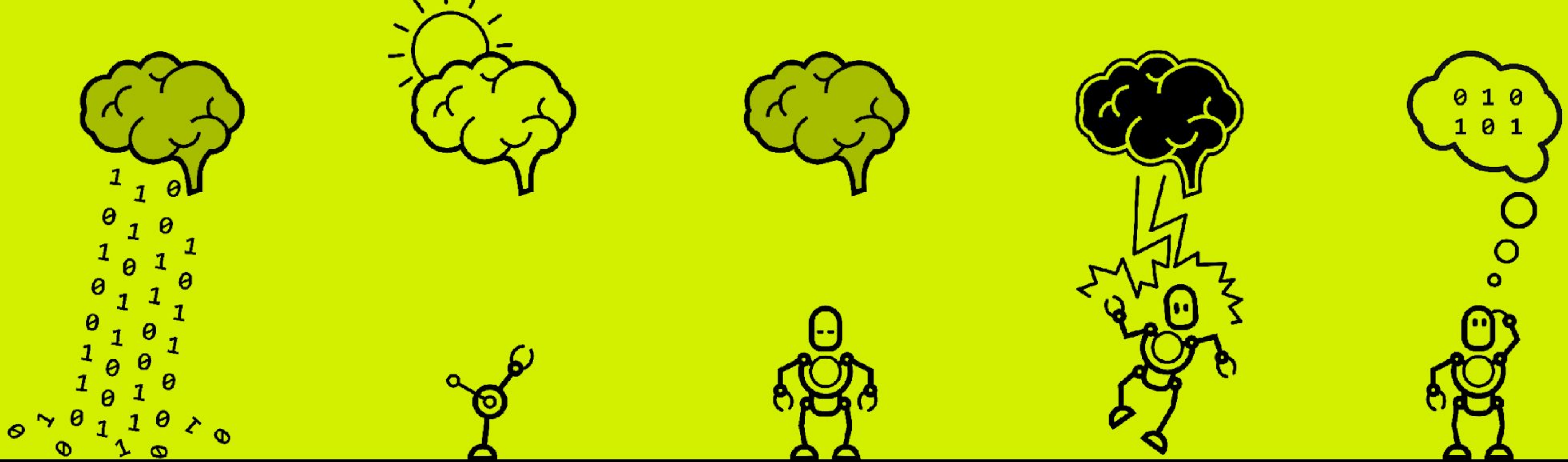


BR41N.IO

THE BRAIN-COMPUTER INTERFACE
DESIGNERS HACKATHON

ECoG Hand Pose Data Analysis

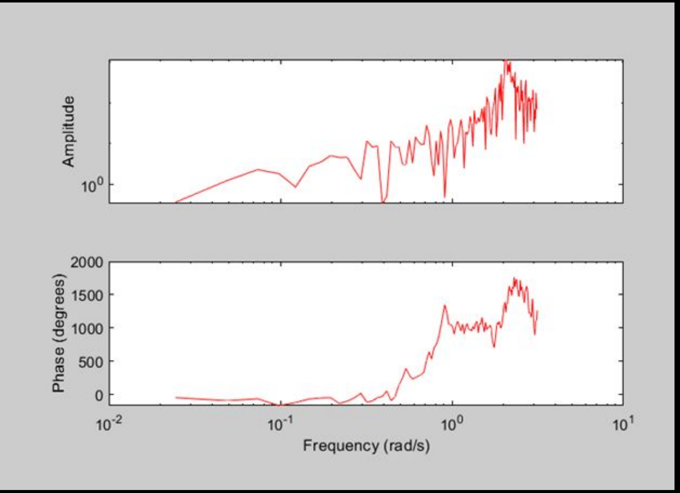
Amardeep, Bruno, Deepa, Lucas, Salina

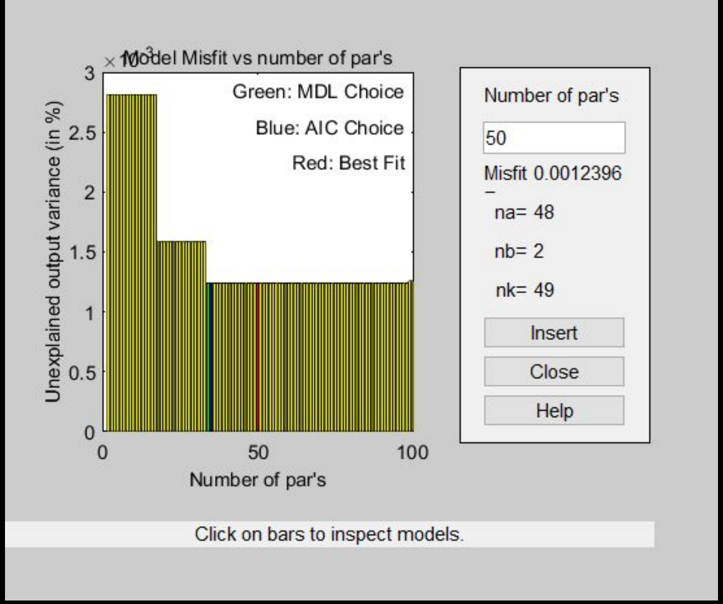
What We Did

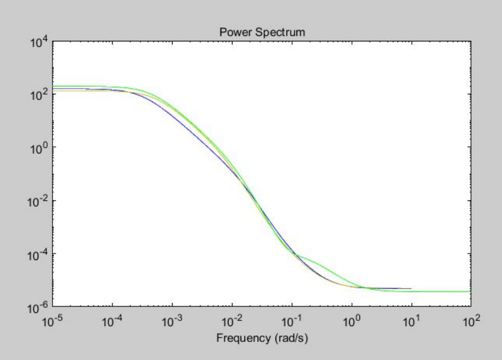
- Preprocessing
  - Detrending
  - Notch Filter 50Hz and Harmonics
  - High-Gama filtering (70-140 Hz Bandpass)
  - Separation on Class Epochs (from -0.2s~2s)
  - Downsampling (200Hz)
  - Hilbert Enveloping
- Various Classifications Methodologies!

What We Did

- ARMAX,ARMIA, FIR, Box Jenkins and SSM
- Time series modelling and forecasting experiments using system identification toolbox.
- Polynomial model tests and validation using error analysis and noise reduction.

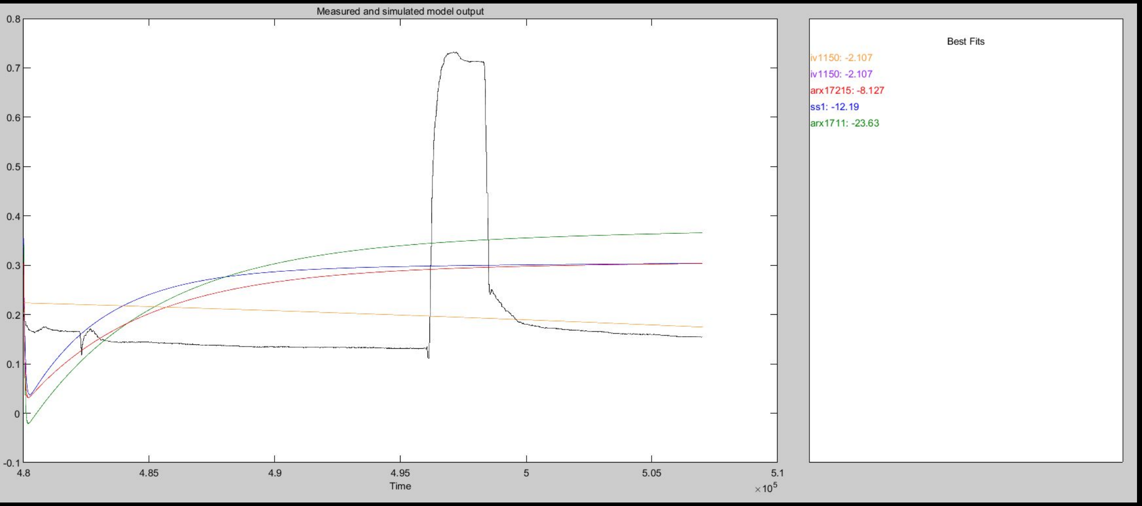






TIME SERIES MODELLING- MATLAB

- Data for hand pose is cleaned and modelled against state space models and polynomial models.
- ARMAX, ARX, FIR and Box Jenkins algorithms were applied on the dataset after removing means and trends.
- System identification toolbox allows for addition of order selection and parameter experiments.



Status: Estimated using N4SID with prediction focus

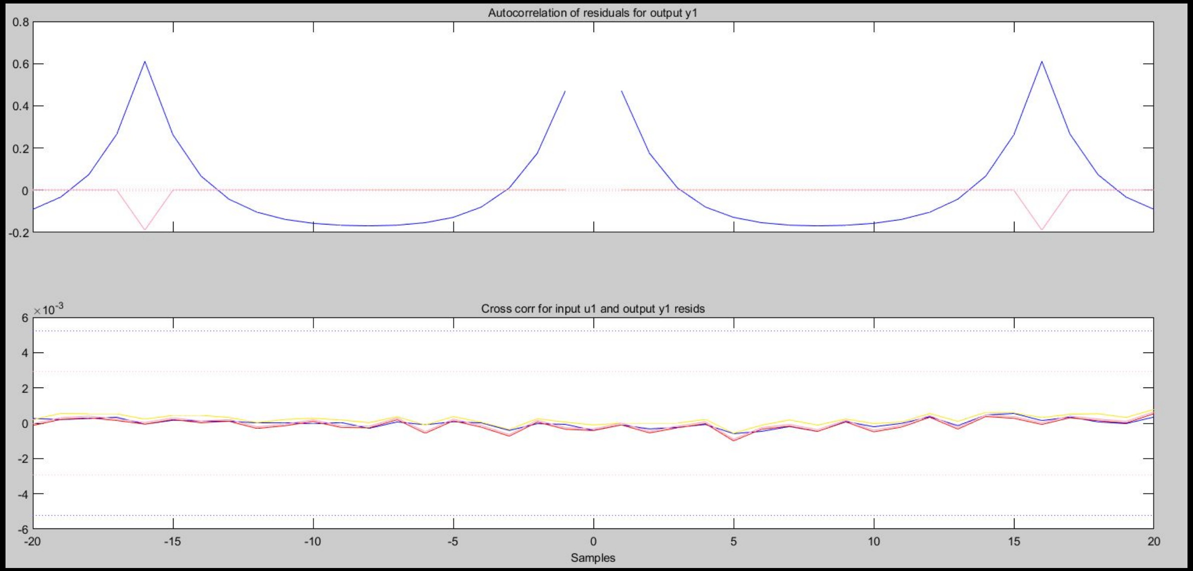
Fit to estimation data: **100%**, FPE: 2.24356e-21

Termination condition: Maximum number of iterations reached.

Number of iterations: 20, Number of function evaluations: 96

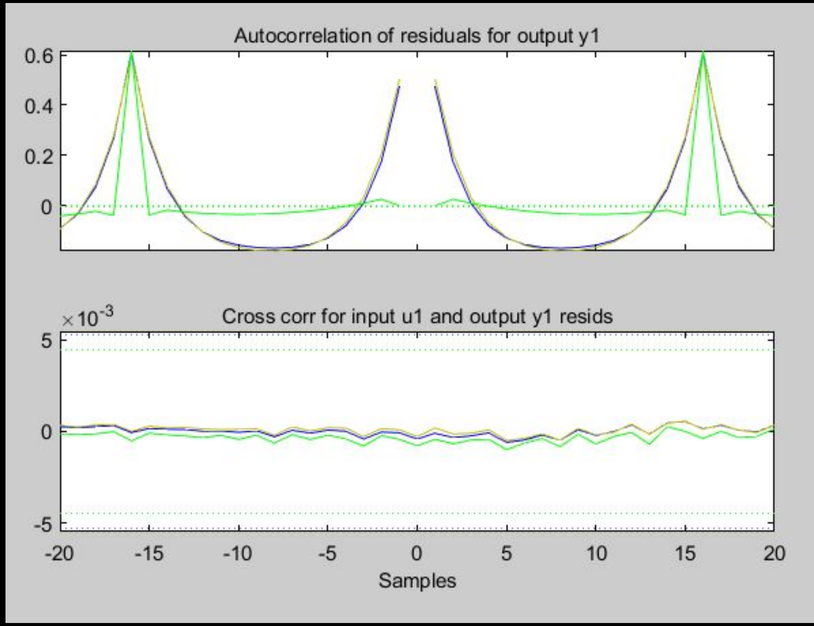
Status: Estimated using PEM with prediction focus

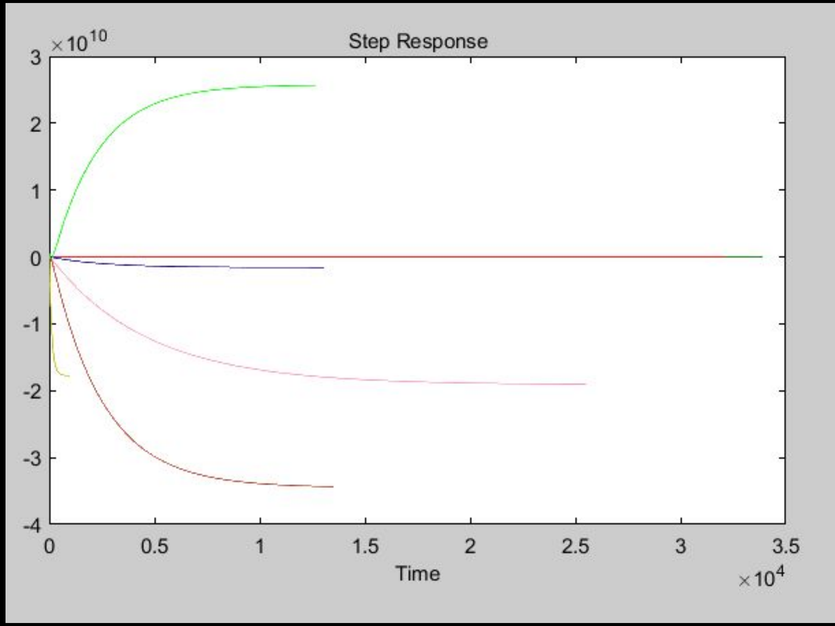
Fit to estimation data: **99.19%**, FPE: 2.96609e-06



Status: Estimated using N4SID with prediction focus

Fit to estimation data: **98.83%**, FPE: 6.56351e-06

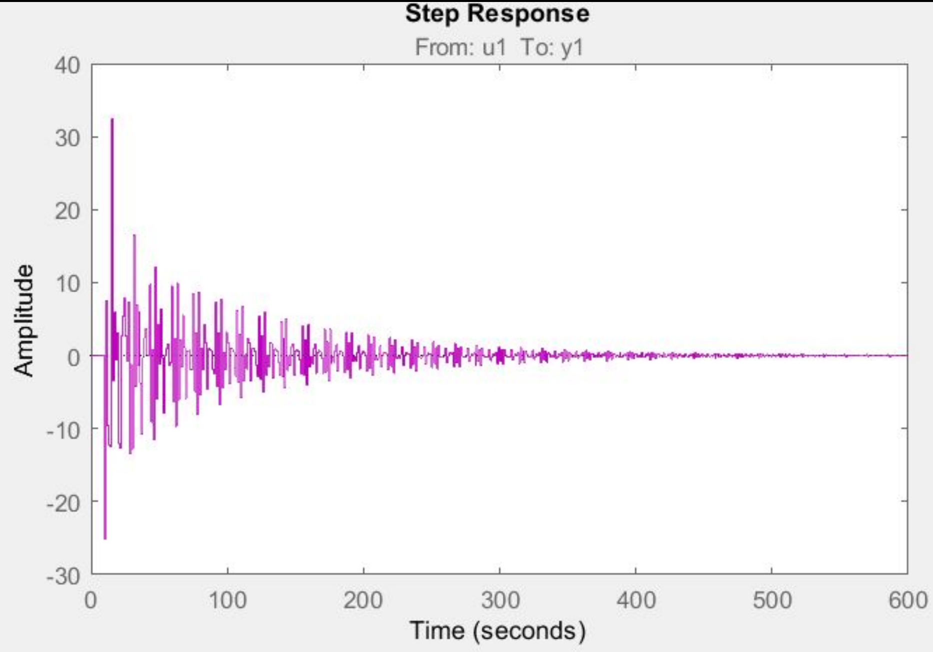


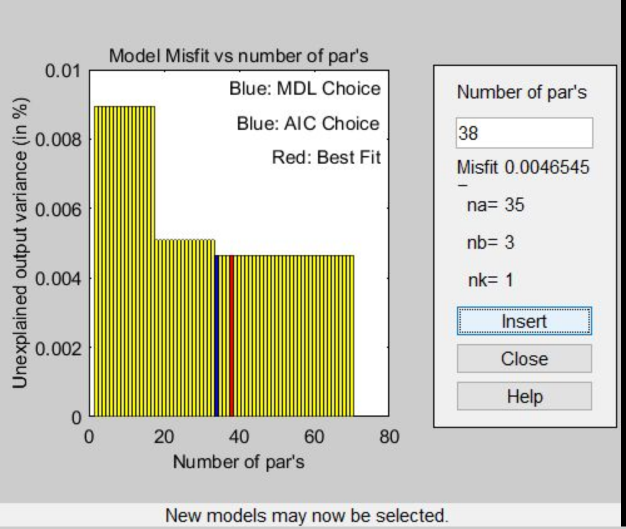


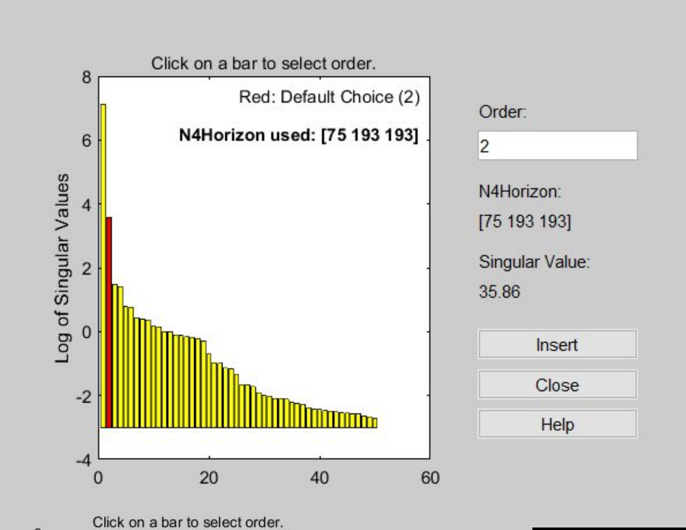

```
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)

  A(z) = 1 - z^-1 - 2.68e-08 z^-2 - 2.037e-08 z^-3 + 4.979e-08 z^-4 - 3.265e-08 z^-5
          - 1.266e-08 z^-6 + 2.085e-08 z^-7 - 2.936e-08 z^-8 + 4.379e-08 z^-9 - 3.041e-08 z^
          -10 + 9.161e-09 z^-11 - 6.709e-09 z^-12 + 5.334e-09 z^-13 - 7.097e-09 z^-14
                                   + 1.079e-09 z^-15 - 0.6541 z^-16 + 0.6541 z^-17
```

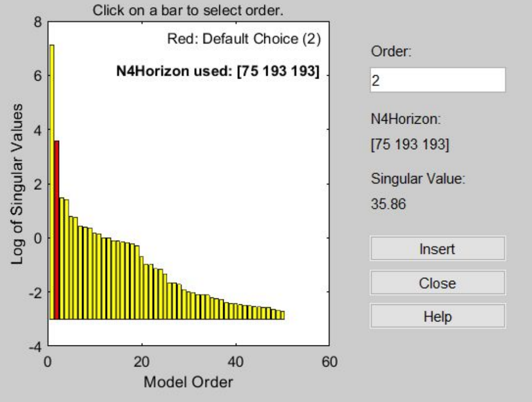
```
Continuous-time identified state-space model:
    dx/dt = A x(t) + B u(t) + K e(t)
     y(t) = C x(t) + D u(t) + e(t)
A =
             x1          x2
 x1  -1.526e-05   -0.002405
```

```
Discrete-time ARIMAX model:  A(z)y(t) = B(z)u(t) + [C(z)/(1-z^-1)]e(t)

  A(z) = 1 + 2.079 z^-1 + 2.415 z^-2 + 2.679 z^-3 + 2.371 z^-4 + 2.066 z^-5 + 2.395 z^-6
                                        + 2.19 z^-7 + 1.526 z^-8 + 0.7685 z^-9 + 0.08559 z^-10


  B(z) = -25.16 z^-10 - 19.58 z^-11 - 9.849 z^-12 - 26.77 z^-13 - 19.3 z^-14 + 17.53 z^-15
                                       + 16.46 z^-16 + 24.31 z^-17 + 32.99 z^-18 + 9.372 z^-19


  C(z) = 1 + 2.169 z^-1 + 2.587 z^-2 + 2.927 z^-3 + 2.595 z^-4 + 2.165 z^-5 + 2.602 z^-6
                                         + 2.575 z^-7 + 1.989 z^-8 + 1.323 z^-9 + 0.4207 z^-10
```









Status: Estimated using N4SID with prediction focus
Fit to estimation data: 98.91%, FPE: 5.0093e-06

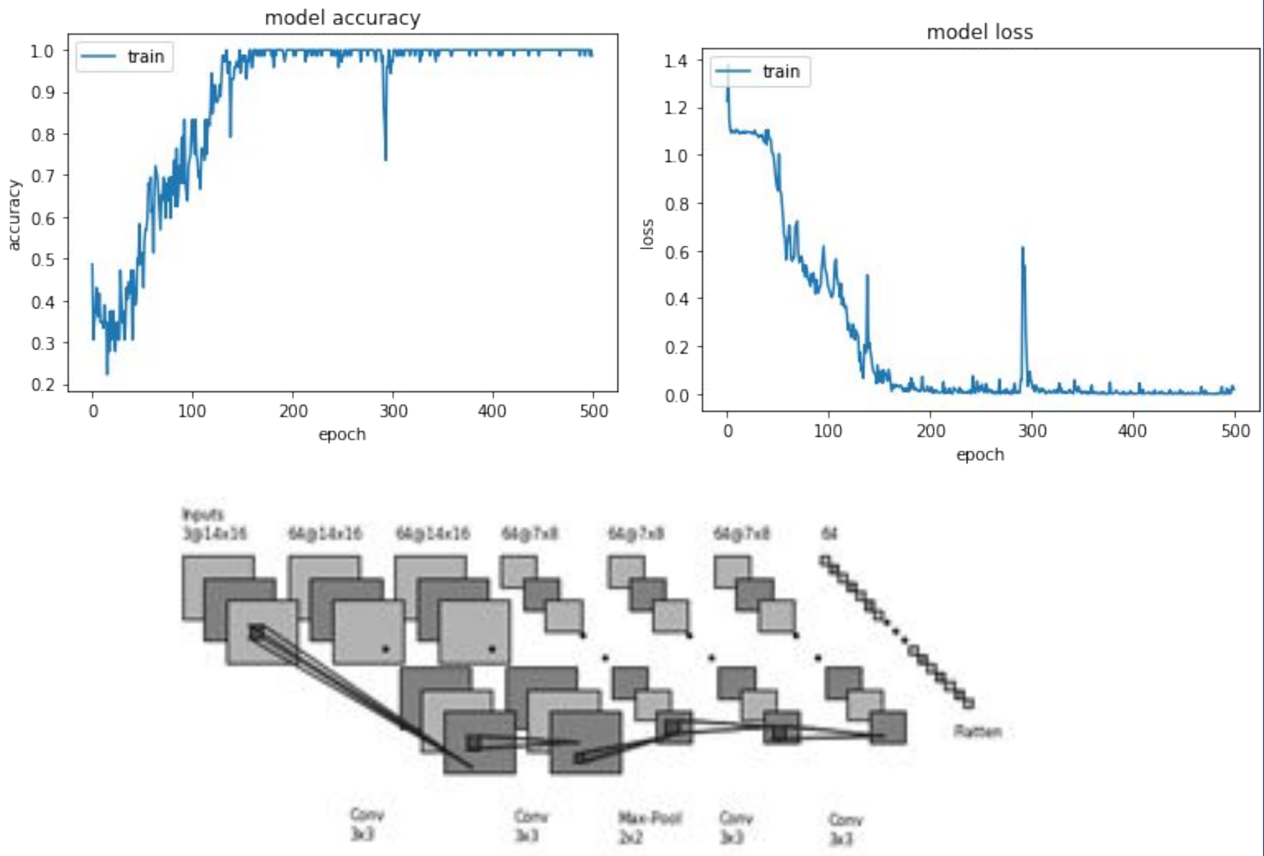
Iteration	Cost	Norm of step	First-order optimality	Improvement (%) Expected	Achieved	Bisections
0	4.69629e-06	-	3.62e+07	20.8	-	-
1	4.17459e-06	3.14	1.42e+07	20.8	11.1	2
2	4.14543e-06	0.591	1.2e+07	11.7	0.699	3
3	4.11706e-06	0.417	1.04e+07	11.4	0.684	3
4	3.96814e-06	0.632	7.53e+06	11	3.62	2
5	3.79486e-06	0.757	1.08e+07	7.81	4.37	1
6	3.73595e-06	0.533	1.87e+07	3.1	1.55	0
7	3.71418e-06	0.347	2.78e+07	1.35	0.583	0
8	3.70704e-06	0.331	8.38e+06	0.948	0.192	0
9	3.69764e-06	0.17	1.58e+07	0.785	0.254	1
10	3.68309e-06	0.235	9.66e+06	0.511	0.393	0
11	3.68238e-06	0.105	7.38e+06	0.194	0.0193	2
12	3.67828e-06	0.0767	1.7e+06	0.138	0.111	0
13	3.67764e-06	0.0999	1.59e+06	0.0323	0.0176	0
14	3.67747e-06	0.0558	1.16e+06	0.0195	0.0046	1
15	3.67732e-06	0.00148	1.83e+05	0.0086	0.00412	0

Termination condition: Near (local) minimum, (norm(g) < tol).
Number of iterations: 15, Number of function evaluations: 46

Status: Estimated using SSEST with prediction focus
Fit to estimation data: **99.07%**, FPE: 3.67745e- 7

What We Did

- 2D-Convolutional Neural Network
  - Using the EEG Signal as a (pseudo)image matrix (61,110,3)
  - Inspired on VGGNet model With only 3x3 convolutions and 2x2 Maxpolings
  - 10-Fold CV shows and average accuracy of 79%



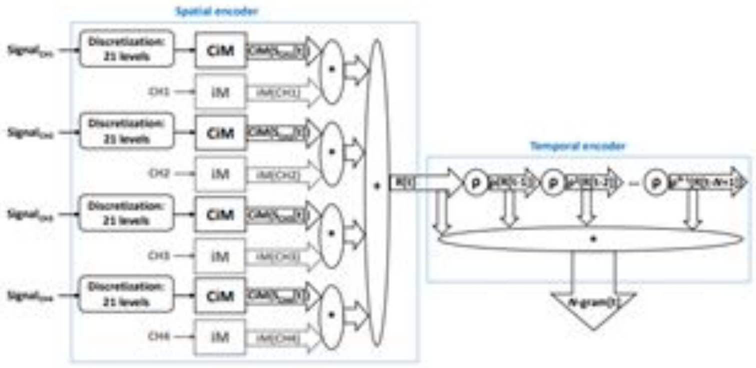
Training accuracy and loss of and early test model and architecture of the model

R. Laezza, “Deep neural networks for myoelectric pattern recognition - An implementation for multifunctional control,” 2018.

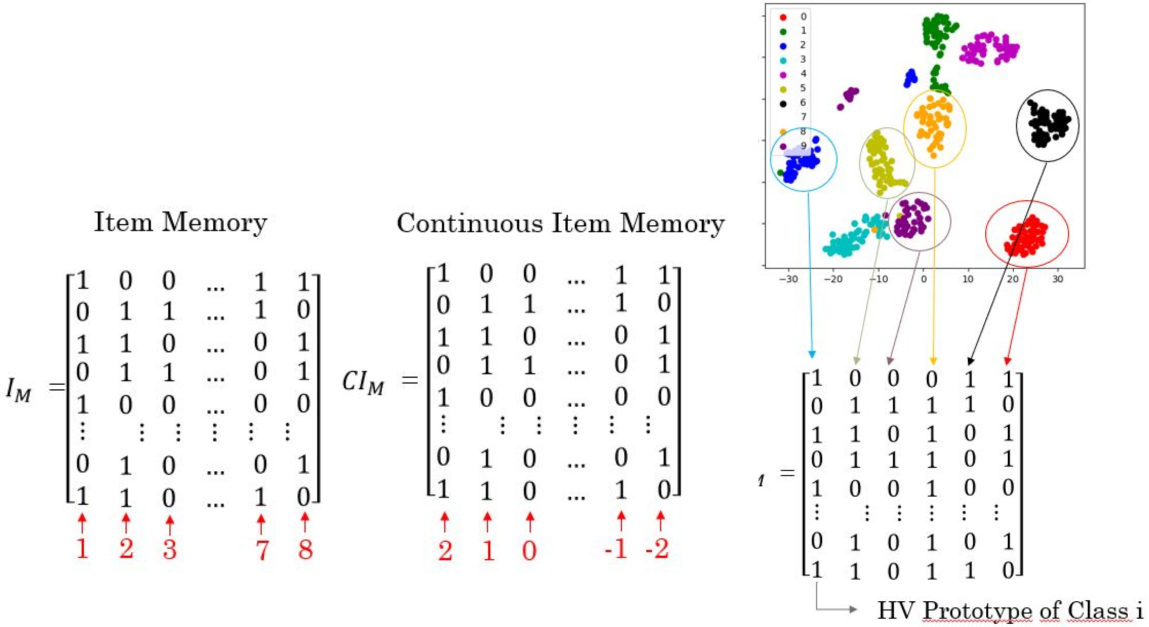
What We Did

- Hyperdimensional Computing
  - Initially proposed by Kanerva, for NPL processing. Some studies on biosignals
  - Brain inspired using high dimension (<10.000) (pseudo)random i.i.d binary vectors.
  - 10-Fold CV shows and average accuracy of 65%

A. Rahimi et al. “Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition,” in 2016
P. Kanerva, “Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors,”



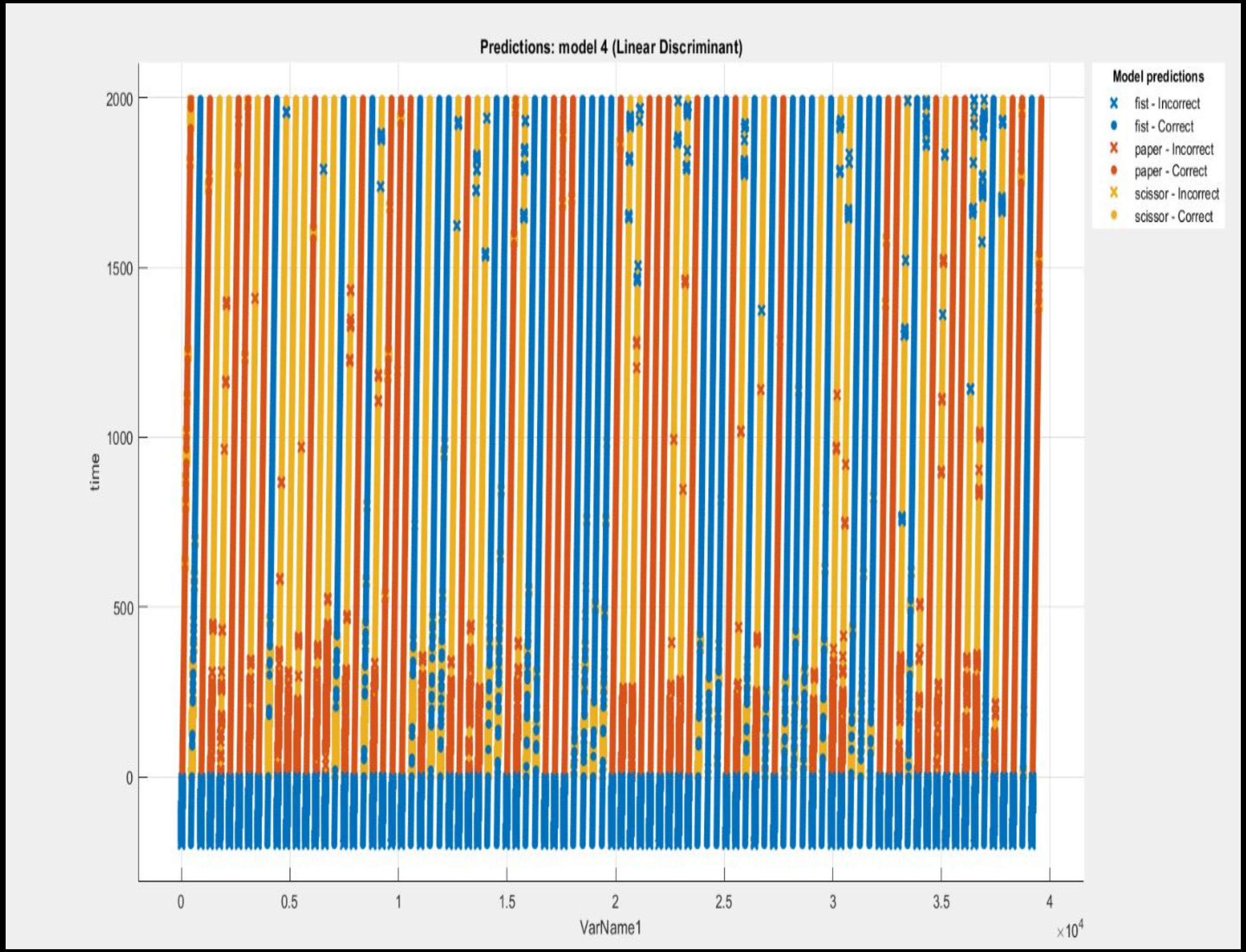
Pipeline of the spatial and temporal encoding process (adapted from reference)



Representations of the Memories used in the classification algorithm

Principal Component and Linear Discriminant Analysis

- First Band pass filter is applied
- then 10 cross validation partitioning the data set into folds and estimating accuracy on each fold.
- Principal Component Analysis (PCA) is first applied to the data set .
- The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the dataset.
- PCA weights are computing based only on training data .
- LDA is then applied to find the most discriminative directions.
- 10-Fold CV shows and average accuracy of 86.1%



LDA

- Methodology

$$\mathbf{y} = U^T \mathbf{x}$$

projection matrix

  - LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:

$$\max \frac{|U^T S_b U|}{|U^T S_w U|} = \max \frac{|\tilde{S}_b|}{|\tilde{S}_w|}$$

products of eigenvalues !

$\tilde{S}_b, \tilde{S}_w$: scatter matrices of the projected data $\mathbf{y}$

What We Would Do

- Better Preprocessing!
  - More Frequencies
  - Channel selection
- More emphasis on length of the classification window
- Rigorous comparison between models

Thank You !