. This is apparent from the fact that uniform models are more accurate on testing sets which are disjoint from the training sets. In fact, one should always ensure that this is the case, aiming for a better generalisation error. We can say that uniform models generalise better, at least in this case.

Therefore, from now on, all models we will be using, in the case of SVMs and LWPR, will be the uniform ones.

2) Classification accuracy: Consider Figure 3 again, panes (b) and (d). The accuracy attained on non-diagonal elements is about 74%, which is rather bad. One cannot expect to correctly drive a prosthesis if one sample in four is misclassified. At the same time, however, a strong "good group accuracy" is obviously present: in each matrix, good accuracy values are obtained on 3×3 submatrices located on the diagonals, corresponding to cross-session accuracy for sessions *belonging to the same group*, that is, where the elastic bands were not removed and no electrode displacement was present.

More in detail, as far as the first day is concerned (same Figure, 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 motivated by the opposite reason: during those sessions, the subject must have explored more relevant 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 we can confirm 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 its effetcs are visible 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.

3) Best models: Lastly, we have considered how to collect *only* relevant information from the gathered data, that is, how to sensibly reduce the training set. To do this, we have collected the two best models for each day and joined them together. For instance, consider Figure 3 again, pane (b). It is apparent that model 4 performs well on sessions 1 to 12, whereas model 13 does well on sessions 13 to 15. We then decided to use these two models 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.

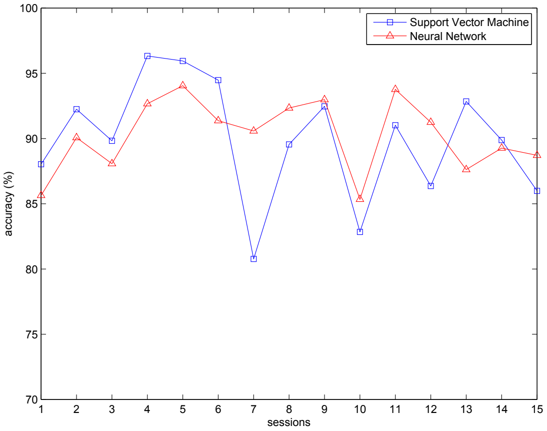


Fig. 4. Classification accuracy of best models, day 1. SVM: $89.90\% \pm 4.51\%$; NN: $90.25\% \pm 2.77\%$.

This procedure was repeated for each problem tackled (classification, regression) and approach tested (NN, SVM, and LWPR). The results are presented below.

B. Grasp Classification

Figure 4 shows the classification accuracy of the best models for classification on all 15 sessions of day 1, for SVMs and NNs. The analysis detailed in the previous Subsection has been repeated for the Neural Network. In that case, models 8 and 15 of day 1 have been used to build the best model. Result for day 2 are similar.

As one can see, there is no clear winner between SVMs and NNs. NNs perform slightly better on day 1 (lower 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 NNs), which is about 12-15 minutes of user activity. But notice, that samples gathered during both days were necessary to have an idea of which sessions to use.

C. EMG to Force Regression

The most interesting part of our research 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 Subsection III-A 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 NNs and $6, 13, 3, 4$ for LWPR.

For regression, we have considered three quite standard indicators of performance:

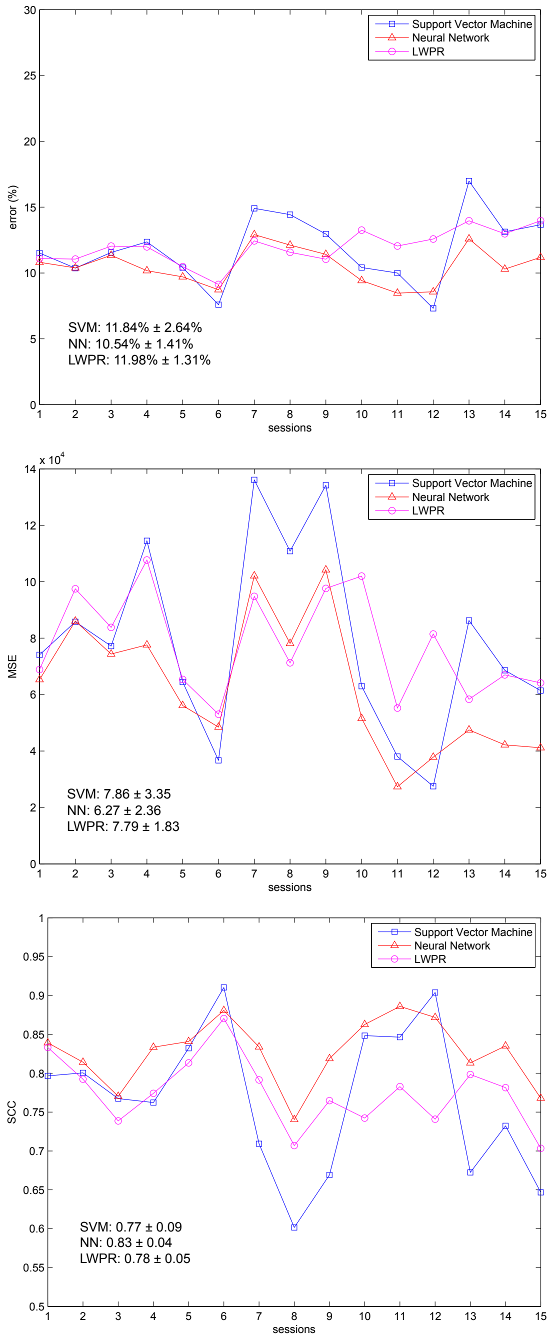


Fig. 5. Regression accuracy of best models, day 1. Top to bottom: MSE, Normalised Root MSE, Squared Correlation Coefficient.

1) the Mean Squared Error (MSE) in its standard definition;
2) 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
3) the Squared Correlation Coefficient (SCC) between the predicted target and the real target.

Figure 5 shows the results for day 1. Consider the first pane, 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. NNs 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%.

Consider now the second and third panes of the Figure. First of all there is a clear inverse correlation between the MSE and the SCC, as it is expected: for both days and for all approaches, a larger MSE corresponds to a lower SCC. Secondly, 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 (Figure 5, second row): 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.

Lastly, in practical terms: the best average MSE obtained by NNs ($6.27 \cdot 10^4 \pm 2.36 \cdot 10^4$ and $4.76 \cdot 10^4 \pm 1.52 \cdot 10^4$) corresponds to, in turn, an average error of 5N and 4.36N. Figure 6 shows some samples of the force values obtained from the OFTS, along with the corresponding values predicted by the best approach, that is, NNs. As one can see, despite the non perfect correspondence of the two curves, the NN definitely follows the real target to a remarkable degree of accuracy, for a wide range of frequencies of the pressing/releasing action.

IV. Discussion

As far as regression is concerned, the results presented above are, to the best of our knowledge, totally novel. 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).

One remarkable point, to be further investigated, is the *correct* sampling of the EMG signal. The signal presents, as already said, a wide variance due to a number of factors. Our work shows clearly that, if the relevant data only is gathered, a remarkable accuracy, both in classification and regression, can be achieved. So far, this has been done using (a) the uniformisation procedure to *reduce* the training sets, that is, to eliminate form the training sets irrelevant data; and (b) a careful a-posteriori selection of the models which had performed best on a thorough cross-sessions analysis.

Both procedures, actually, can be seen as the need to sample the relevant portion of the input space in a uniform way. One can think of an online version of the uniformisation procedure, in which new samples deemed irrelevant (according to a suitable inter-sample distance measure, such as the Mahalanobis distance) are rejected and never used for training. The same procedure would, when presented with samples sufficiently distant from the current training set, decide to actually use them. This could possibly lead to an incrementally growing training set, which would eventually reach a plateau, when all relevant portions of the input space have been explored by the user.

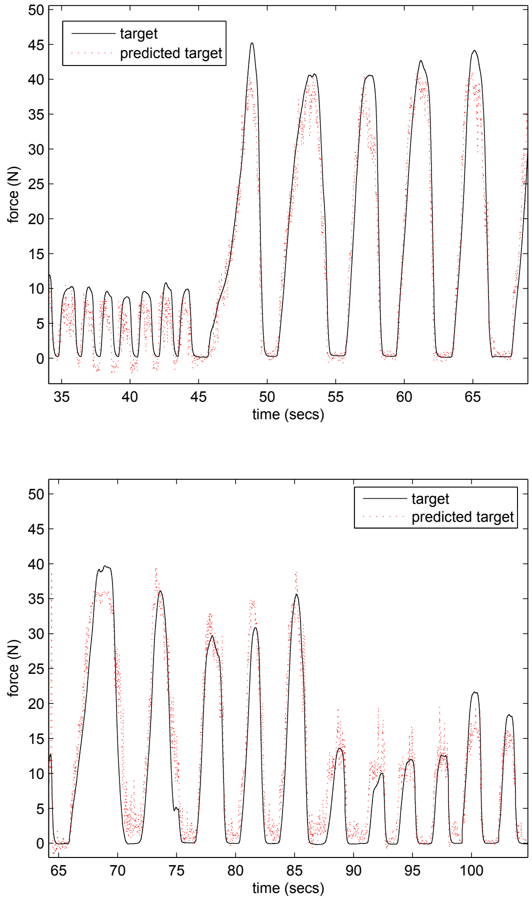


Fig. 6. Examples of the force target value, as guessed by our Neural Network. (upper pane) session 6, day 1, MSE: $4.85 \cdot 10^4$; (lower pane) session 10, day 2, MSE: $5.48 \cdot 10^4$

The current setup and analysis can be improved in a number of ways, inclding a better way of gathering the force at the finger joints, subsampling, pre- and post-filtering, and sampling in different positions of the arm (not only the forearm), and/or while the arm is moving.

V. Conclusions

In this paper we have presented a machine learning approach to joint classification of grasping and regression on measured human grasp force, using forearm surface electromyography. The approach is totally non-invasive and easy to set up and use, and it can require as little as about 15 minutes of training to achieve good results. (This does not include the time required for uniformisation, which was not optimised at this stage.)

Our experiments, carried out using a Support Vector Machine with Gaussian kernels, a Neural Network with sigmoidal activation function and Locally Weighted Projection Regression, indicate that the approach achieves an average accuracy of around 90% in classification of grasp types and a normalised root MSE of 10% in prediciton of the force applied during the grasp. This makes it suitable for driving a force-controlled robotic hand, and opens a new field of applications in prosthetic hands.

The approach has recently been applied to the DLR-II Hand (see Figure 1 and [18]) with good results, which will be shortly made available to the scientific community. Future work is aimed at testing the real usability of the proposed approach by applying it to a dexterous prosthetic hand and experimenting on disabled patients.

Acknowledgments

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

References

[1] H. Huang, L. Jiang, D. Zhao, J. Zhao, H. Cai, H. Liu, P. Meusel, B. Willberg, and G. Hirzinger, “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*, 2006, pp. 3791–3796.

[2] “The CyberHand project.” [Online]. Available: http://www.cyberhand.org

[3] “The i-Limb system.” [Online]. Available: www.touchbionics.com

[4] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, “Control of multifunctional prosthetic hands by processing the electromyographic signal,” *Critical Reviews in Biomedical Engineering*, vol. 30, no. 4—6, pp. 459—485, 2002.

[5] C. J. D. Luca, “The use of surface electromyography in biomechanics,” 1997. [Online]. Available: www.health.uottawa.ca/biomech/courses/apa4311/biomec∼1.htm

[6] S. Ferguson and G. R. Dunlop, “Grasp recognition from myoelectric signals,” in *Proceedings of the Australasian Conference on Robotics and Automation, Auckland, New Zealand*, 2002.

[7] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, “A human-assisting manipulator teleoperated by EMG signals and arm motions,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 210—222, apr 2003.

[8] S. Bitzer and P. van der Smagt, “Learning EMG control of a robotic hand: Towards active prostheses,” in *Proceedings of ICRA, International Conference on Robotics and Automation, Orlando, Florida, USA*, may 2006, pp. 2819–2823.

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. ACM press, 1992, pp. 144–152.

[10] S. Vijayakumar, A. D’Souza, and S. Schaal, “Incremental online learning in high dimensions,” *Neural Computation*, vol. 17, pp. 2602—2634, 2005.

[11] “SpaceControl GmbH Optical Force-Torque Sensor leaflet.” [Online]. Available: spacecontrol.de/fileadmin/user_upload/LEAFLET_OFTS_100.pdf

[12] “National Instruments PCI-6023E data sheet.” [Online]. Available: www.ni.com/pdf/products/us/4daqsc202-204_ETC_212-213.pdf

[13] “Otto Bock MYOBOCK 13E200=50 electrodes.” [Online]. Available: http://www.ottobockus.com/products/upper_limb_prosthetics/myoelectric_hands_myobockr.asp

[14] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *SIAM Journal on Applied Mathematics*, vol. 11, pp. 431—441, 1963.

[15] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Knowledge Discovery and Data Mining*, vol. 2, no. 2, 1998.

[16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. CUP, 2000.

[17] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[18] J. Butterfass, M. Fischer, and M. Grebenstein, “Design and experiences with DLR hand II,” in *Proceedings of the World Automation Congress*, 2004.