Myoelectric Control of isometric Force

Applied Case Studies of Machine Learning and Deep Learning in Key Areas 2

Cristian Brugnara, Zeno Darani, Tommaso Sommaruga

D3A - SUPSI

SUPSI

May 11, 2025

Contents

This report is a summary of the project, and due to the page limit (5 without titles, table of contents, and references; but we couldn't do less than 6, but there are many plots and tables), it's an approximation of the work done. For a more comprehensive understanding of the task, we suggest the fully commented notebooks with many more visualizations and experiments, including SoTA analysis and ICA trial.

1 Introduction and Dataset Description

The project aimed to predict the direction and magnitude of a force applied to a handle, given the EMG signals of eight muscles. EMG signals are processed in different ways depending on the task. In our case, we were mostly interested in a slow-changing force, thus, our preprocessing and feature engineering were done keeping that in mind. We were also requested to analyze the performance of Polynomial Regression using different degrees and create a robust cross-validation pipeline. After doing that, more complex models, ranging from tree ensembles to Neural Networks, were employed.

The data was compressed into a MATLAB array, and extracting it was convoluted at the start. The dataset consisted in 8 EMG signals related to different arm muscles, sampled at 1000 Hz, for each combination of direction (12) and repetitions (4). For each group, there were also two series corresponding to the force applied along the X and Y coordinates, sampled at 100 Hz. Specifically the observed muscles were: Anterior Deltoid (aDEL), Posterior Deltoid (pDEL), Biceps long head (lBIC), Biceps short head (sBIC), Triceps (TRI), Brachioradialis (BRA), Major Pectoralis (mPEC) As reported in the paper of the experiment [LS18], it is difficult to obtain the EMG of recordings for different portions of the Triceps. They chose to create two series from the same EMG signal, one for long (lTRI) and the other for short heads (sTRI) of the triceps. Both EMGs and forces although with different frequencies, were measured over around 4 seconds.

---

2 Preprocessing and Simple EDA

Before training our models, we cleaned and transformed the signals, estimated noise levels, and explored the data through visualizations to improve quality and performance.

2.1 Signal Redundancy Removal

First, we checked the correlation between the signals. As reported in the previous section, lTRI and sTRI have a correlation of 1 because they are the same signal. Since our approach is purely machine learning based, we removed the redundancy by dropping the sTRI signal.

2.2 SNR Estimation via PSD

To support the preprocessing, we also found a way to estimate the Signal-to-Noise of the signals. Since EMG is a stochastic signal and we had many of them, we couldn't select a segment of noise with inspection for each one. Thus, we searched for a way to estimate it based on PSD (Power Spectral Density).

First, computing its short-time Fourier transform to get a power spectral density. A one-dimensional minimum filter is then applied across each frequency bin's time course to track the background noise floor. Within the muscle-activity band, the average PSD gives total power, and the filtered signal gives noise power. Subtracting the latter from the former yields a signal power estimate.

2.3 Filtering

We know that EMGs are analyzed in the frequency range 20 Hz and 500 Hz, but since our Nyquist frequency is 500 Hz, we applied a band-pass filter in the range 20–450 Hz, implemented with a Butterworth filter of order 4. Then, a Notch filter was applied to attenuate the 50 Hz spikes due to powerline interference. After this step, the SNR increased from 17 to 18.

2.4 Rectification

We proceeded with Full-Wave Rectification, a popular technique to process EMG signals in which we take their absolute value. This step is done to prevent the signal from balancing out when computing certain statistics, reducing the noise by smoothing, finding more informative features, and more.

2.5 Envelope Extraction and Downsampling

Finally, two things needed to be considered. Since our force signals were sampled at 100 Hz, we had to downsample the EMGs by a factor of 10. The Nyquist-Shannon sampling theorem applied to downsampling states that our signal must not have frequency content over the new Nyquist frequency, meaning half of the new frequency. In other words we needed to low-pass the signal with a cut-off of at least 50 Hz. However, to better model the force, we wanted to extract a linear activation envelope, which is obtained by heavily smoothing the signals. This envelope should be strongly correlated with isometric force signals, which tend to change less rapidly than EMG signals, on top of reducing noise and uncorrelated spikes.

Thus, we decided to apply a low-pass filter at 10 Hz, in order to both extract the envelope and respect the

Nyquist-Shannon theorem. Finally, we downsampled the data by decimating it by 10, and each combination now had around 400 samples.

2.6 Visualization and Dimensionality Reduction

The process was supported with many visualizations of the series on top of PCA and t-SNE plots.

---

3 Spatial Muscle Analysis

This section analyzes the directional activation of key arm muscles using EMG data, visualized via polar plots and histograms, to understand their functional roles in movement. A summary of these characteristics is presented in Table 1.

Table 1: Muscle Activation Summary

Muscle	Peak Dir. (°)	Mean Act.	CV	Primary Activation
aDEL	180	0.005	0.380	Posterior
pDEL	150	0.019	1.140	Posterior
lBIC	210	0.015	0.756	Posterior
sBIC	30	0.022	0.745	Anterior
lTRI	210	0.025	1.043	Posterior
BRA	0	0.017	1.018	Anterior
mPEC	330	0.008	0.811	Anterior

3.0.1 Posteriorly Activated Muscles

Posterior muscles, active during pulling motions, shoulder extension, or backward elbow flexion, generally peak between 150° and 210°. The **aDEL (Anterior Deltoid)** peaks at 180°, while the **pDEL (Posterior Deltoid)** shows significant variability at 150°. Both the **lBIC (Long Head of Biceps)** and **lTRI (Long Head of Triceps)** peak at 210°, with the latter exhibiting high variability, indicating flexible recruitment based on task demands.

3.0.2 Anteriorly Activated Muscles

Anterior muscles, engaged in pushing, reaching, shoulder flexion, and forward elbow extension, activate around 0° to 60° and 330°. The **sBIC (Short Head of Biceps)** peaks at 30°, while the **BRA (Brachialis)** specializes at 0°, and the **mPEC (Middle Pectoralis)** peaks at 330°. Muscles like **pDEL** and **lTRI** show higher variability, while **BRA** demonstrates a more focused activation, indicating specialized roles in movement.

3.1 Implications for Motor Control and Myoprocessor Models

These directional activation patterns are essential for improving EMG-driven models, such as myoprocessors, which help predict joint torques and motor strategies. Understanding muscle variability aids in personalizing assistive and rehabilitative technologies, ensuring more effective, individualized therapeutic interventions.

---

4 Feature Extraction

An important part of the project was extracting features from the signals. While raw signals often carry strong information about the dependent variables by themselves, or are processed by powerful models independently, extracting features from them, also considering prior information about the task, is usually a great idea. Our extraction involved the use of sliding windows to embed each feature with some temporal information and was targeted at the three domains we studied: Time, Frequency, and Time-Frequency.

4.1 Time-Domain Features

The features in the time-domain were mostly based on the amplitude of the signal evolving with time. They included the mean of the signal, the Root Mean Square which is sensitive to larger values, the variance to describe the distribution, and the difference between the last and first value in the window, to get an estimation of the window trend.

4.2 Frequency-Domain Features

For each sliding window, we apply Welch's method (using a Hamming window) to estimate the PSD of the signal. Welch's technique breaks the window into more overlapping segments, computes a periodogram for each, and then averages them to get a smoother, lower-variance PSD estimate than a single FFT. From this, we can extract information such as the mean and median of the frequency, the total power, and the variance.

4.3 Time-Frequency Features

For each window, we perform a Short-Time Fourier Transform, which also creates overlapping segments from it, then computes an FFT on each. From the matrix we can obtain the spectrogram, from which we extract features. These features are supposed to reflect both how much energy is present and how it's distributed over frequency in a window.

---

5 Modelling

We were asked to apply polynomial regression to predict the force direction and magnitude. For each model, multiple degrees of polynomials were compared, although this step was often limited by the time complexity. The test split was done by isolating the 4th repetition, as it was recorded last, so it made sense to use it to evaluate the predictions. The validation scheme used was a GroupKFold CV performed on the remaining repetitions. Thus, we obtained three folds in which the model was trained on two repetitions to predict the remaining one. As 3 is a generally lower number of folds, the scheme may not be as robust, but it was the only way to preserve the repetition-consistency we defined. With a bigger dataset with many patients and repetitions, we could do a GroupKFold on the patients to obtain generalizable models.

5.1 Raw Signals

The raw signals after preprocessing alone were used to perform the regression.

Regularization	Degree	CV avg RMSE	CV avg R^2 score
None	1	6.76	0.81
None	2	5.16	0.89
None	3	5.13	0.89
None	4	5.47	0.88
None	5	7.00	0.78
Ridge	1	6.76	0.81
Ridge	2	5.16	0.89
Ridge	3	5.08	0.89
Ridge	4	5.04	0.89
Ridge	5	7.00	0.78
Lasso	1	7.36	0.77
Lasso	2	7.43	0.77
Lasso	3	7.21	0.78
Lasso	4	7.00	0.79

Table 2: Raw signals polynomial regression results without PCA

These first results showed promising results and good predictive capabilities even with the raw signals and their interactions. In general we report that L2 and non-regularized regression were almost equivalent, showing the best results with degrees 3 and 4. The LASSO model was much worse. From the best model, we extracted the test residuals to do some analysis. Even the best model was making very noisy predictions, and while the trend of the true force was followed closely, the smoothness wasn't. First, we checked the normality of the residuals, a key assumption of the model, and they showed an almost perfect normal distribution, meaning that the inference was reliable. The check for residual homoscedasticity, meaning constant variance across the levels of the independent variable. We analyzed it using the Breusch-Pagan test, and the hypothesis was rejected with a very small p-value for both forces. While this is not too important in a typical ML pipeline, and our scores were good, it suggests that the model is missing some type of structure that will hopefully be accounted for by the more complex models. By re-running cross-validation, we also extracted the aggregated validation residuals, this time to do bootstrapping, ie. obtaining uncertainty estimates from the quantiles of the errors. The usual way to do this is using the train residuals, but using validation ones makes the interval broader and more general. In general, in the healthcare field, it's always good to have interpretable models that can quantify their uncertainty too. Bootstrapping is an attempt at doing so for a linear model. A good sign was that the ground truth laid within the quantile range at every point, and it would have been true even for a narrower range.

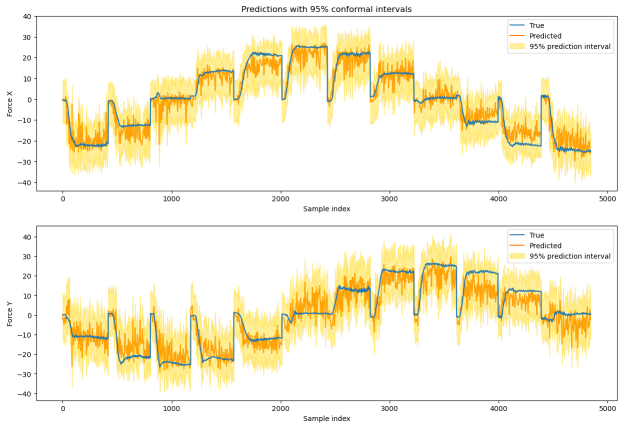

Figure 1: Ridge Predictions with 95% prediction interval over the 12 directions of repetition 4

5.2 Time-Frequency features and PCA

For the following models, we exploited the time-frequency domain's features extracted by the EMGs and used PCA as a dimensionality reduction method to make the cross-validation feasible. Since the number of features generated by the polynomial approach slows down fitting time by a lot, we chose to first apply a lower variance PCA (t = 0.7) and got around 10 features. Then, after the polynomial generation, we apply a more concessive PCA (t = 0.9) and obtain over 30 features.

Since the scores decreased with respect to the previous approach, we ran the cross-validation, but this time we kept more components in the first PCA application (t = 0.9), and by using lower degrees, we didn't need the second PCA. The better results indicate that PCA has a strong impact on the performance of the models, and it's likely more important to keep more features at the start rather than many of their interactions and polynomials. This makes sense as different sources of variance are usually better for a supervised task. The residuals of these models had very similar behaviors as the raw signal models. But while the prediction interval helped, the prediction followed the trend of force X less well, although the prediction seemed to have weaker high frequencies, being smoother.

We also tried feature selection methods instead of dimensionality reduction with PCA, we coded Permutation Feature Importance but the running time was significant, so we went back to a simpler method, Select K Best, using a varying k (mostly 100), while the performance didn't improve, it wasn't significantly worse proving that a good performance can be achieved even with much less features.

Regularization	First PCA variance	Degree	CV avg RMSE	CV avg R2 score
None	0.7	1	6.94	0.78
None	0.7	2	8.33	0.71
None	0.7	3	11.24	0.39
None	0.7	4	14.20	0.02
None	0.9	1	6.38	0.81
None	0.9	2	5.34	0.88
Ridge	0.9	1	6.37	0.81
Ridge	0.9	2	5.33	0.88

Table 3: Raw signals polynomial regression results

5.3 Advanced Models using the Extracted Features

We tried a lot of advanced models in order to check if there is still room for improvement using polynomial features and feature selection as described in the previous section.

- Tree-Based Models: This section includes many ensemble tree-based methods, from parallel methods like Random Forests and ExtraTrees, to boosting methods like XGBoost, CatBoost, and LightGBM.
- Neural Networks: A simple MLP from sklearn, and a custom Feed-Forward NN using keras.
- Support Vector Regressor, a method to fit a function to the data using a tolerance margin ϵ.

Combining this with the linear models used earlier, we used a large quantity of model categories to see if one has advantages over the others. Each of the advanced models was optimized with a Grid Search that took a long time due to the number of models and their complexity. A large part of the models obtained very good results, which were considerably better than the ones obtained in the previous section. The best model was the ExtraTree which obtained an R-squared of . Compared to the linear models, the new model learned well to model smooth transitions from rest to steady-state forces, in most directions, as well as the shape and amplitude during steady states. Rapid transitions and direction reversals, particularly in Force Y, are harder to follow.

Even if we measure good performances on the test set, the performances are not equal for all directions. In particular, we noticed that the regressor struggles

when one of the force components is close to zero, meaning that the arm is pushing/pulling perfectly vertical or horizontal; the worst case is Direction 11.

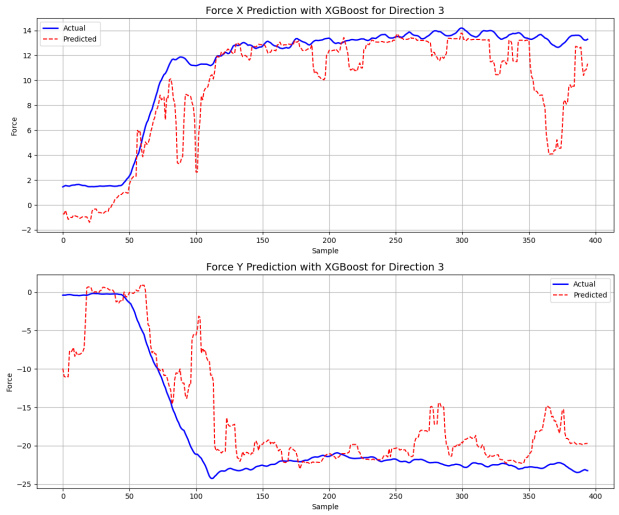

Figure 2: XGBoost Direction 3 prediction

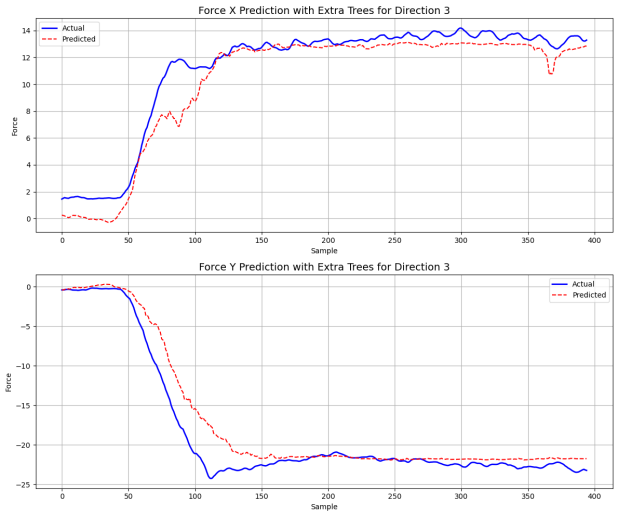

Figure 3: Extra Trees for Direction 3

The models in this section seemed to favor time-domain features (mean, RMS) over frequency domain measures, but the current model shows more even distribution across these measures.

---

6 Neural Network on Raw Signals

The methods used so far involved using either the signals as predictors or features obtained from them. However, we could also let a model learn the best features. Signals are a series of sequential points, which means we can naturally apply Recurrent NNs, which are by design able to process time series-like data. However a signal could also be seen as a 1D image, since points close together are very informative for each other. Thus, we can also look at Convolutional NNs, which can be seen as feature extractors.

To make use of both we'll create a 1D CNN-LSTM. First, the NN applies a one-dimensional convolutional layer to extract the feature maps, then we apply batch normalization to reduce covariate shift and add regularization, finally, downsamples the signal with a 1D max pooling. This block can be repeated a number of times to extract more specific patterns. The feature vectors are passed through an LSTM layer, then dropout to further regularize and provide probabilistic predictions. A fully connected head uses the information extracted by the CNN and RNN parts to make the final predictions, two neurons, one for force X and one for force Y. The input passed to the network are the signals, reshaped to use 50 steps (0.5 seconds) of 7 muscles to predict two force measures. To preserve the sequence assumed by the structure, the windows are created after grouping by repetition and direction.

The same test split and CV scheme as before is applied, and the models are trained with a learning rate scheduler and early stopping. We train one model per fold, each seeing only two repetitions, then a final one that sees all 3 but cannot use the callbacks since they are based on the validation data. The model's performance was very good too, with all 4 ranging from 0.97 to almost 0.99 R-squared.

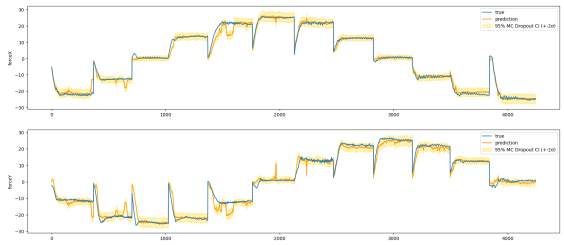

Figure 4: CNN-LSTM Predictions with 95% MC intervals over the 12 directions of repetition 4

The full model wasn't the best due to not having a validation set, thus being subjected to overfitting. Activating Dropout also during the inference phase allows us to get Monte Carlo intervals, which were quite narrow since the model was already good.

7 Explainability and Feature Importance

In each experiment, we were able to obtain a score to rank the feature. Each approach was different, but they were able to let us identify the most important contributors to the force. The linear models are intrinsically interpretable due to their coefficients directly multiplying the covariates, while adding an interaction feature makes it harder to understand, we could extract the average importance for each muscle. The deltoids were the highest contributors to X forces, with the tricep being a close third. This last muscle was also the most important one for force Y, and the overall winner after taking the mean between the two directions. Select K Best top 10 most important features were almost entirely tricep features, with a few brachial ones (to predict both forces, likely dominated by the Y). In the advanced models section, the SHAP plots and model embedded feature importance both made apparent that the most important muscles for force X were the deltoids, while for force Y, they were the triceps and brachialis. SHAP analysis provides also directionality of impact, showing not just which features were important, but how they influenced the predictions. For instance, higher `pDEL_mean` values consistently increased Force X outputs, while `lTRI_mean` was a strong positive driver for Force Y. SHAP also captured non-linear effects that traditional feature importance methods overlooked, offering a clearer understanding of how specific muscles contribute differently depending on the force axis. Even PCA's first component had lTRI as the muscle bringing the most variance. These agreeing results made us conclude that the most important features were DEL + PEC for force X and TRI + BRA for force Y. After analyzing the muscle activation, we noticed that the deltoids and triceps have less intense activations compared to the others. This doesn't go against the previous results as it may seem: the fact that other muscles are more active means that the least active ones are often the ones that make the difference.

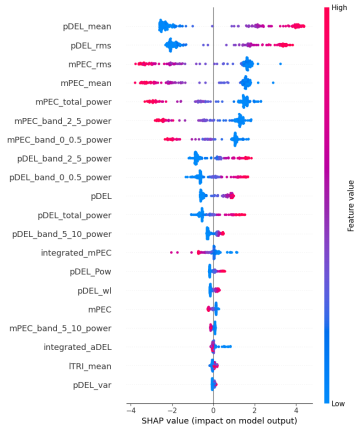

Figure 5: Shape Summary Plot for Force X

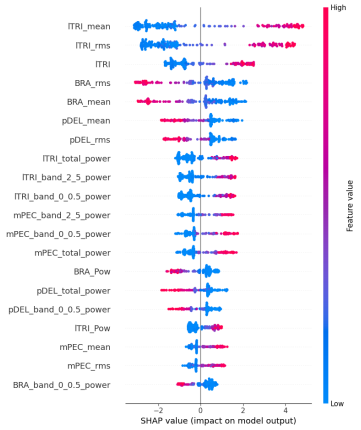

Figure 6: Shape Summary Plot for Force Y

During the signal preprocessing phase we wanted to find a way to deal with cross-talking artifacts, a common problem for EMG signals. In the paper of the experiment is reported that artifacts from the ECG may appear, and so with the recorded ECG, they were able to clean the signals. We don't have at our disposal the recordings of the ECG so we looked for other techniques. We found that Independent Component Analysis can be used in a high-density EMG scenario to retrieve the original signals that are strongly affected by cross-talk [Sta+07]. In our case, we don't have high-density signal recordings, but we wanted to experiment with this approach because of its simplicity and fast implementation. The obtained results were not significantly better than the ones without ICA; the comparison has been made with the "Raw Signal" models. With no benefits in using it we dropped it and we continued to develop other models without this preprocessing step.

8 Bibliography

References

[Sta+07] Didier Staudenmann et al. “Independent component analysis of high-density electromyography in muscle force estimation”. en. In: *IEEE Trans Biomed Eng* 54.4 (Apr. 2007), pp. 751–754.

[LS18] Nicola Lotti and Vittorio Sanguineti. “Estimation of Muscle Torques from EMG and Kinematics During Planar Arm Movements”. In: Aug. 2018, pp. 948–953. DOI: 10.1109/BIOROB.2018.8488131.