Predicting Finger Force Trajectories from sEMG Data

Ricardo Calderon
University of Chicago
May 27, 2025

Introduction and Motivation

Approximately 0.6% of the U.S. population will undergo an amputation during their lifetime. For these individuals, high-quality prosthetics are essential, not just for regaining mobility but for returning to everyday life. Current upper-limb prosthetics often rely on cycling through preset gestures, a clunky and unintuitive process. Supporting fine motor skills requires the ability to vary grip strength in real time, which necessitates force regression (FR).

Ultimately, this work aims to facilitate future research on FR models, thereby improving the functionality and accessibility of prosthetics for end users.

Background and Related Work

Previous research has focused primarily on gesture recognition (GR) due to its lower cost and simplicity. In GR, surface electromyography (sEMG) signals are classified into a discrete set of predefined actions. Those models are resistant to noise due to their reliance on *relative* sEMG strength, allowing electrode inputs to be normalized without impacting performance. Despite their success, GR models have two major drawbacks: the above mentioned limited menu of gestures and the inability to abort or modulate movements once they are underway.

Force regression is the opposite. It offers greater functionality to users but incurs a greater upfront cost. These models require *absolute* EMG values, limiting the allowable preprocessing. Unfortunately, raw electrode data is both large and noisy, so some form of cleaning is still necessary. Later in this paper, we will show a real-world example of improper filtering. Nevertheless, FR provides unparalleled fine control and can seamlessly interpret complex movements.

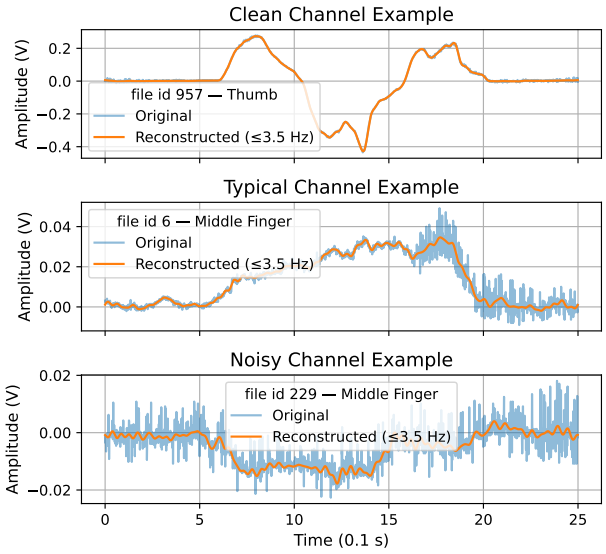

Figure 1. Examples of force signals compared to their low-frequency reconstructions.

Gijsberts et al. (2014, Frontiers in Neurorobotics) showed the practicality of Incremental Ridge Regressions by predicting finger forces from sEMG signals with a percent error of 16.7%. Chihi et al. (2023, Biosensors) used a multi-model process to predict finger forces with a NRMSE (normalized by range) of 2.6%. In this project, we have reached a NRMSE of 4.5% using Ridge Regressions and 5% using CNNs.

Both GR and FR approaches have high accuracy requirements, but FR has an additional precision requirement. Hence, beyond choosing and tuning a FR model, a lot of care has to go into collecting data and decoding signals, making generating training data exceedingly expensive. This project originally sought to compare the usefulness of different model types for FR; however, the discovery of a performance cap constraining the model's

prediction error to at least 4% of the target range prompted the need to examine sources of systemic error.

Dataset and Challenges

This study uses the PhysioNet Open Access Dataset and Toolbox of High-Density sEMG Recordings (Hyser), specifically its NDoF and random task sub-datasets. The data includes 256 channels arranged in 8x8 surface electrode arrays over four muscle groups, sampled at 2048 Hz. Synchronized finger force data is sampled at 100 Hz.

One major issue plaguing the Hyser dataset was the inconsistent signal clarity. Figure 1 displays a range of noise levels. The middle plot highlights a broader issue with the dataset: imperfect experimental setup and recording procedures. The sudden onset of higher frequencies at the 1.5s mark is typical of connections losing full contact.

Beyond signal quality issues, additional challenges arise from the modeling objectives themselves, which impose strict constraints on how the data can be structured and processed.

To simulate real-time prosthetic control, data was divided into randomly selected time "chunks". Shorter chunks reduce dimensionality but flatten force plots, making signal harder to distinguish from noise. At 0.1 seconds, the forces in Fig. 2 already show little variance.

Conversely, longer inputs become computationally impractical. The high dimensionality means choosing too long of a chunk leads to space issues; for example, attempting to model electrode interactions with polynomial terms would yield over 33 thousand terms per timestep for quadratic interactions or over 2.8 million terms per timestep for cubic interactions. A naive approach would be unmanageable for a desktop computer, let alone a prosthetic's controller. Dimensionality reduction is therefore warranted.

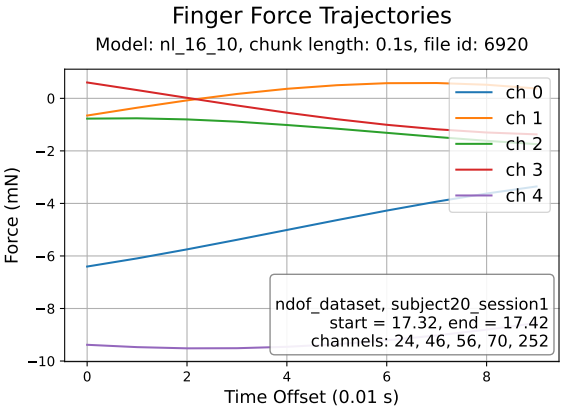

Figure 2. Finger force trajectories. Channel 0 is the thumb and channel 5 is the pinky finger.

Due to its massive size, the Hyser dataset also includes preprocessed signals. The final challenge was to recognize that the provided preprocessing was partly incorrect and partly unnecessary.

Dimensionality Reduction and Preprocessing

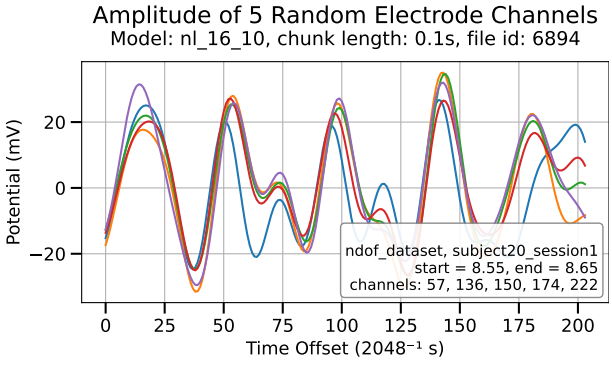

Figure 3. Voltage of five randomly selected electrode channels over a 0.1 second interval.

Because absolute sEMG values are needed, traditional normalization cannot be applied. Aggregating electrodes is a tempting solution but comes with drawbacks. Averaging inputs can dilute signals and introduce interference from neighboring finger movements. Summing inputs is vulnerable to destructive interference from misaligned electrodes. Taking the maximum within a group is highly susceptible to high-frequency noise—a recurring issue with this dataset.

Nevertheless, electrodes within the same array exhibit a high degree of collinearity. Figure 3 shows a typical example. Applying principal component analysis (PCA) to reduce redundancy is therefore justified.

Regarding the force data, the preprocessing focused on creating a smooth signal that would accommodate all reasonable motions. The raw blue signal in the third plot of Fig. 1 is not reasonable. A random sample of 500 force plots was generated and each plot manually labelled "noisy" or "clear". Power spectra of the clear channels were visually inspected to choose a cutoff frequency of 2.5 Hz. To err on the side of caution, this cutoff was increased to 3.5 Hz. That corresponds to a finger moving back and forth 3.5 times in a second. The reader is encouraged to test how quickly they can bend/extend their fingers. A corresponding low-pass filter was then applied to obtain a smoother signal, the orange curves in Fig. 1.

Model Architectures

Given the high dimensionality, ridge regression was used as a baseline due to its regularization capabilities. In addition, both 1D and 2D convolutional neural networks (CNNs) were tested in the hope that their architecture could naturally leverage the spatial structure of the electrode arrays without explicit interaction terms.

The first models had electrode chunk lengths of 0.44 seconds and force chunk lengths of 0.29 seconds. The 0.29 seconds correspond to the inverse of the 3.5Hz cutoff frequency. The electrode chunks had a lead time of 0.15 seconds to account for human reaction time. Later, chunk length would be varied to try breaching the performance cap, see Table 3 in Appendix II. Additionally, we varied the filters applied during preprocessing, the number of channels preserved after PCA, and for the CNNs, architectural parameters like number/type of convolutional, pooling, and dense layers, kernel size, etc. Since more than 60 models were tested, all showing similar performance, the dataset/model parameter combinations and performance metrics are included in Table 5, Appendix II.

Performance was very consistent across models of the same type and similar between model types. In general, ridge regression outperformed CNNs, although 1D CNNs slightly outperformed their 2D counterparts. The consistency across architectures suggests a performance ceiling imposed by the data's noise rather than model limitations.

The Performance Cap and Data Quality

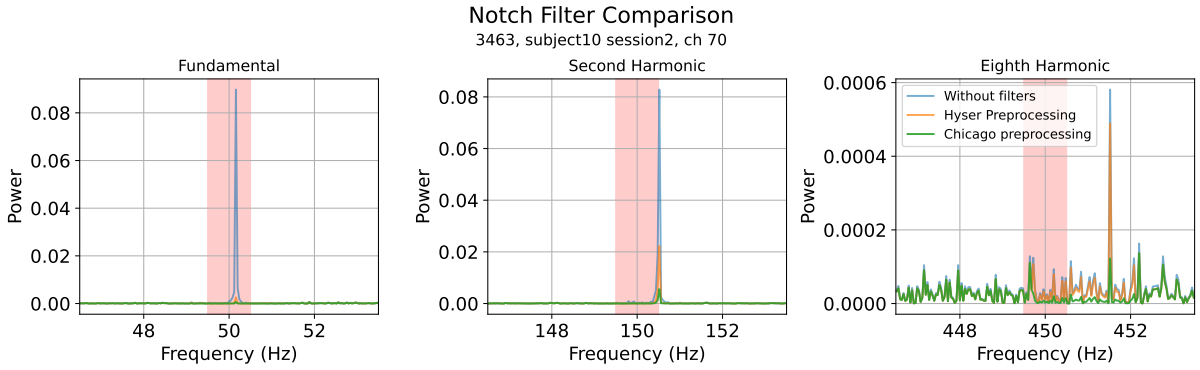

Figure 4. Power spectrum around the power grid frequency and some harmonics. The shaded area shows the approximate range of the Hyser notch filter. It barely catches the second harmonic and completely misses the eighth harmonic.

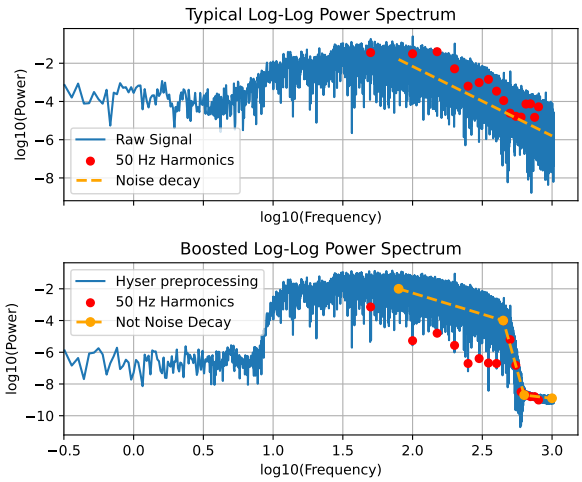

Figure 5. Comparison of log-log power spectra before and after Hyser preprocessing.

Despite tuning, the initial models could not surpass a NRMSE of 0.06 (normalized by output range). While this may be adequate for coarse tasks, it is insufficient for applications requiring fine motor control. This suggested that the bottleneck lies in the data quality.

Hyser's preprocessing pipeline included a notch filter at multiples of 50 Hz to remove power grid interference. However, the notch was too narrow and failed to account for frequency drift; e.g., a 50.1 Hz grid would render the filter ineffective. Figure 4 illustrates such a case, with sharp blue spikes representing grid noise several magnitudes stronger than the sEMG signal. After applying the wider notch filter centered at 50.1 Hz (see the green signal in Fig. 4), model performance improved to an average NRMSE of 0.053, with the best model returning a NRMSE of 0.045.

Additionally, the bandpass filter (10–500 Hz) used in the Hyser preprocessing appeared to amplify high-frequency noise. Raw log-log power spectra (Fig. 5) showed the expected $1/f^{\alpha}$ decay, but Hyser's preprocessed spectrum displayed a suspiciously elevated plateau between 1 and 2.5 in log-space, indicating a boost in higher frequencies that is hard to justify for our purposes.

Conclusion and Future Work

These results suggest that the real limitation is the quality and preprocessing of the dataset rather than the models' architecture. The Hyser dataset has an electrode sampling rate of 2048Hz. However, this study proves that such a high sample frequency is excessive because the low-pass filter of 100 Hz did not have a relevant effect on NRMSE. This is good because it lowers the hardware specs for prosthetics manufacturing. The infimum cutoff would theoretically be comparable to twice the inverse of contraction durations (Nyquist theorem). In practice however, an elbow plot could be use to find the cutoff frequency that balances model performance with efficient input collection. Incidentally, we have also shown that there is no value to boosting really high frequencies like what Hyser did with the 10-500Hz boost.

The good news does not stop with the sampling frequency optimization. Sixteen electrodes will work as well as 256. That was suggested early in the development process by an elbow plot but it is confirmed by the comparable NRMSE between models using 16, 25, 40, or 256 channel inputs.

Future work should focus on improving data quality: potentially through reprocessing raw sEMG signals with better filters. In addition, testing models on a per-subject or per-session basis may improve performance by reducing inter-subject variance. While the original goal of building a prosthetic "brain" was hindered by these limitations, this analysis provides useful insight into the challenges of working with sEMG data and informs development of more effective signal processing pipelines for prosthetic control systems.

Bibliography

Gijsberts A, Bohra R, Sierra González D, Werner A, Nowak M, Caputo B, Roa MA, Castellini C. Stable myoelectric control of a hand prosthesis using non-linear incremental learning. Front Neurorobot. 2014 Feb 25;8:8. doi: 10.3389/fnbot.2014.00008. PMID: 24616697; PMCID: PMC3935121.

Castellini C, Gruppioni E, Davalli A, Sandini G. Fine detection of grasp force and posture by amputees via surface electromyography. J Physiol Paris. 2009 Sep-Dec;103(3-5):255-62. doi: 10.1016/j.jphysparis.2009.08.008. Epub 2009 Aug 7. PMID: 19665563.

Nguyen, Phuc & Su, Shun-Feng & Kuo, Chung-Hsien. (2024). A Frequency-Based Attention Neural Network and Subject-Adaptive Transfer Learning for sEMG Hand Gesture Classification. IEEE Robotics and Automation Letters. PP. 1-8. 10.1109/LRA.2024.3433748.

Baldacchino T, Jacobs WR, Anderson SR, Worden K, Rowson J. Simultaneous Force Regression and Movement Classification of Fingers via Surface EMG within a Unified Bayesian Framework. Front Bioeng Biotechnol. 2018 Feb 26;6:13. doi: 10.3389/fbioe.2018.00013. PMID: 29536005; PMCID: PMC5834453.

Chihi I, Sidhom L, Kamavuako EN. Hammerstein–Wiener Multimodel Approach for Fast and Efficient Muscle Force Estimation from EMG Signals. Biosensors. 2022; 12(2):117. https://doi.org/10.3390/bios12020117

Appendix I: Work Beyond Machine Learning

The following are statements derived from Signal Processing. While that is not part of Machine Learning itself, it is part of the broader data analysis to be done in the construction of a model. They are included in this appendix for the sake of completeness.

Here is a conjecture that was not addressed due to time constraints. We believe that the correct input signal for a model predicting forces is a rolling integral of the square of the electrode signal.

Before we begin, note the following facts. First, there exists a baseline electric field around living things. Its exact magnitude or how it varies across subjects is irrelevant at the moment. Second, the power (energy per time) of an electromagnetic field in Ohmic systems is directly proportional to the potential squared and inversely proportional to the resistance of the medium. Third, the energy in a chemical reaction is proportional to the energy in the resultant electric field.

A muscle contraction's "strength" is not necessarily proportional to the amplitudes measured by sEMG arrays, and even if the voltages measured by the sEMG were an accumulation of the action potentials, what matters in chemical reactions is the amount of energy exchanged. To measure the *power* of a contraction, i.e. the amount of energy released per unit time from muscle fibers contracting, we can use the square of the amplitude measured by the electrode. (Assuming resistance is locally constant, it can be ignored for now.) Hence, the integral over a period of time of the sEMG amplitude squared gives a quantity proportional to the energy deposited in that period of time plus a constant. The origins of that constant have to do with the resting voltage not being zero but a negative value. Presumably, the negative resting voltage is a product of the subject's baseline electric field. The overall constant can easily be discarded by any model. In the case of the integral, it will appear as a linear contribution to the energy.

Now to reconcile the constant of proportionality (including resistance) there are many considerations. For example, the amplitudes measured by sEMG arrays are proportional to the resistance of the skin which varies from subject to subject and from day to day. Luckily, we do not have to concern ourselves with finding this value because it can be easily calibrated by asking the prothesis users to carry a small dynamometer that they can press every now and then. More expensive models could include a built-in dynamometer.

Appendix II

The models in this appendix represent a selected subset of those generated during this study. The preprocessing pipeline was varied along three main axes: (1) the method of preprocessing applied to the raw electrode signals, (2) the number of electrode dimensions retained after principal component analysis, and (3) the chunk length, defined as the number of time points per channel.

The tables that follow summarize the evaluated configurations. Due to time constraints, CNN 2D models were only trained for one configuration: 'hs_16_29' (terminology defined below).

Table 1. Summary of preprocessing configurations.

ID	Preprocessing Description
ne	• Notch filter at 50.1Hz and harmonics • High frequency signal components intact • Sampling frequency left intact at 2048Hz • 12600 chunks for training, 4200 chunks for testing
nl	• Notch filter at 50.1Hz and harmonics • Low-pass filter at 100Hz • Sampling frequency left intact at 2048Hz • 12600 chunks for training, 4200 chunks for testing
oe	• Notch filter at 50.1Hz and harmonics • High frequency signal components intact • Sampling frequency left intact at 2048Hz • 25200 chunks for training, 8400 chunks for testing
hs	• Hyser notch filter at 50.0Hz and harmonics • High frequency signal boosted from 10Hz to 500Hz • Sampling frequency left intact at 2048Hz • 12600 chunks for training, 4200 chunks for testing

Table 2. Number of electrode dimensions after PCA (important electrodes)

PCA Dim	Description
16	16 electrodes after PCA
25	25 electrodes after PCA
40	40 electrodes after PCA

Table 3. Number of points per channel (chunk length)

Chunk Length	Description
10	Both electrode and finger signals are 0.10 seconds long
25	Both electrode and finger signals are 0.25 seconds long
40	Both electrode and finger signals are 0.40 seconds long
29	Electrode signals are 0.44 seconds and finger signals are 0.25 seconds long

Table 4. Summary of CNN architectures used in the study.
Tensor notation: B is the batch size, T' is the number of time steps per finger, and 5 represents the number of fingers. For 2D models, output is shaped as $[B, 1, 5, 29]$, where 1 is a singleton channel dimension.

Model	Type	Encoder	Extra Path	Output (Decoder)
CNN1d_enc_dec	1D CNN	3 conv layers (stride=2)	None	$[B, 5, T']$ via AdaptiveAvgPool1d
CNN1d_enc_dil_dec	1D CNN	3 conv layers (stride=2)	Dilated conv (dilation=50)	$[B, 5, T']$ via AdaptiveAvgPool1d
CNN2DEncoderDecoder	2D CNN	3 conv2d layers	None	$[B, 1, 5, 29]$ via AdaptiveAvgPool2d
CNN2_4conv_1max	2D CNN	4 conv2d layers	None	$[B, 1, 5, 29]$ via AdaptiveMaxPool2d
CNN2_4conv_1den	2D CNN	4 conv2d layers	None	$[B, 1, 5, 29]$ via Fully Connected layer
CNN2_6conv_no_decode	2D CNN	6 conv2d layers	None	$[B, 1, 5, 29]$ (encoder-only)

Table 5. NRMSE for various preprocessing and model combinations.

Preprocess	Model	Training NRMSE	Testing NRMSE
ne_16_10	Ridge CV before PCA	0.04499	0.04769
	Ridge CV after PCA	0.04504	0.04771
	CNN 1d enc dec	0.04527	0.04784
	CNN 1d enc dil dec	0.04527	0.04781
ne_16_25	Ridge CV before PCA	0.05199	0.04547
	Ridge CV after PCA	0.05211	0.04549
	CNN 1d enc dec	0.05259	0.04557
	CNN 1d enc dil dec	0.05259	0.04555
ne_16_40	Ridge CV before PCA	0.05290	0.05331
	Ridge CV after PCA	0.05312	0.05334
	CNN 1d enc dec	0.05374	0.05406
	CNN 1d enc dil dec	0.05374	0.05406
ne_25_10	Ridge CV before PCA	0.04731	0.05106
	Ridge CV after PCA	0.04734	0.05107
	CNN 1d enc dec	0.04758	0.05278
	CNN 1d enc dil dec	0.04758	0.05278
ne_25_25	Ridge CV before PCA	0.04975	0.05936
	Ridge CV after PCA	0.04984	0.05942
	CNN 1d enc dec	0.05033	0.05952
	CNN 1d enc dil dec	0.05033	0.05952
ne_25_40	Ridge CV before PCA	0.05506	0.05520
	Ridge CV after PCA	0.05520	0.05522
	CNN 1d enc dec	0.05593	0.05531
	CNN 1d enc dil dec	0.05593	0.05531

(Continued on next page)

(Continued from previous page)

Preprocess	Model	Training NRMSE	Testing NRMSE
nl_16_10	Ridge CV before PCA	0.04502	0.04769
	Ridge CV after PCA	0.04506	0.04772
	CNN 1d enc dec	0.04527	0.04781
	CNN 1d enc dil dec	0.04527	0.04781
nl_16_25	Ridge CV before PCA	0.05206	0.04547
	Ridge CV after PCA	0.05218	0.04549
	CNN 1d enc dec	0.05259	0.04557
	CNN 1d enc dil dec	0.05259	0.04555
nl_16_40	Ridge CV before PCA	0.05303	0.05334
	Ridge CV after PCA	0.05323	0.05335
	CNN 1d enc dec	0.05374	0.05340
	CNN 1d enc dil dec	0.05374	0.05340
nl_25_10	Ridge CV before PCA	0.04736	0.05176
	Ridge CV after PCA	0.04736	0.05176
	CNN 1d enc dec	0.04758	0.05278
	CNN 1d enc dil dec	0.04758	0.05278
nl_25_25	Ridge CV before PCA	0.04983	0.05936
	Ridge CV after PCA	0.04991	0.05942
	CNN 1d enc dec	0.05033	0.05952
	CNN 1d enc dil dec	0.05033	0.05952
nl_25_40	Ridge CV before PCA	0.05519	0.05521
	Ridge CV after PCA	0.05523	0.05522
	CNN 1d enc dec	0.05593	0.05531
	CNN 1d enc dil dec	0.05593	0.05531
oe_16_10	Ridge CV before PCA	0.04761	0.05557
	Ridge CV after PCA	0.04765	0.05558
	CNN 1d enc dec	0.04785	0.05571
	CNN 1d enc dil dec	0.04785	0.05571
hs_16_29	Ridge CV before PCA	0.06767	0.06910
	Ridge CV after PCA	0.06787	069100
	CNN 1d enc dec	0.06788	0.06847
	CNN2DEncoderDecoder	0.06883	0.06861
	CNN2_4conv_1max	0.06883	0.06862
	CNN2_6conv_no_decode	0.06014	0.07232

Appendix III

These are the statistics for a subset of the models considered. Although CNN 2D models are not included in Apendix I, they returned NRMSE values compatible with the following table.

Table 6. Summary statistics for training and testing NRMSE.

	Training NRMSE	**Testing NRMSE**
Mean	0.050440	0.053294
Standard Deviation	0.003536	0.004868
Minimum	0.044990	0.045470
25% Quantile	0.047580	0.047810
50% Quantile	0.050330	0.055215
75% Quantile	0.053148	0.057170
Maximum	0.055930	0.059520