Analysis of Neural Network based Proportional Myoelectric Hand Prosthesis Control

Michael Wand, *Member, IEEE*, Morten B. Kristoffersen, Andreas W. Franzke, Jürgen Schmidhuber

***Abstract*— *Objective:* We show that state-of-the-art deep neural networks achieve superior results in regression-based multi-class proportional myoelectric hand prosthesis control than two common baseline approaches, and we analyze the neural network mapping to explain why this is the case. *Methods:* Feedforward neural networks and baseline systems are trained on an offline corpus of 11 able-bodied subjects and 4 prosthesis wearers, using the R^2 score as metric. Analysis is performed using diverse qualitative and quantitative approaches, followed by a rigorous evaluation. *Results:* Our best neural networks have at least three hidden layers with at least 128 neurons per layer; smaller architectures, as used by many prior studies, perform substantially worse. The key to good performance is to both optimally regress the target movement, *and* to suppress spurious movements. Due to the properties of the underlying data, this is impossible to achieve with linear methods, but can be attained with high exactness using sufficiently large neural networks. *Conclusion:* Neural networks perform significantly better than common linear approaches in the given task, in particular when sufficiently large architectures are used. This can be explained by salient properties of the underlying data, and by theoretical and experimental analysis of the neural network mapping. *Significance:* To the best of our knowledge, this work is the first one in the field which not only reports that large and deep neural networks are superior to existing architectures, but also explains this result.**

***Index Terms*— Electromyography, Machine learning, Neural Networks, Prosthesis**

I. Introduction

Recent years have seen major improvements in the development and commercialization of electric hand prostheses which are capable of performing complex movements involving a large number of degrees of freedom (DOF). A common method to obtain control signals for such prostheses is based on surface *electromyography* (EMG) [1], where small electrical currents are captured from the residual muscles in the arm stump via electrodes which are built into the prosthesis socket. Unfortunately, it is still an open question how to obtain optimal control signals for these highly dexterous prostheses. Conventional devices, which have been in widespread use for years, are based on a *direct control* scheme: A pair of electrodes is attached to two antagonist muscles, allowing straightforward proportional control of *one* degree of freedom [2]. In order to switch between functions of the prosthesis (e.g. wrist rotation vs. grips), a special signal (usually a *co-contraction* of muscles) must be performed. While this method is conceptually simple, users nonetheless describe it as "too time consuming, unreliable, non-intuitive, and mentally exhausting" [3]. Furthermore, it does not allow to control multiple DOFs simultaneously.

In the past years, more sophisticated electromyographic control schemes have been researched [5]–[8] and brought to the market (e.g. [9]). These systems are based on *pattern recognition*: The user performs a phantom movement corresponding to the desired prosthesis action, and a *machine learning* (ML) system translates the observed myoelectric signals to a motor command to be executed by the prosthesis. An easy way to achieve *simultaneous proportional* control of multiple DOF is to treat this task as a *regression* problem, i.e. as a mapping from EMG frames to real-valued vectors [10].

Over the past few decades, *deep learning* based on *neural networks* (NN) has become the method of choice for solving a large variety of complicated classification and regression problems [11], [12]. NNs have been applied to myoelectric prosthesis control [13]–[17], but usually with very small NN topologies, only very recently, larger networks [18], [19] and specialized topologies [20] have been employed. Little work has been done on systematically comparing state-of-the-art NN regressors, in particular using large neural networks with several hidden layers, with the linear baseline systems, and on explaining why the former yield superior performance compared to the latter.

The present study aims to fill this gap: We systematically optimize a state-of-the-art feedforward neural network, trained as a nonlinear regressor from EMG features to movement commands, and analyze the mapping it performs. We compare this network to two baseline systems (linear regression and LDA-based classification), showing not only that the NN has indeed superior performance, but also explaining why this is

Manuscript submitted for review on May 2, 2021. This work was supported in part from the EU H2020 research and innovation programme under grant agreement number 687795 (Acronym: INPUT).

Michael Wand is with the Swiss AI Lab IDSIA, USI & SUPSI, Lugano-Viganello, Switzerland, and with the Institute for Digital Technologies for Personalized Healthcare, SUPSI, Lugano-Viganello, Switzerland. Jürgen Schmidhuber is with the Swiss AI Lab IDSIA, USI & SUPSI, Lugano-Viganello, Switzerland, and with the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.

When this study was performed, Morten B. Kristoffersen and Andreas W. Franzke were with University of Groningen, University Medical Center Groningen, Department of Rehabilitation Medicine, Groningen, The Netherlands; Morten B. Kristoffersen is now with the Center for Bionics and Pain Research, Mlndal, Sweden, and with the Department of Orthopaedics, Institute of Clinical Sciences, Sahlgrenska Academy, University of Gothenburg, Gothenburg, Sweden.

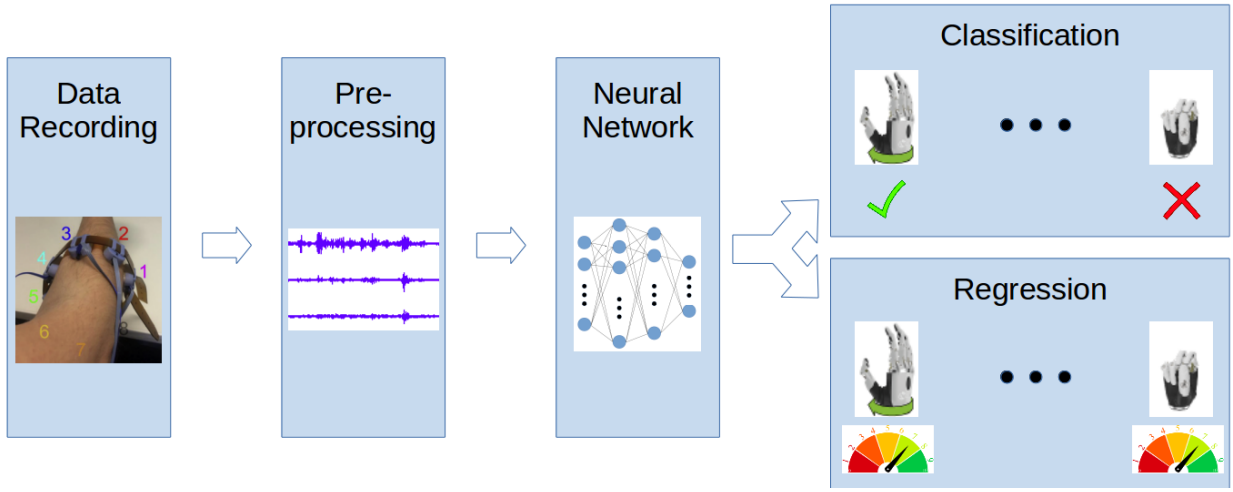


Fig. 1: Overview of the processing pipeline for EMG-based prosthesis control (image of electrode ring taken from [4] under the terms of the CC BY license).

the case. To the best of our knowledge, the latter is a genuinely new contribution in the field of NN-based myoelectric prosthesis control; we likewise believe the optimization part to be extensive and useful to researchers and practitioners alike.

We consider both single and simultaneous movements. Our evaluation metric is the R^2 score [21] on an offline corpus taken from [22] of 15 subjects, 4 of whom are transradial amputees, and 11 are able-bodied. While it is acknowledged that the offline performance of a myoelectric controller does not fully correspond to the usability of the prosthesis [23], it nonetheless is efficient for comparing a large number of different systems; for a presentation of our full system, including user studies and training, we refer to the companion paper [24].

II. Related Work

While the first myoelectric hand prosthesis has been developed as early as during the 1940's ([25], cited according to [26]), here we focus on more contemporary work, considering only techniques based on pattern recognition. Two categories of systems have been presented in literature, table I presents an overview of their key properties; figure 1 displays the processing chain graphically.

Classification-based systems translate every (preprocessed) EMG frame into one of several movement classes, e.g. wrist rotations, grips, etc. Linear Discriminant Analysis (LDA) is commonly used as classifier due to its simplicity, the possibility of real-time implementation, and its relatively good performance [22], [27]–[30]. Other linear methods have occasionally been applied, e.g. Common Spatial Patterns [31], as well as nonlinear approaches, e.g. Quadratic Discriminant Analysis [32]. *Neural networks* (NN) have likewise been used as classifiers [33]. The contraction level is estimated separately, for example directly from the EMG signal energy, or (better) by regressing on the basis of some nonlinear feature derived from the raw EMG signal, e.g. the root mean square [16], [30]. While this method provides natural and intuitive control and scales well in the number of possible movements, it still does not allow to control several DOF simultaneously, since only one movement is even recognized at a given time step. In order to overcome this limitation, somewhat complex setups for multi-way classification have to be devised [28], [34].

	Classification	Regression
Targets	Categorical	Real-valued
Simultaneous Movements	No	Yes
Strength Estimation	No∗	Yes
Suppression of Spurious Movements	Easy	Problematic
Graceful Degradation	No	Yes

∗ can be performed separately

TABLE I: Comparison of Classification and Regression setup for ML-based prosthesis control

Second, one can consider the mapping between EMG frames and control commands as a **regression** task, which means that the system outputs a vector of movement commands for each input frame. This is a natural way to allow simultaneous proportional control of multiple DOF, additionally, it has been shown to be more robust to signal nonstationarities than the classification approach [10]. Linear Regression is the most straightforward choice for this task, its performance can be substantially enhanced by using well-chosen EMG features [16], or by using kernel methods [17]. Neural networks have been applied [13], [15]–[17] using very small network topologies with only one hidden layer.

ML-based systems require annotated *training data*, i.e. training data with target movement labels, in order to bootstrap the system. In the case of able-bodied control subjects, it is easy to measure the movement which the user performs, for example by force/torque transducers [13], or even by manually or automatically annotating a visual recording of the data collection. For prosthesis wearers, this method is clearly not possible. A common idea for *unilaterally* amputated subjects consists in the subjects mirroring the intended movement with the healthy arm [15], [33]. Alternatively, one gives heuristic real-time feedback during the recording process [16], [31].

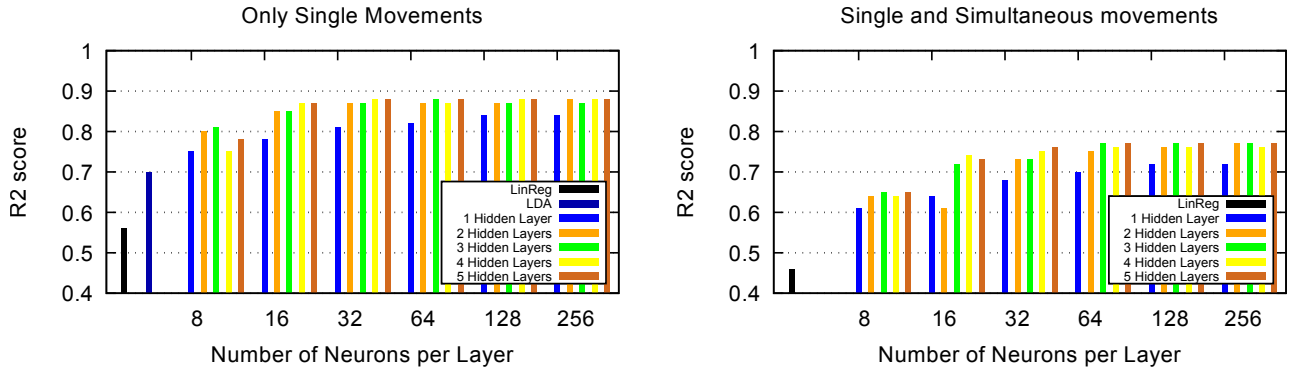


Fig. 2: Breakdown of test R^2 scores for different network topologies, standard Hudgins features, corpus with and without simultaneous movements. Averaged over development subjects.

This latter method was also used for this study, it has the advantage of also working for bilateral amputees.

Finally, when the prosthesis is re-donned, the pretrained ML system may not exactly fit the new recording conditions, due to slight position shifts, environmental conditions, etc. Several methods to efficiently *adapt* the system to the new setup have been presented, often relaxing the requirement to have exact data annotations for the adaptation data (*unsupervised* or *semi-supervised* adaptation) [32], [35]–[37].

III. Data Corpus and Preprocessing

Corpus Our experiments are based on a data corpus of 15 subjects, namely 11 able-bodied persons (subjects 1 – 11) and 4 amputees (subjects 12 – 15). This corpus was first presented in [22], where we refer the reader for further details. The recording protocol was as follows.

8 equally spaced double differential electrodes were placed around the forearm of the subject, at approximately 6.5 – 7 cm distance to the elbow. For each amputee, an individual hard shaft prosthesis socket housing the electrodes was manufactured; for the able-bodied subject, a standard commercially-available electrode ring (13E200 = 50AC Otto Bock Healthcare Products GmbH, Vienna, Austria) was used. For able bodied subjects, recordings were performed on five different days; on each day, three sessions were recorded, where the electrodes were rotated with a shift of ± 8 mm between the sessions. For amputees, 10 sessions were collected on 5 days, where the position of the electrodes varied naturally, an exact repositioning was not possible since the prosthesis socket covered the electrodes. Recordings were performed at a sampling rate of 1kHz; the total corpus length is slightly above 34 hours.

A session consists of 120 sample recordings, each of which lasts 5 seconds. For each recording, the user was prompted to perform one out of seven movements, namely *fine pinch, hand open, key grip, wrist extension, wrist flexion, wrist pronation, wrist supination*, or to keep the arm in a resting position (*no movement*). In the case of actual movements, the user was asked to perform a trapezoidal contraction trajectory, reaching a maximum contraction level of 30%, 60%, or 90% MVC; a corresponding reference line was shown to the subject along with the RMS value across all EMG channels as biofeedback. The subjects were asked to follow the trapezoidal reference lines as well as possible, so that the contractions would be consistent and repeatable.

Data Augmentation Since the original corpus does not contain EMG recordings of simultaneous movements, we simulate these as follows. Movements are exclusive within any of the three groups: a) grips (*fine pinch, key grip, hand open*), b) *wrist extension/flexion*, c) *wrist pronation/supination*. However movements from two different groups can be freely combined; we did not consider combinations of three movements. Thus, for example *wrist pronation* combined with a *key grip* would be a valid combination. For each possible combination of two movements, we randomly chose two corresponding sample movements from the corpus and summed the raw EMG signals for each channel, thus simulating a combined movement. Note that the two movements can have different maximum contraction levels. This data augmentation approximately doubled the corpus size, i.e. around half the samples in the final corpus are the original single movements, the other half are simultaneous movements. It has been demonstrated experimentally that EMG signals behave approximatively linearly when overlaying independent contractions [38].

Features To compute features from single or simultaneous movements, each EMG channel is windowed into overlapping windows of 128 ms with a step size of 50 ms; in a real-time application this allows to transmit control commands at 20 Hz with less than 200 ms delay. We experimented with a set of time-domain features following Hudgins [39] (mean absolute value (MAV), zero crossing rate (ZC), waveform length (WL), slope sign change (SSC)) and Hahne [16] (variance (VAR), log-variance (LV), RMS). In order to keep the complexity of our experiments under control, we refrained from using more complex features, like autoregressive or wavelet coefficients [13]. Since Hahne's features, which are related to the original Hudgins features by relatively simple transformations, did not improve any of our systems (see section V), we finally chose the set of four Hudgins features as our *standard* feature set; thus a standard feature vector has $4 \times 8 = 32$ components. Features of each recording session are z-normalized separately.

Data Splits Our systems are always subject-dependent.

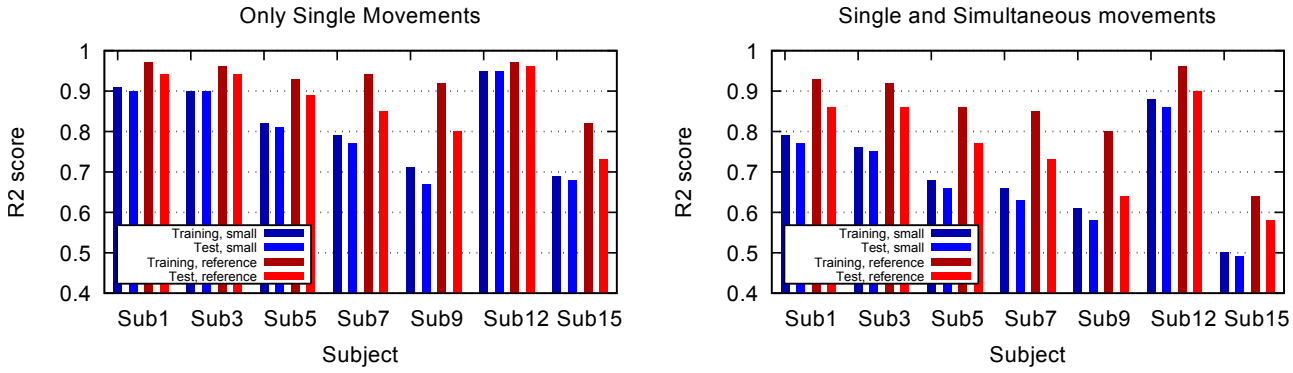


Fig. 3: Breakdown of R^2 scores on training and test subsets for different subjects and two network topologies (small: one hidden layer with 32 neurons, reference: three hidden layers with 128 neurons), standard Hudgins features, corpus with and without simultaneous movements.

From each recording session of a subject, 25% of the samples are chosen for testing, while the remaining samples are used for training the neural networks. The choice is performed in a stratified manner, so that all single and all simultaneous movements appear in both the training and the test subsets. Data from all sessions of a subject is combined in both training and test. Furthermore, data of 5 able-bodied subjects (#1,3,5,7,9) and two prosthesis subjects (#12 and #15[1]) are used for developing the system and tuning parameters, data of the remaining subjects were held back for final evaluation.

IV. Machine Learning based Control System

The ML control system consists of three components [24]: (1) feature extraction, as described above, (2) computation of movement commands from EMG features, (3) post-processing of movement commands to improve usability and suppress errors, applied only in online settings.

We use a feed-forward NN as a regressor to compute hand movement commands from EMG features. The NN consists of a series of hidden layers, each followed by a ReLU nonlinearity, and a final linear layer with seven target neurons for regression to the seven possible movements.

Based on a series of preliminary experiments, we limit our investigation to networks where all hidden layers have the same number of neurons, thus we did not investigate "bottleneck" or "funnel" style networks. We experimented with using dropout in various configurations to improve the NN performance on the test data, but did not observe any improvement; see section V for a further consideration. We likewise did not obtain improved results by training our neural networks on raw EMG data, using convolutional architectures as in [40] (for speech recognition). Finally, we excluded *recurrent* neural networks (RNNs) from our experiments since they did not yield any practical benefit, neither in the offline evaluation nor in user studies; however they required substantially more computations in training and application, which may be detrimental if the system is deployed on an embedded microcontroller. A further disadvantage of RNNs could be their very capability of learning entire *sequences* of movements: During system application, the RNN might be biased towards movement sequences which occurred during training.

The NN is trained with the ADAM optimizer [41] with standard parameters. The training criterion is the Mean Squared Error (MSE) between the regression output and the target. The system evaluation is based on the R^2 score, which derives from the MSE; thus it makes sense to use the MSE as training target. For this study, the algorithm was implemented in PyTorch [42].

When the system is applied in real-time ("online") mode, the regression output is processed as follows: First, when conflicting movement commands (like wrist flexion and extension) occur, the command with higher amplitude is kept. Second, a set of postprocessors can be applied; in our experiments, the most useful ones were thresholding of low-amplitude movements and a low-pass (momentum) filter. For the offline evaluation in this paper, no postprocessing was performed.

We compare the NN regressor with two baseline systems. The first one applies linear regression from the EMG features to the movement commands. The second one applies LDA as a movement classifier, and linear regression to estimate the strength of the movement [30], note that this method does not allow to perform simultaneous movements. The baseline systems were implemented in Scikit-Learn [43].

V. System Optimization

Figure 2 displays a breakdown of R^2 scores for different network topologies, with and without augmenting the data corpus with simultaneous movements (always for both training and test data, see section III). All results are averaged over the test sets of the seven development subjects, unless indicated otherwise. Note that the LDA cannot be applied for simultaneous movements.

A. Topology Optimization

As a first observation, we note that the R^2 score is substantially higher in the (simpler) case where only single movements need to be recognized. Apart from this, the behavior is

[1]Subject 15 was chosen to be in the development set because preliminary evaluations had shown that this subject is a negative outlier regarding data quality, thus making it interesting for our experiments.

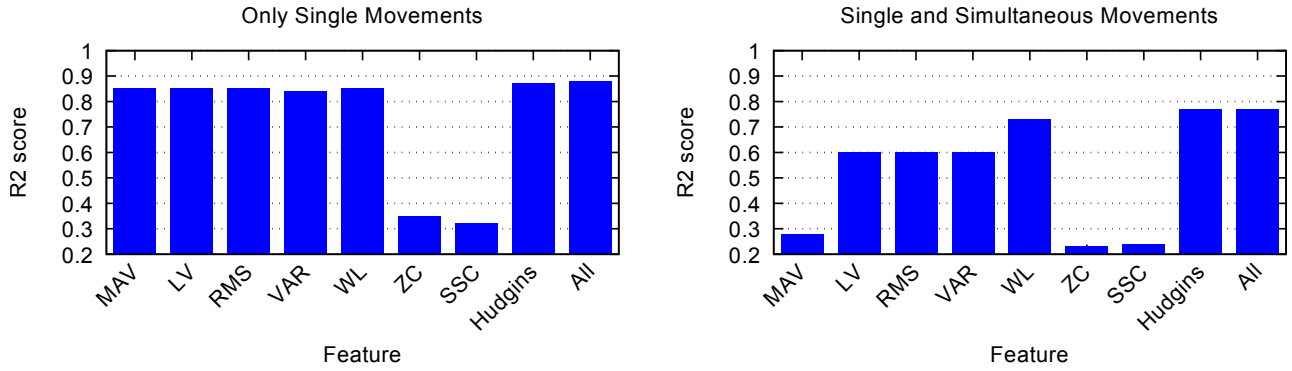


Fig. 4: R^2 scores on different single features, as well as on the standard Hudgins feature set and an extended feature set (see section III for details), using the reference NN architecture. Averaged over development subjects.

consistent across both these data setups: Both linear methods, i.e. linear regression and LDA, perform substantially worse than the neural networks. The NNs improve with increasing size and depth, up to approximately three layers with 128 neurons each; yet even a single hidden layer with only 8 neurons offers substantial benefit. In the remainder of this paper, the NN with three 128-neuron layers will be our *reference* NN; its average R^2 score over the development subjects is 0.87 for single movements, and 0.77 for single and simultaneous movements. For future reference, we also define a *small* architecture, namely an NN with one hidden layer with 32 neurons, here the R^2 scores are 0.81 and 0.68 for single and simultaneous movements, respectively. Remarkably, increasing the size of the neural network does *not* cause the system to degrade, as one observes is some other machine learning tasks, where this is explained with the network *overfitting* to the training data set.

For *single* movements, LDA gives much better results than linear regression, yielding an R^2 score of 0.70 compared to 0.56 for linear regression. The average accuracy of the LDA classifier is 78% on eight classes including *No Movement*, which is substantially better than chance level, but still means that more than 1 in 5 frames is classified wrongly; for these frames, the recognized prosthesis movement command is completely off (which can however be ameliorated with a smoothing postprocessing, see section IV). This lack of graceful degradation is a common property of many classifiers, it stems from the discreteness of the class space and may be very relevant in online applications, where the prosthesis user can adapt to errors of the ML system [10]. Still, from the fact that LDA (a simple linear classifier) attains an accuracy of almost 80%, we can deduce that the movement classes in feature space are quite well separated; we cover this topic further in section VI. We finally report that LDA performs much better on high-amplitude movements than on low-amplitude movements: On movement frames with normalized amplitude greater or equal 0.5, the average accuracy is 89%, whereas on movements with amplitude below 0.5 (excluding the *no movement* class), the accuracy is only 57%.

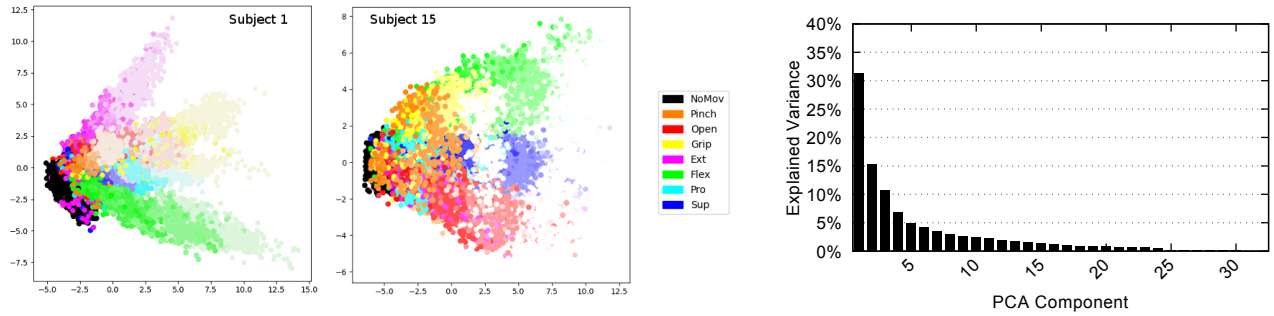
When training and testing with simultaneous movements, the R^2 scores are always lower than in the above single-movement case. A further analysis of the results shows that the score is decidedly better when testing is applied to the single movements in the test corpus. For example, with the reference NN, the average R^2 score is 0.86 on single movements and 0.70 on simultaneous movements. Yet, we obtain the encouraging result that regression on the simultaneous movements is possible despite the increased complexity of the task.

B. In-depth Analyses

Having established the neural network system, we turn to a further analysis of its performance. In a first step, we analyze differences between the subjects, and we aim at understanding how much the system overfits to the training data. Figure 3 shows a breakdown of training and test R^2 scores for the development subjects, considering two different network topologies (small and reference) and the corpora with and without simultaneous movements.

Subject 15, who is a prosthesis user, reaches low R^2 scores, in particular when using combined movements. Yet subject 12 is very good, despite likewise being an amputee. Thus even with the small amount of data considered so far, we can conclude that our method can be made to work not only on able-bodied subjects, but also for prosthesis wearers. In general, the variation between subjects is substantial; note that subject 12 had rather ample experience in recording data for ML based prosthesis control, whereas all other prosthesis subjects were novices.

We confirm that the larger (reference) network performs consistently better than the small one. Notably, the performance on the training data is not much higher than on the test data. Consider the single-movement corpus: The average R^2 scores of the small NN are 0.82 and 0.81 for training and test data, respectively: a difference of 1.2%. The reference NN achieves a training score of 0.93 and a test score of 0.87 (6.5% difference). Thus we can confirm some overfitting on the training data, in particular for the larger reference network, which is what was to be expected; yet the difference is *not* very large, compared to other challenging machine learning tasks. This would also explain why we did not observe an improvement when using Dropout: Dropout improves the



(a) Scatter plot of features in PCA space (two most relevant components), for a typical subject (#1) and a badly performing subject (#15). Lighter color means greater contraction strength.

(b) Explained variance per PCA component on features from all movements, averaged over development subjects.

Fig. 5: Analysis of the feature space, using standard Hudgins features

generalization capability of the network, yet in our case, generalization is already quite good.

As a final evaluation and future reference, we now consider the performance on diverse feature sets, always using the reference NN. Figure 4 shows average R^2 scores on all single features, as well as on the standard four-feature Hudgins set, and on a feature set which combines all seven features defined in section III. Augmenting the standard Hudgins features with Hahne's additional features does not yield any substantial benefit, thus confirming our choice of using the former as the basis of our experiments. In terms of single features, two of them (zero-crossing rate and slope sign change) perform badly in isolation. The other features perform more or less equal in the case of single movements, the picture is more complicated when we train and test on combined movements: here the MAV feature performs comparatively badly across all subjects. This result is clearly unexpected, however since feature optimization is not the main focus of this paper, we leave its further investigation to future work.

VI. System Analysis

Here, we analyse our trained systems, shedding some light on why our systems behave in the way they do. We draw on results from the previous section by always using the Hudgins features, and by considering the *reference* architecture (3 hidden layers, 128 neurons/layer) and occasionally the *small* architecture (1 hidden layer with 32 neurons).

A. Regression as a Linear Map

We first aim at understanding the most striking result from the previous section, namely, we ask why linear regression performs poorly on our task. It is helpful to consider the question from a theoretical perspective first, noting that linear regression is a linear mapping from the 32-dimensional feature space to the 7-dimensional "regression" space, with a bias which calibrates the regression output and which we can mostly ignore for this analysis. Each component of the regression vector is computed as the scalar product between the input feature vector and a *weight* vector. The entire mapping can also be written as matrix product, where the mapping matrix is formed by the stacked weight vectors.

It is sufficient to consider only the case of single movements. Take the first movement (*fine pinch*) as an example: The features corresponding to this movement need to be mapped to a one-dimensional subspace of the regression space, namely, to the space spanned by the target vector $(1, 0, 0, 0, 0, 0, 0)$. Clearly, the first component of the mapped features should reflect the strength of the contraction which the user performed.

It is usually not hard to find such a mapping for any single movement. Consider the left part of figure 5a, which shows a scatter plot of EMG features of the different movements for the rather typical subject 1, using a projection to the two most important principal components for visualization. The lightness of each data point corresponds to the contraction level (the lighter, the stronger). From the examples of the movements *hand extension* (EXT) and *hand flexion* (FLEX), one con clearly discern a direction along which the contraction level can be easily recovered; the same is true for the other movements, even if less discernible in the PCA space[2].

We can verify this observation by the following experiment: We filter the dataset so that it contains only samples of a *single* movement, and then train a linear regressor with one-dimensional output to recognize the contraction strength for this movement only. In this way, we remove any effects which may arise from the interference of several movements, which might overlap in feature space. Averaged over all seven movements (excluding the *no movement* class) and all seven development subjects, this yields an R^2 score of 0.81 on the test samples, which is almost as high as with the optimal nonlinear neural network system (0.87), and far higher than linear regression on the entire dataset (0.56).

Thus it is proven that the difficulty lies in finding a mapping which performs well for *all seven* movements simultaneously. In particular, for perfect regression, the weight vectors for the seven movements would have to be pairwise orthogonal (otherwise activity from one movement "leaks" into the other one). However, this condition alone is not enough to guarantee separation of movements: Assuming two movements m and n with corresponding weight vectors w_m and w_n, for good regression of movement m, its features need to have high

[2] *Hand flexion* and *extension* often have rather high amplitudes, so it is not a coincidence that they appear clearly in PCA space.

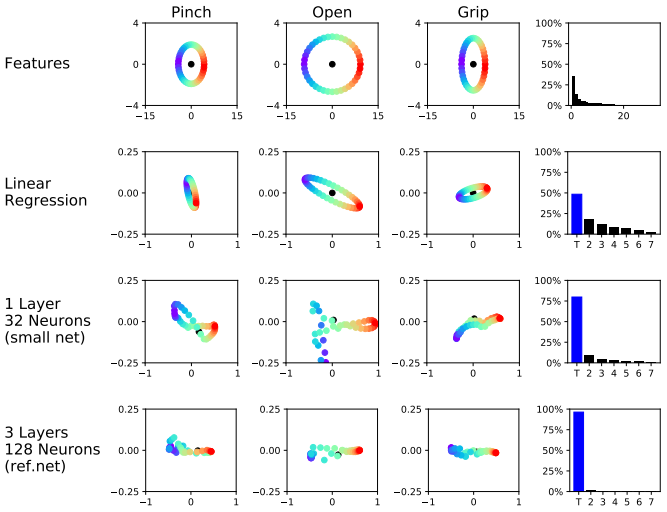


Fig. 6: Mapping of diverse movements of subject 7 in PCA space, for several regressors. Right column shows explained variances averaged over all movements.

variance along direction w_m and (almost) zero variance in direction w_n. Furthermore, the variation of the features in direction w_m must reflect the contraction strength.

Can such a condition be satisfied, at least approximately, for all movements? The low performance of the linear regression system demonstrates that this is not the case. Figure 5 indicates where the difficulties lie. First, while the features are 32-dimensional, 80% of the variance is found in an 8-dimensional subspace (as can be seen by summing up the values of the first 8 bars in figure 5b), thus greatly limiting the choice of possible weight vectors (this may have to do with the fact that eight EMG electrodes were used, we do not investigate this question here). Second, directions which optimally reflect the contraction strength are by no means orthogonal between movements, which can be visually discerned from figure 5a and rigorously concluded from the different performances of the single-movement and the full linear regressor.

From figure 5a, it also becomes clear that each single movement spans at least two dimensions in feature space. Making the simplifying assumption that the entire data spans an 8-dimensional space, and that each movement spans a two-dimensional subspace, it is clear that finding a perfect mapping would mean that for *each* movement, one direction needs to be mapped to the target, and another one to zero. Since the mapping is surjective from 8 to 7 dimensions, *all* the directions which are mapped to zero would have to be identical. From figure 5a, it is obvious that such a mapping is not possible. Still, when training the linear regression system, the best possible mapping is found, given the constraints. We have observed that this mapping tends to focus on three to four movements which are recognzied quite well, usually, hand flexion and extension are in this group. For the other movements, the regression output is almost constant, and the bias is calibrated such that the output command is approximately 0.4 . . . 0.5 (i.e. the mean value of the training target).

We finally note from figure 5a that at least for subject 1, the *classes* are indeed quite well separated, confirming our earlier observation derived from the LDA classification accuracy. Overlap occurs mostly at low contraction levels. From the case of low-performing subject 15, it can be seen that the movements can be much more intermingled than for subject 1; this is however not a typical case and can be prevented by suitable user training [4], [44], [45].

B. Nonlinear Mappings solve the Problem

Having established why linear regression cannot solve the given task very well, we turn our consideration to neural networks. Remember that the samples of a movement class in feature space span at least two dimensions.

We aim at displaying the mapping graphically. For this purpose, we plot a two-dimensional space at feature or regression level, where we filter the dataset to contain only a single movement. At feature level, we obtain the plot space from the two first PCA components, computed from all samples of the respective movement. At regression level, the x-axis is the target dimension, the y-axis corresponds to the first PCA component computed on the remaining components of the regression space (the "spurious" space).

In feature space, we approximate each class distribution with an ellipse whose axes correspond to the two main PCA directions for this movement, scaled by the respective explained variance. We orient the ellipse such that samples with low contraction strength are at the left (blue color), and samples with high contraction strength are at the right (red color). Note that this approximation is not very accurate, since the class distributions are *not* Gaussian – still, we will draw conclusions from examining how this ellipse is transformed by different NN architectures and by linear regression.

We display how the class approximations are mapped by diverse systems in figure 6, taking three movements (namely the grips, and hand opening) of subject 7 as a typical example. All systems were trained on single movements only. The first row of the figure shows the ellipses in feature space, note that they are aligned to the coordinate axes since both the coordinate axes and the axes of the ellipse are taken from the PCA of the features of that class. The size of the ellipse differs by class and represents the feature amplitude.

The lower three rows show how these ellipses are transformed by linear regression, and by two NNs (small and reference). In the linear case, we see that the ellipses are scaled and rotated, but retain their elliptical form. It is instructive to compare classes: For *Hand Open*, we see that the mapped features are close to one-dimensional in the regression space, and this one dimension is almost aligned with the target direction (i.e. the x-axis). However, the nonzero variance along the y-axis indicates that some spurious movements are generated, and also that they are correlated with the main movement; one could now analyze the PCA in the "spurious" space to obtain further understanding about which error is made by the mapping. The same applies for the *Key Grip* movement, but the variance in the target direction is not as large as it should be (the regression needs to span an output range from 0.0 to 0.9, note that the mean output is calibrated

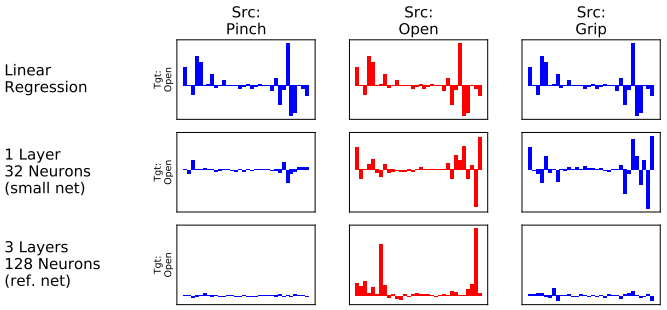


Fig. 7: Linearization of the mapping for the target movement *Hand Open* in different regions, for several regressors. Each panel shows the linearization of the mapping as a 32-dimensional weight vector in feature space, each column refers to the region of the features for a specific source movement.

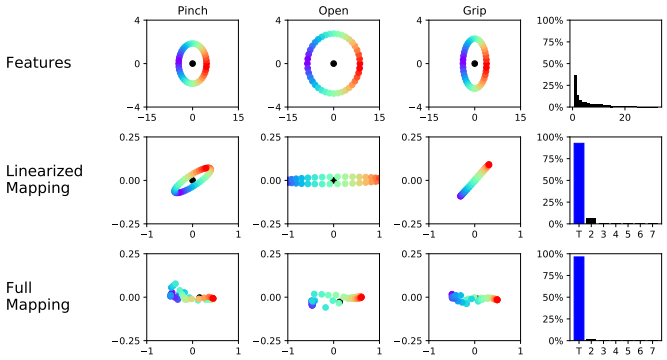


Fig. 8: *Linearized* mapping of diverse movements of subject 7 in PCA space, for the reference neural network. Right column shows explained variances averaged over all movements.

by the NN bias), indicating that the strength of this movement command is underestimated. A user would thus find it difficult to achieve full force or velocity for the movement *Key Grip*. In the case of *Fine Pinch*, we see that more activity is created for spurious movements than for the target class. The R^2 score (over all classes) for linear regression is 0.53.

The small NN achieves a substantially better R^2 score, namely 0.77. This is reflected in the processing of the movement classes. First, note that *Key Grip* now has substantially more variance in the target direction. The ellipses get substantially distorted, a common pattern is the compression of the second axis of the ellipse (i.e. the first "spurious" direction) for *strong contractions* (red-colored). This means that movements with strong contraction levels are well regressed to their target classes, with little spurious activity. For low contraction strengths, we see a more diverse pattern (but this may also be an artifact of the slightly incorrect modelling of classes as two-dimensional Gaussians: we have observed that the ellipses overestimate the variance of the features at low contraction levels). Finally, the reference NN reaches an R^2 score of 0.85, and we see that the mapping is much better in particular for lower contraction strengths (blue-colored). In both cases, we also see that the contraction strength (from blue to red) is well regressed to the target dimension.

The rightmost column of figure 6 displays the explained variance by PCA component in feature space (first row) and in the target spaces (lower rows), averaged over *all* movements. As before, in regression space, the first bar ('T', blue-colored) corresponds to the target dimension, and the other bars represent PCA components along spurious dimensions. Clearly, the larger the NN architecture, the more regression-level variance is concentrated on the target dimension.

We can also confirm our observations numerically: Averaged over all development subjects and all movements, the linear regressor maps 48% of the total variance in regression space to the target class, the remainder is mapped to spurious movements. For the small NN, in contrast, 80% of the variance in regression space is found along the target direction, and for the reference NN, this number rises to 96%.

C. The NN exhibits Localized Behavior

How does the NN achieve its mapping power? A major restriction of linear mappings is *globality*, i.e. they behave identially in their entire domain of definition. Nonlinear mappings can instead be very different in different regions.

The NN always computes a mapping from the 32-dimensional feature space to the 7-dimensional regression space, and we can analyze it by computing its *linearization* at specific points in feature space. This linearization takes the form of a 32-dimensional weight vector and is computed as the gradient of the NN at that point, ReLU networks specifically are piecewise linear and thus allow an even easier computation.

Figure 7 shows linearizations of the regression mapping in different regions, taking three movements of subject 7 as the usual typical example. As before, all systems were trained on single movements. Linearizations were computed for the "target" movement *Hand Open*, the reference points for the linearization were the sample means of the features of the "source" movements *Fine Pinch*, *Hand Open*, and *Key Grip*. The linearizations are displayed as bar plots of weight vectors, note that as before, we can ignore the biases. Three systems were considered, namely linear regression and the small and reference NNs.

Clearly, in the linear case the mappings for each movement class are identical everywhere. The behavior of the neural networks is very consistent: the mapping is approximately zero where target and source movements do not match (blue-colored plots); the larger network achieves this optimal mapping more accurately than the smaller one. Taking the square Euclidean norms of the linearized mappings, we numerically verify this observation by taking means over non-matching respectively matching movements, and then computing the ratio between these two numbers. Averaged over all development subjects, this ratio is 0.07 for the small NN and 0.08 for the reference NN in the case of single movement systems, when systems are trained on single and simultaneous movements, the ratios are 0.45 and 0.15 respectively, indicating somewhat different behaviors of these two setups. (By definition, the ratio is 1.0 for linear systems.) We observed this pattern in a variety of configurations and for all subjects. There

System	R^2 score	p-value
Only single movements		
Linear Regression	0.47	
LDA	0.61	$1.09 \times 10^{-2}*$
1 HL, 8 neurons (very small)	0.66	$3.34 \times 10^{-2}*$
1 HL, 32 neurons (small)	0.78	$3.70 \times 10^{-6}*$
3 HL, 128 neurons (ref.)	0.86	$6.08 \times 10^{-4}*$
Single and simultaneous movements		
Linear Regression	0.42	
1 HL, 8 neurons (very small)	0.54	$3.90 \times 10^{-6}*$
1 HL, 32 neurons (small)	0.63	$5.97 \times 10^{-5}*$
3 HL, 128 neurons (ref.)	0.74	$3.72 \times 10^{-6}*$

TABLE II: R^2 scores on evaluation subjects for diverse configurations. p-values indicate significance of improvement w.r.t. the previous line in the table, improvements marked with $*$ are significant ($p < 0.05$).

are occasional exceptions where the mapping for a specific movement responds also to (usually) one different movement; we believe that this indicates an overlap of movements in feature space.

D. Class distributions are nonlinearly distorted

Finally, we ask whether the locality of the nonlinear mapping completely explains its superiority over linear regression. While figure 7 clearly shows that non-matching movements are approximately mapped to zero, strictly speaking this only holds for the *mean* of the respective classes in feature space, which is where we computed the linearization. Also, we do not know much about how the features of the *matching* movements are transformed.

It turns out that the nonlinear distortion of each class, which is very evident in figure 6, plays a major role. We verify this by comparing the NN mapping with its linearization in the same way as we did in section VI-B, i.e. by representing each movement class with an ellipse and displaying the transformation it undergoes. The result is displayed in figure 8, again taking subject 7 (without combined movements) and the reference NN as example.

From the figure, it is clear that the linearized mapping and the full mapping are different, one sees this particularly in the case of the movements *Fine Pinch* and *Key Grip*. Thus, the inner distortion of each movement class clearly plays a role for performing an optimal mapping. Yet, already from the linearized mapping, we discern the dimensionality reduction between the input features and the regression output. As in figure 6, the rightmost column displays the explained variance in the diverse spaces, averaged over all movements. As usual, we report results averaged over the entire development set: For the linearized mapping, 90% of the variance is mapped to the target direction, for the full mapping, this number rises to 96%, which means a reduction of spurious movements by about 60% relative. With simultaneous movements, the numbers are slightly lower, namely 87% and 93% explained variance in the target dimension, respectively. There is a lot of variance between subjects: in some cases, the linearized mapping has almost the same properties as the full mapping.

System	Target Variance Ratio	p-value
Only single movements		
Linear Regression	0.46	
1 HL, 32 neurons (small)	0.81	$3.68 \times 10^{-7}*$
3 HL, 128 neurons (ref.)	0.94	$5.57 \times 10^{-5}*$
Single and simultaneous movements		
Linear Regression	0.41	
1 HL, 32 neurons (small)	0.71	$1.01 \times 10^{-6}*$
3 HL, 128 neurons (ref.)	0.92	$1.08 \times 10^{-4}*$

(a) Explained variance in target direction, p-value for improvement w.r.t. previous row

System	Norm Ratio	p-value
Only single movements		
Linear Regression	1.00	
1 HL, 32 neurons (small)	0.30	$7.71 \times 10^{-7}*$
3 HL, 128 neurons (ref.)	0.10	$9.10 \times 10^{-4}*$
Single and simultaneous movements		
Linear Regression	1.00	
1 HL, 32 neurons (small)	0.44	$3.46 \times 10^{-7}*$
3 HL, 128 neurons (ref.)	0.12	$2.18 \times 10^{-4}*$

(b) Ratio of average squared norms of linearized nonmatching and matching movements, p-value for improvement w.r.t. previous row

System	Target Variance Ratio		p-value
	Linear	Full	
Only single movements			
Linear Regression	0.46	0.46	—
1 HL, 32 neurons (small)	0.72	0.81	$2.43 \times 10^{-3}*$
3 HL, 128 neurons (ref.)	0.87	0.94	9.90×10^{-2}
Single and simultaneous movements			
Linear Regression	0.41	0.41	—
1 HL, 32 neurons (small)	0.56	0.71	$1.83 \times 10^{-4}*$
3 HL, 128 neurons (ref.)	0.84	0.92	$3.82 \times 10^{-2}*$

(c) Explained variance in target direction for linearization and full mapping, p-value w.r.t. improvement between columns

TABLE III: Evaluation of several indicators used for analyzing and comparing our systems. Differences marked with $*$ are significant ($p < 0.05$).

VII. Evaluation

In this section, we rigorously evaluate the results from both section V and VI. We start by considering the R^2 scores on the evaluation subjects (see section III); comparing the two linear baselines and three neural network architectures (including a "very small" net with only 8 hidden neurons, in order to be in line with prior work [13], [15]–[17]). From table II, it becomes clear that the NN systems indeed improve substantially over both linear systems, and that increasing the size of the NN brings further improvement. We statistically validated the improvements by performing a t-test (one-tailed for paired samples) between each pair of subsequent results: All improvements marked with $*$ are significant ($p < 0.05$). The reference NN achieves an R^2 score of 0.86 on single movements, and 0.74 on single and simultaneous movements; this result is not very different from the one obtained on the development subjects.

We now statistically validate the results of the system analysis, considering again the usual three architectures, namely Linear Regression and the small and reference neural networks, both with and without simultaneous movements, with all values averaged over the evaluation subjects. We formu-

late the following three hypotheses: 1) the larger the NN architecture, the greater the percentage of explained variance in the target direction (see section VI-B); 2) the larger the NN architecture, the lower the ratio of squared mapping norms between nonmatching source and target movements (see section VI-C); 3) the linearized mapping maps less variance to the target direction than the full mapping (see section VI-D). All differences are statistically validated using a t-test (one-tailed for paired samples), at a significance level of 0.05.

From table IIIa, we see that our hypothesis 1 is valid: for linear regression, the target direction accounts for 46%/41% of the explained variance in regression space without and with augmented simultaneous movements, respectively; with the reference network, these numbers rise to 94% resp. 92%. The small NN is worse than the reference NN, but far better than linear regression. All improvements are significant.

Table IIIb likewise shows that hypothesis 2 holds: in the linear case, the mapping for any movement remains the same in all areas of the feature space (hence a ratio of 1.0 between areas of matching and nonmatching movements), whereas this ratio is far lower for the NN systems. All ratio reductions between systems are highly significant.

Finally in table IIIc, we compare the linearized and full mappings, always for the usual three systems. It is not trivial to find a metric for the difference between mappings, so we resort to the metric which we successfully used before, namely, we evaluate how much of the variance in regression space is concentrated on the target direction; this is fundamentally a measure of how little spurious activity is generated. In all cases, this value is higher for the full mapping than for the linearized mapping, the difference is significant in three out of four cases – thus, not only the locality of the nonlinear mapping, but also the nonlinear inner distortion of each class play a substantial role.

VIII. Conclusion

In this paper, we have thoroughly investigated simultaneous proportional prosthesis control, based on nonlinear regression implemented with feedforward neural networks. Our key result is twofold: First, we have shown that sufficiently large architectures solve the task better than both linear systems and small-scale neural networks. Growing the NN in number and size of the layers, the quality of the mapping, given by the R^2 score, converges to a stable value; this is a good result because it means that optimizing such a system to a new task (e.g. with different movements, or different EMG recording setup) should not be too difficult. Second, we have analyzed the behavior of the NN and compared it to the linear regression baseline, explaining why the latter is *theoretically* inferior to the former, given the data at hand. To the best of our knowledge, this is the first analysis of this kind in the field of prosthesis control. Based on the theoretical analysis, we believe that our results will also hold with different regression-based setups (for example, with different movements, or a larger amount of training data).

Since classification-based systems are conceptually unsuitable for controlling simultaneous movements, we have not investigated the extent to which our results transfer to neural networks trained for classification. Since classification-based systems using linear methods are commercially relevant [9], this could be an interesting extension of our research. Finally, we remark that we observed the surprising anomaly that the well-known MAV feature performs badly on simultaneous movements; we leave the explanation, and mitigation, of this result to future work.

Acknowledgment

The authors with to thank Sebastian Amsüss for providing the data corpus and valuable advice, and Alessio Murgia and Raoul Bongers for helpful feedback and discussions.

References

[1] R. Merletti and P. A. Parker, Eds., *Electromyography - Physiology, Engineering, and Noninvasive Applications*. John Wiley and Sons, Inc., 2004.

[2] A. D. Roche *et al.*, "Prosthetic Myoelectric Control Strategies: A Clinical Perspective," *Current Surgery Reports*, vol. 2, no. 44, 2014.

[3] A. W. Franzke *et al.*, "Users' and Therapists' Perceptions of Myoelectric Multi-function Upper Limb Prostheses with Conventional and Pattern Recognition Control," *PLoS ONE*, vol. 14, no. 8, p. e0220899, 2019.

[4] M. B. Kristoffersen *et al.*, "Serious Gaming to Generate Separated and Consistent EMG Patterns in Pattern-recognition Prosthesis Control," *Biomedical Signal Processing and Control*, vol. 62, p. 102140, 2020.

[5] T. A. Kuiken *et al.*, "A Comparison of Pattern Recognition Control and Direct Control of a Multiple Degree-of-Freedom Transradial Prosthesis," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 4, p. 2100508, 2016.

[6] I. Vujaklija *et al.*, *Wearable Robotics: Challenges and Trends*. Springer, 2017, ch. Clinical Evaluation of a Socket-Ready Naturally Controlled Multichannel Upper Limb Prosthetic System, pp. 3 – 7.

[7] L. Resnik *et al.*, "Evaluation of EMG Pattern Recognition for Upper Limb Prosthesis Control: a Case Study in Comparison with Direct Myoelectric Control," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, no. 23, 2018.

[8] A. W. Shehata *et al.*, "Machine Learning for the Control of Prosthetic Arms: Using Electromyographic Signals for Improved Performance," *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 46 – 53, 2021.

[9] O. Bock, "Myo Plus Pattern Recognition," available from: https://www.ottobock.co.uk/prosthetics/upper-limb-prosthetics/product-systems/myo-plus/.

[10] J. M. Hahne *et al.*, "User Adaptation in Myoelectric Man-Machine Interfaces," *Scientific Reports*, vol. 7, no. 4437, 2017.

[11] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1 – 127, 2009.

[12] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

[13] J. L. G. Nielsen *et al.*, "Simultaneous and Proportional Force Estimation for Multifunction Myoelectric Prostheses Using Mirrored Bilateral Training," *IEEE Transaction on Biomedical Engineering*, vol. 58, no. 3, pp. 681 – 688, 2011.

[14] S. Muceli and D. Farina, "Simultaneous and Proportional Estimation of Hand Kinematics From EMG During Mirrored Movements at Multiple Degrees-of-Freedom," *IEEE Transactions on Neural Systems And Rehabilitation Engineering*, vol. 20, no. 3, pp. 371 – 378, 2012.

[15] N. Jiang *et al.*, "EMG-based Simultaneous and Proportional Estimation of Wrist/hand Kinematics in Uni-lateral Trans-radial Amputees," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 42, 2012.

[16] J. M. Hahne *et al.*, "Linear and Nonlinear Regression Techniques for Simultaneous and Proportional Myoelectric Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 2, pp. 269 – 279, 2014.

[17] A. Ameri *et al.*, "Support Vector Regression for Improved Real-Time, Simultaneous Myoelectric Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 6, pp. 1198 – 1209, 2014.

[18] A. Ameri *et al.*, "Regression Convolutional Neural Network for Improved Simultaneous EMG Control," *Journal of Neural Engineering*, vol. 16, p. 036015, 2019.

[19] U. Côté-Allard *et al.*, "Unsupervised Domain Adversarial Self-Calibration for Electromyography-Based Gesture Recognition," *IEEE Access*, vol. 8, pp. 177 941 – 177 955, 2020.

[20] Y. Teh and L. Hargrove, "Using Latent Representations of Muscle Activation Patterns to Mitigate Myoelectric Interface Noise," in *Proc. NER*, 2021, pp. 1148 – 1151.

[21] A. d'Avella *et al.*, "Control of Fast-Reaching Movements by Muscle Synergy Combinations," *Journal of Neuroscience*, vol. 26, no. 30, pp. 7791 – 7810, 2006.

[22] S. Amsüss *et al.*, "Self-Correcting Pattern Recognition System of Surface EMG Signals for Upper Limb Prosthesis Control," *IEEE Trans. Biom. Eng.*, vol. 61, no. 4, pp. 1167 – 1176, 2014.

[23] N. Jiang *et al.*, "Is Accurate Mapping of EMG Signals on Kinematics Needed for Precise Online Myoelectric Control?" *IEEE Transactions on Neural Systems And Rehabilitation Engineering*, vol. 22, no. 3, pp. 549 – 558, 2014.

[24] M. B. Kristoffersen *et al.*, "User Training for Machine Learning Controlled Upper Limb Prostheses: A Serious Game Approach," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 32, 2021.

[25] R. Reiter, "Eine neue Electrokunsthand," *Grenzgebiete der Medizin*, vol. 4, no. 133, 1948.

[26] D. S. Childress, "Historical Aspects of Powered Limb Prostheses," *Clinical J Prosthetics & Orthotics*, vol. 9, no. 1, pp. 2 – 13, 1985.

[27] K. Englehart and B. Hudgins, "A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848 – 854, 2003.

[28] A. J. Young *et al.*, "Classification of Simultaneous Movements Using Surface EMG Pattern Recognition," *IEEE Transactions of Biomedical Engineering*, vol. 60, no. 5, pp. 1250 – 1258, 2013.

[29] B. Peerdeman *et al.*, "Myoelectric Forearm Prostheses: State of the Art from a User-centered Perspective," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 719 – 738, 2011.

[30] E. Scheme *et al.*, "Motion Normalized Proportional Control for Improved Pattern Recognition-Based Myoelectric Control," *IEEE Transactions on Neural Systems And Rehabilitation Engineering*, vol. 22, no. 1, pp. 149 – 157, 2014.

[31] S. Amsuess *et al.*, "A Multi-Class Proportional Myocontrol Algorithm for Upper Limb Prosthesis Control: Validation in Real-Life Scenarios on Amputees," *IEEE Transactions on Neural Systems And Rehabilitation Engineering*, vol. 23, no. 5, pp. 827 – 836, 2015.

[32] M. M.-C. Vidovic *et al.*, "Improving the Robustness of Myoelectric Pattern Recognition for Upper Limb Prostheses by Covariate Shift Adaptation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 9, pp. 961 – 970, 2016.

[33] S. Muceli *et al.*, "Multichannel Surface EMG based Estimation of Bilateral Hand Kinematics during Movements at Multiple Degrees of Freedom," in *Proc. EMBC*, 2010, pp. 6066 – 6069.

[34] N. Jiang *et al.*, "Extracting Simultaneous and Proportional Neural Control Information for Multiple-DOF Prostheses From the Surface Electromyographic Signal," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1070 – 1080, 2009.

[35] X. Zhai *et al.*, "Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network," *Frontiers in Neuroscience*, vol. 11, p. 379, 2017.

[36] J. Liu *et al.*, "Towards Zero Retraining for Myoelectric Control Based on Common Model Component Analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 4, pp. 444 – 454, 2016.

[37] K. Proroković *et al.*, "Meta-Learning for Recalibration of EMG-Based Upper Limb Prostheses," in *Proc. AutoML*, 2020.

[38] S. Muceli *et al.*, "Extracting Signals Robust to Electrode Number and Shift for Online Simultaneous and Proportional Myoelectric Control by Factorization Algorithms," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 3, pp. 623 – 633, 2014.

[39] B. Hudgins *et al.*, "A New Strategy for Multifunction Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 40, pp. 82 – 94, 1993.

[40] P. Golik *et al.*, "Convolutional Neural Networks for Acoustic Modeling of Raw Time Signal in LVCSR," in *Proc. Interspeech*, 2015, pp. 26 – 30.

[41] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.

[42] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. NIPS*, 2019, pp. 8024–8035.

[43] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825 – 2830, 2011.

[44] M. A. Powell and N. V. Thakor, "A Training Strategy for Learning Pattern Recognition Control for Myoelectric Prostheses," *Journal of Prosthetics and Orthotics*, vol. 25, no. 1, pp. 30 – 41, 2013.

[45] M. B. Kristoffersen *et al.*, "The Effect of Feedback During Training Sessions on Learning Pattern-Recognition-Based Prosthesis Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 10, pp. 2087 – 2096, 2019.