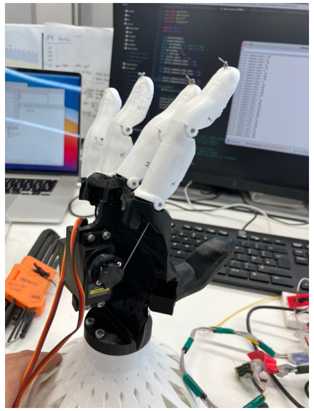
WORKSHOP 6
Human-Machine Interfacing and Mechatronics for Myoelectric Prosthetics

Goal: Explore from designing a bionic limb, assembling it, and implement a control system to control using sEMG

1st Day
Build your bionic limb

Fusion 360



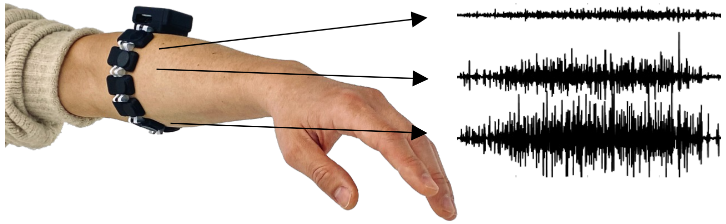
Lecture content (~20 min):

- Generics on Bionic Limb reconstruction

Task (~1h 30 min):

- CAD (get an idea on how to design a bionic limb)
- Assemble simplified limb
- Actuate the hand by a code command

2nd Day
Muscle interfacing technics



Lecture content (~30 min):

- Fundamental signal for motor control
- Interfacing with the Nervous system
- Feature extraction
- Data driven prosthesis control

Task (~1h 30 min):

- Implement an algorithm to create a 2DOF regressor

3rd Day
Integration into online bionic limb control

Task1 (~45min):

- Online interfacing by using a neural interface device
- Exploration on different gestures and labels

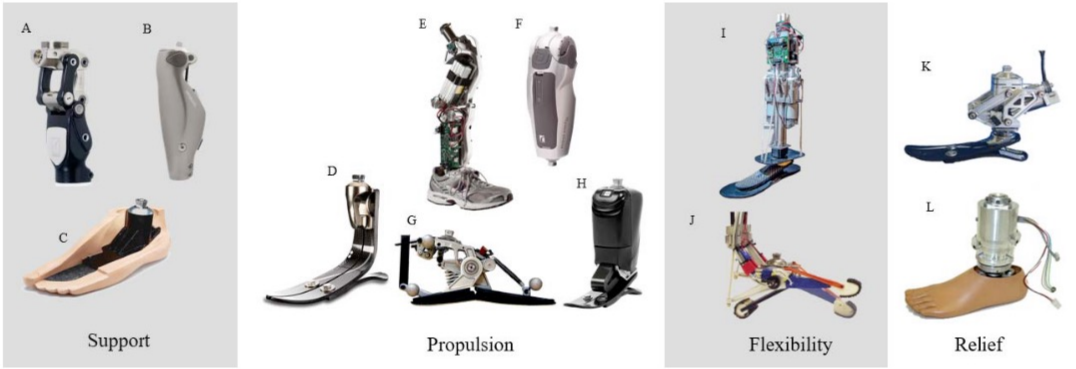
Task2 (~45min):

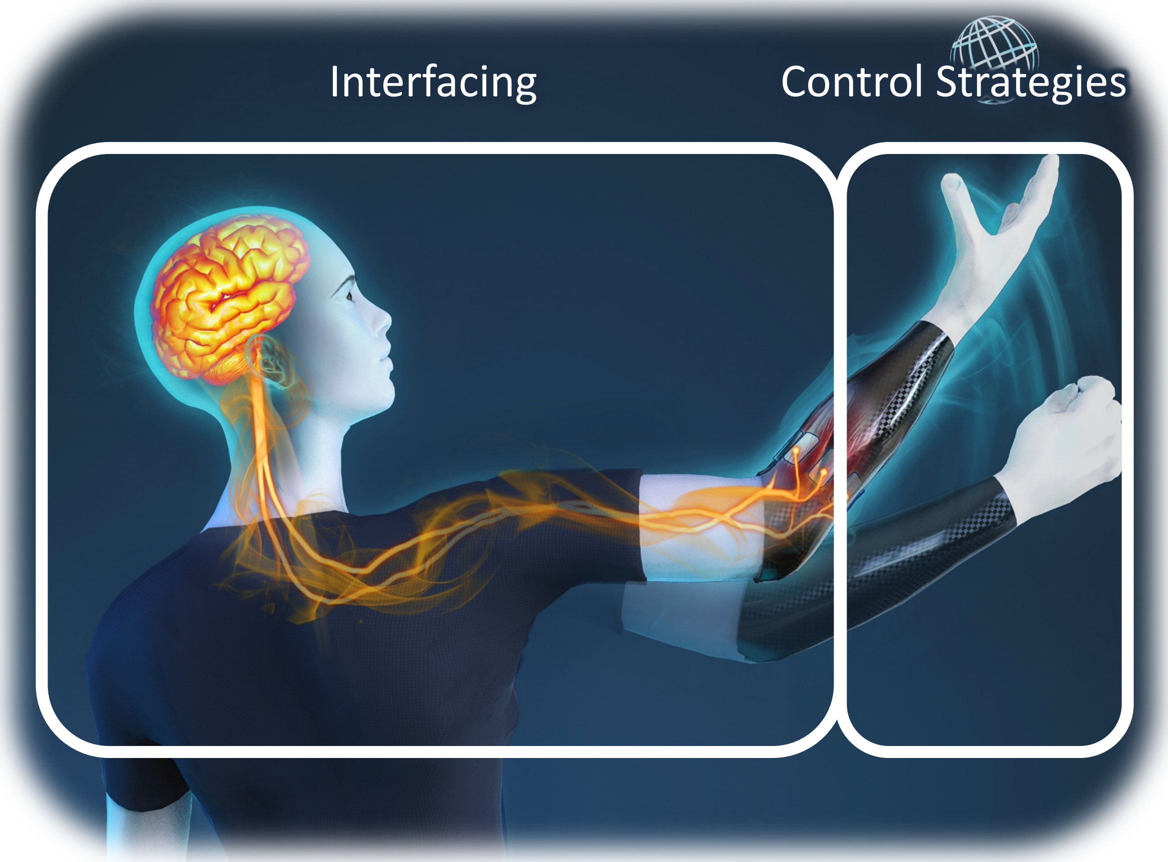
- Connect to a bionic hand and control by using your own muscle signals

Wrap up and Q&A (30min)

Controlling prosthesis by BCI

❖ Interfacing and control are the remaining challenges





Mechanics on prosthesis are well matured, but the interfacing to human's neural system and the control strategies are yet to meet the user fulfillment

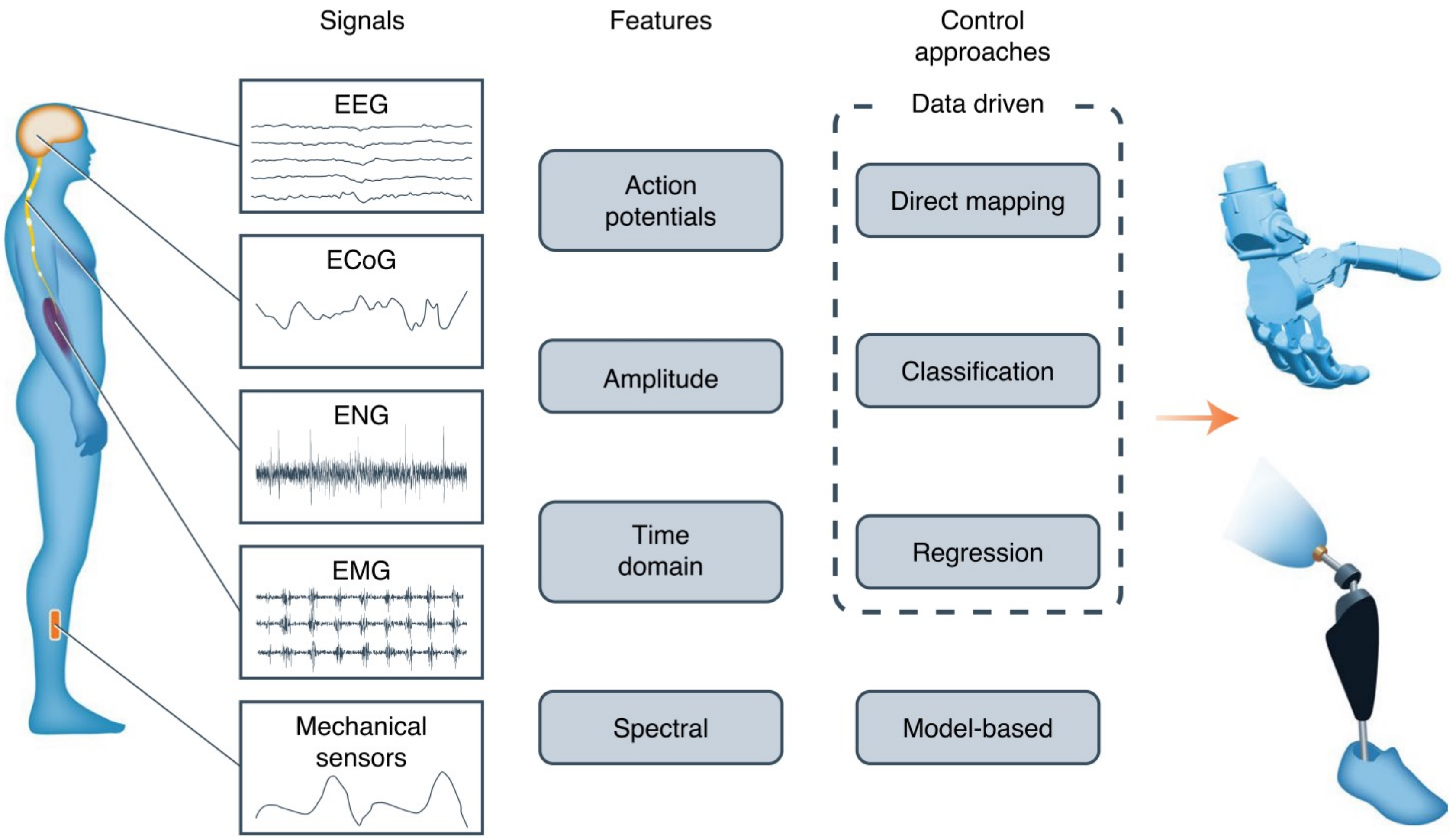
Roche et al., Current Surgery Reports, 2014

Price et al., *IEEE Trans. on Neu. Sys. and Reha. Eng.* 2019

Interfacing

Controlling prosthesis by BCI

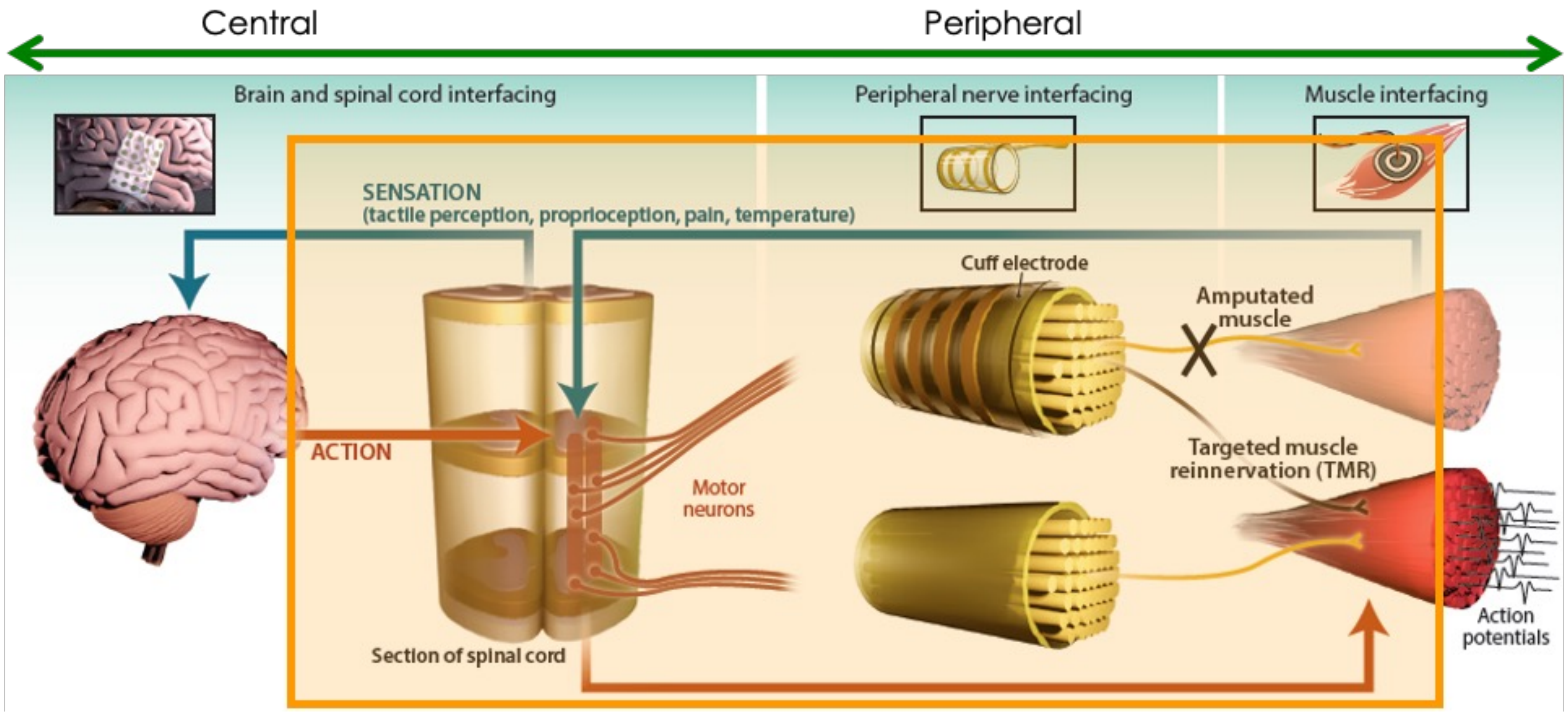
❖ Multiple ways to interface biological signals and control assistive devices



Farina et al., *Nature Biomed. Eng.* 2021

Muscle is a biological amplifier

❖ Muscles reflects the neural activities

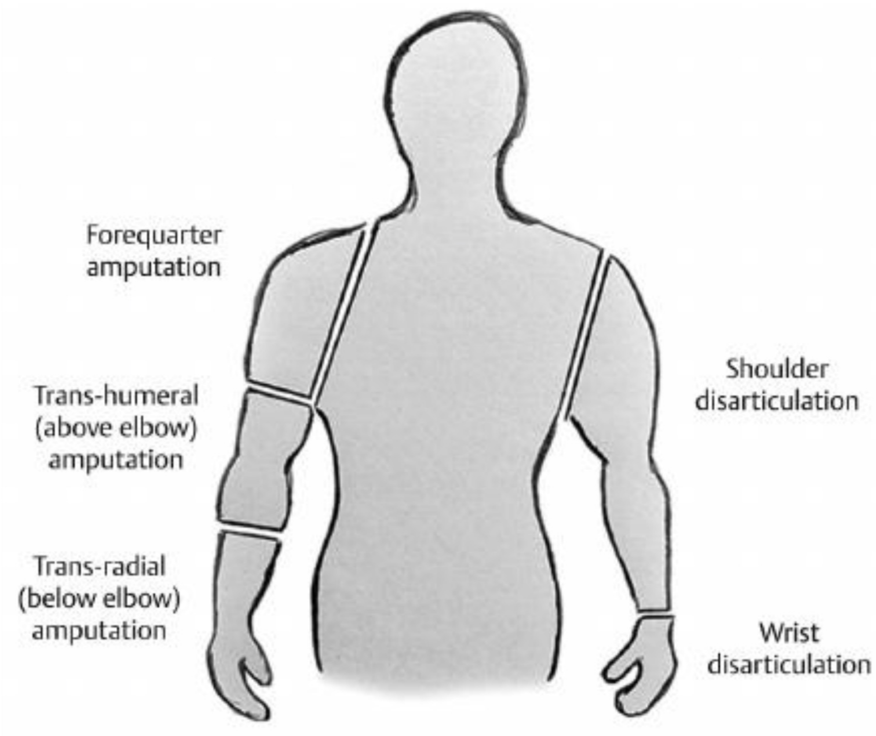


Muscle behaves as “bioscreen” of the neural activity

Farina & Aszmann, Science Translational Medicine, 2014

Ways to interface

❖ Multiple ways to interface muscle



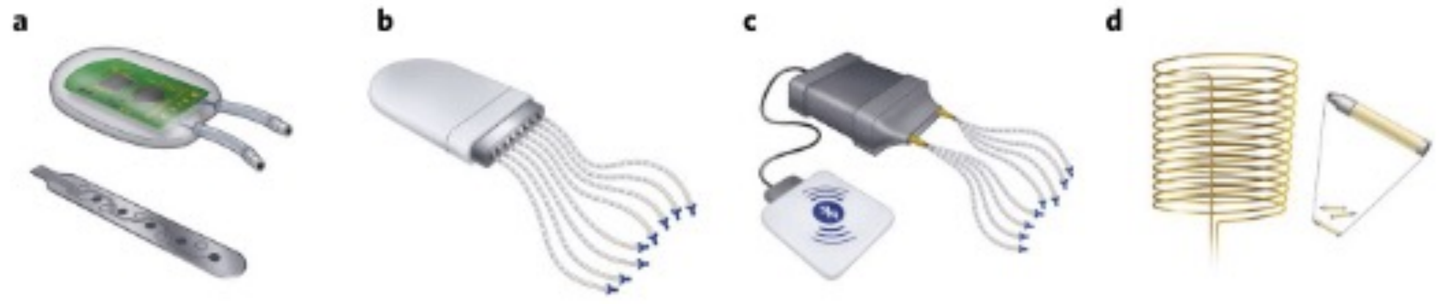
- Invasive
- Non-invasive
- Minimum invasive

Intramuscular

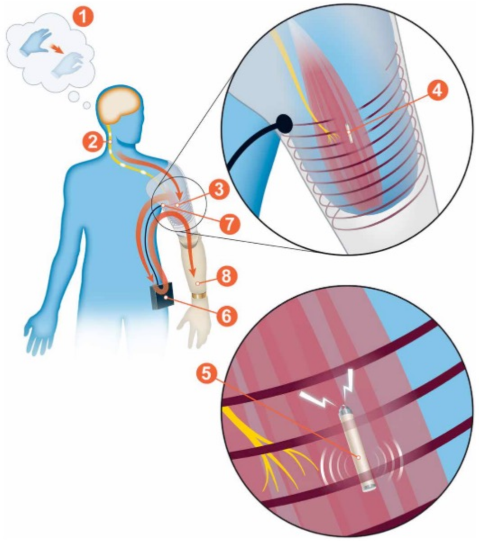
Surface

Invasive Method

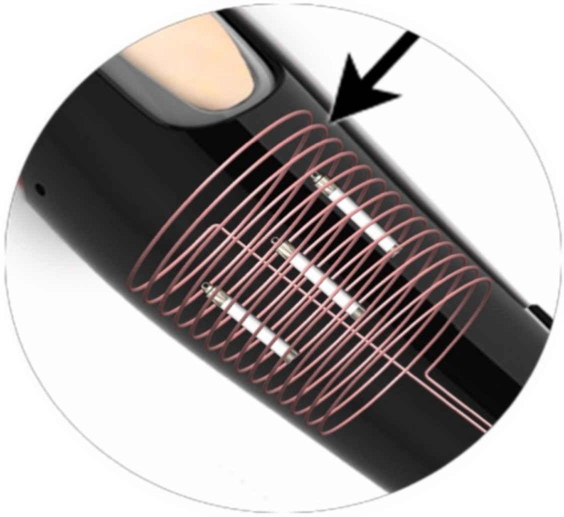
❖ Implantable electrode

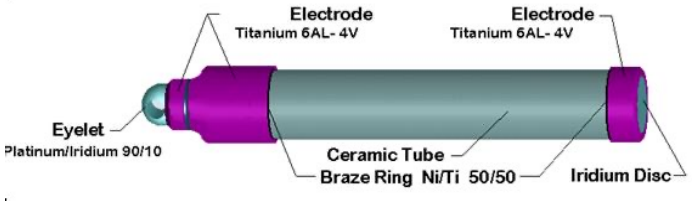


MyoPlant MIRA iSens IMES



Reverse and forward telemetry is used for the signal transmission (transcutaneous magnetic links)



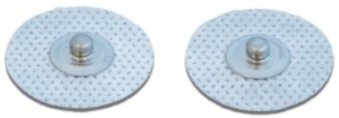


Includes differential amplifier

Salminger et al., *Science Robotics*, 2019

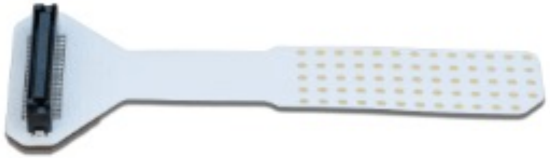
Non-invasive Method - Materials

❖ Surface EMG

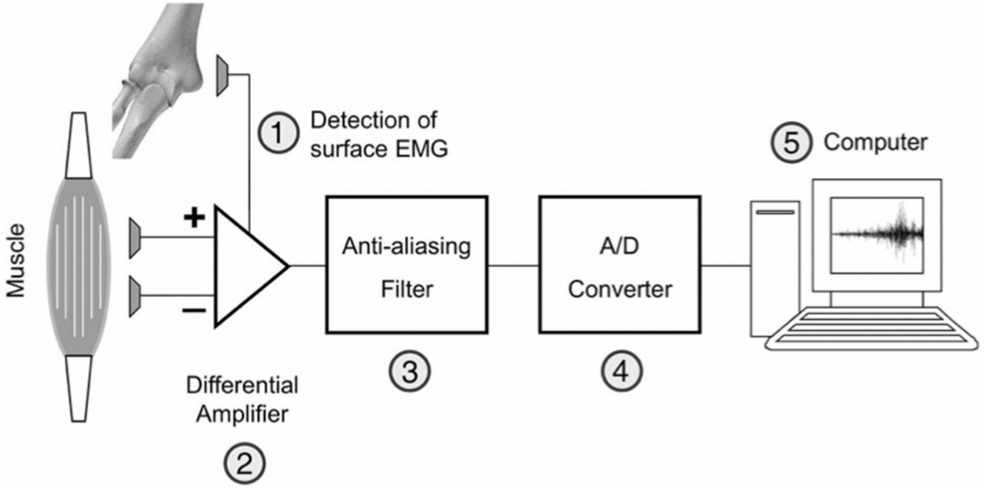


Monopolar

Bipolar



High-density



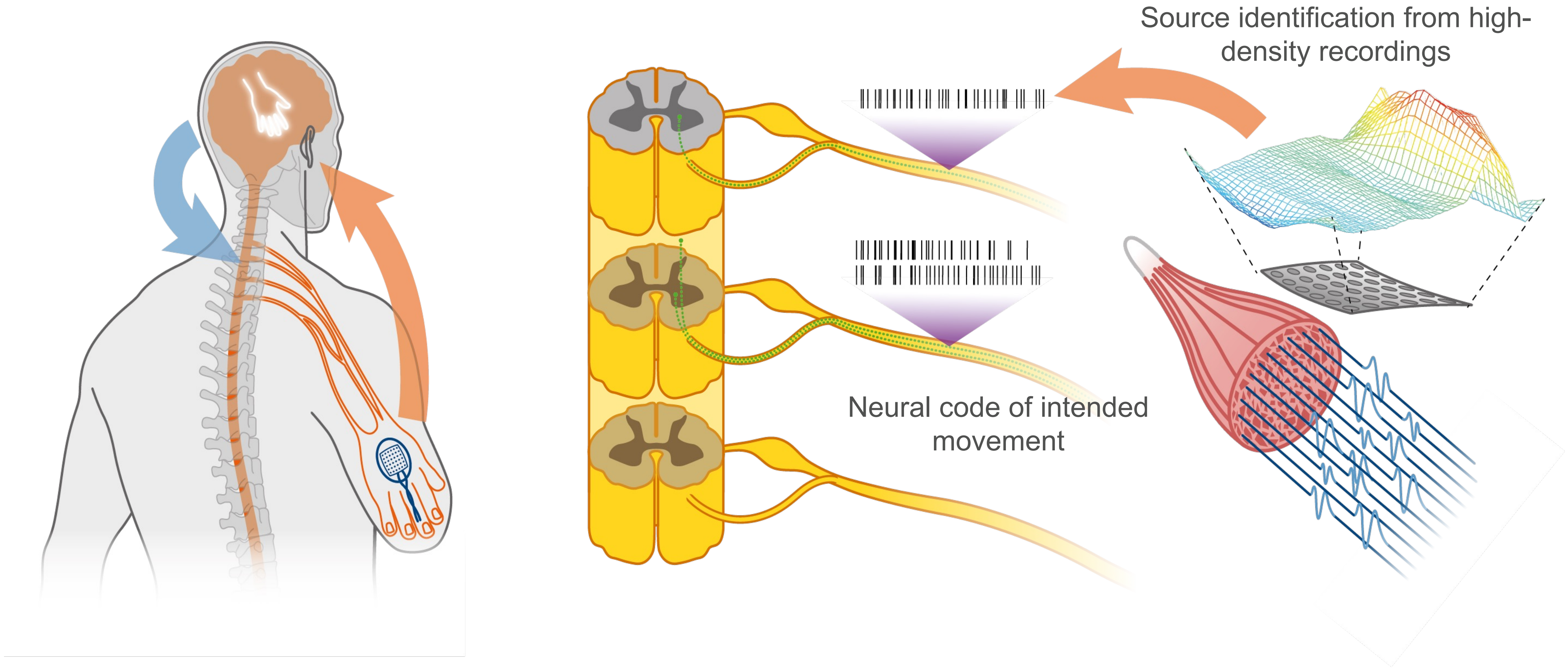
Amplifier

OT Bio elettronica

Cavalcanti Garcia & M. Vieira, 2019

Non-invasive Method – HD-EMG

❖ Spinal signal source separation by HD-EMG

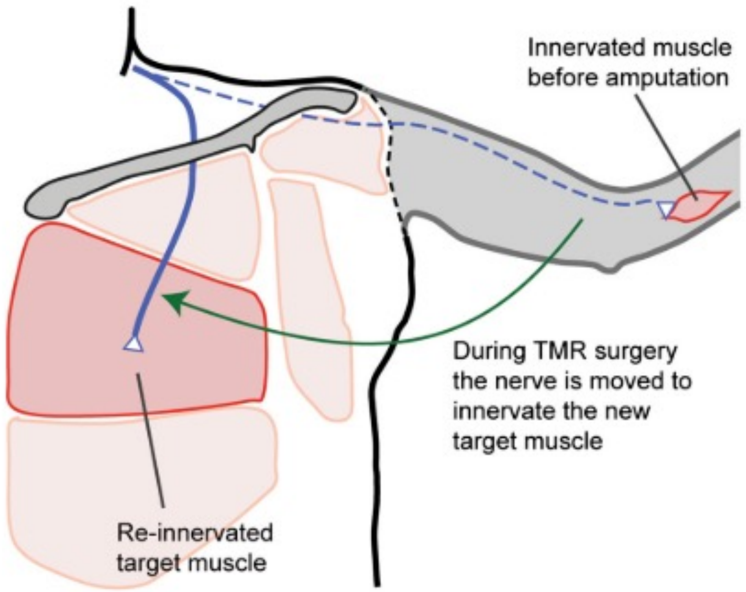


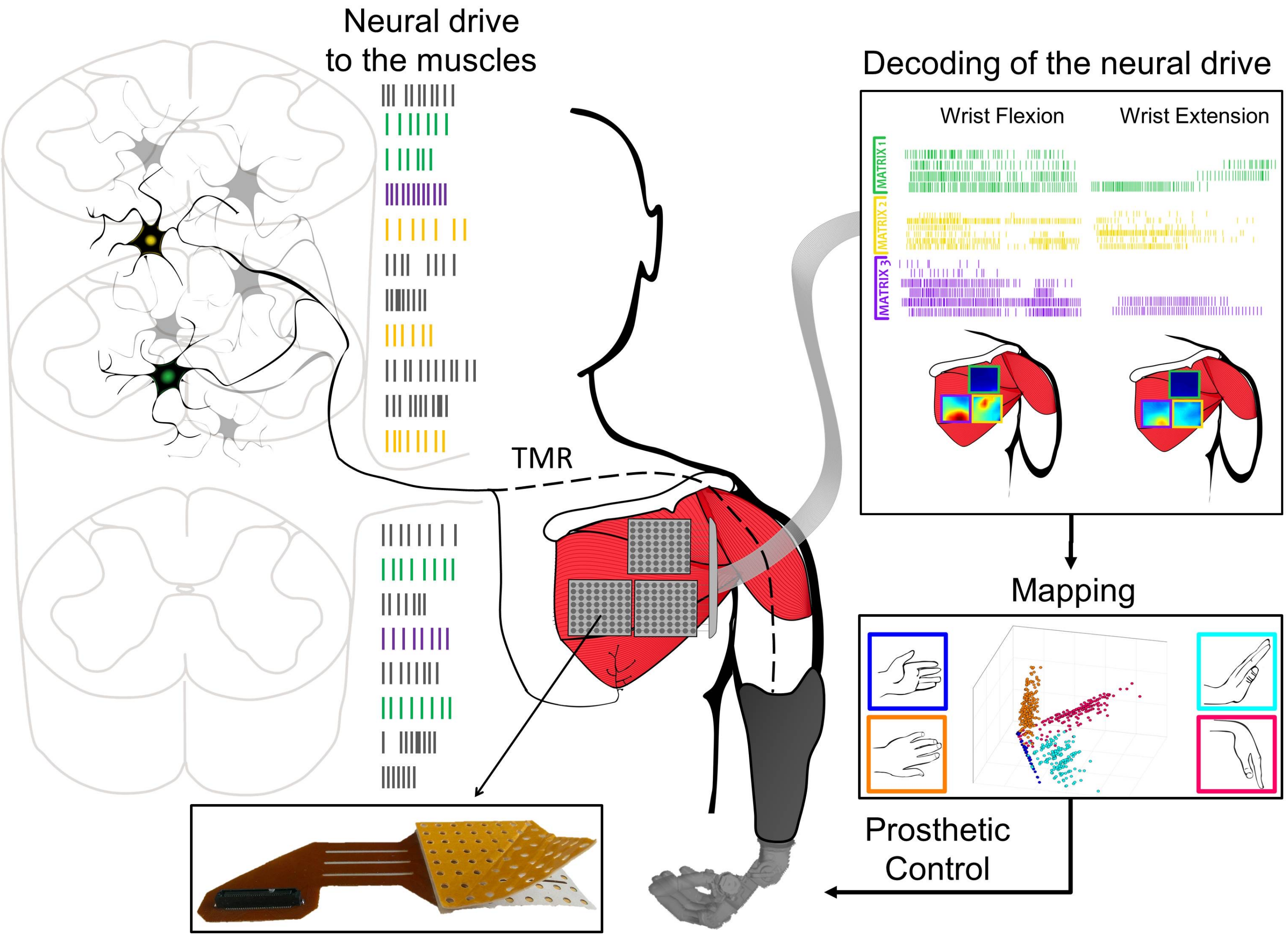
Non-invasive Method - TMR

❖ Biological signal source separation by HD-EMG with TMR surgery

Target Muscle Reinnervation

➢ The reinnervated muscle behaves as a "bioscreen" for the missing limb's neural activity





Roche et al., Current Surgery Reports, 2014

Farina et al., Nature Biomed Eng. 2017

Non-invasive Method - TMR

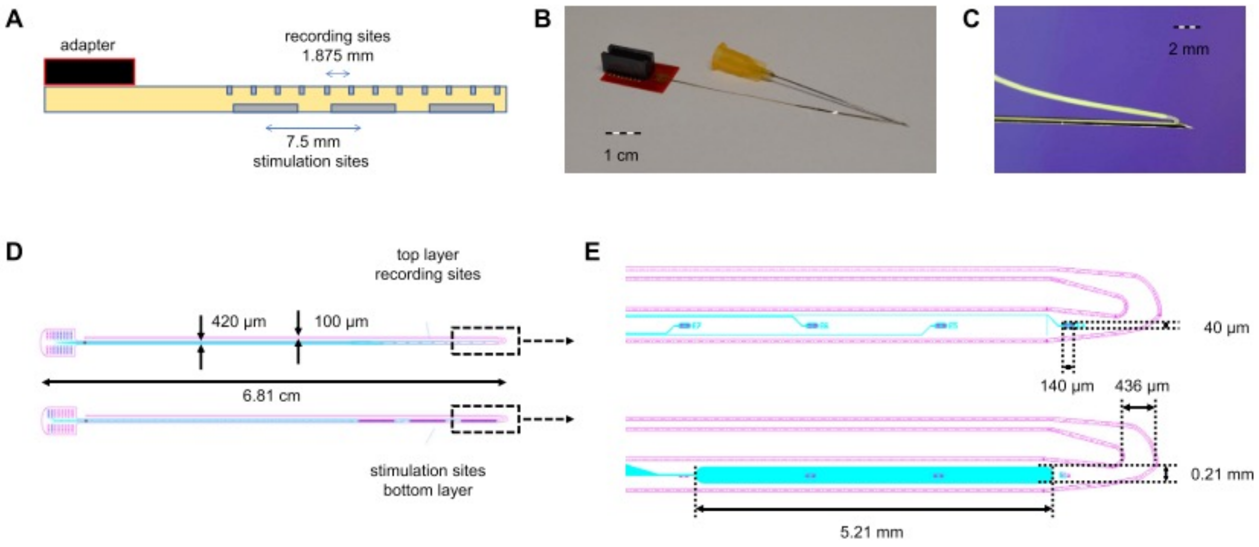
❖ Biological signal source separation by HD-EMG with TMR surgery

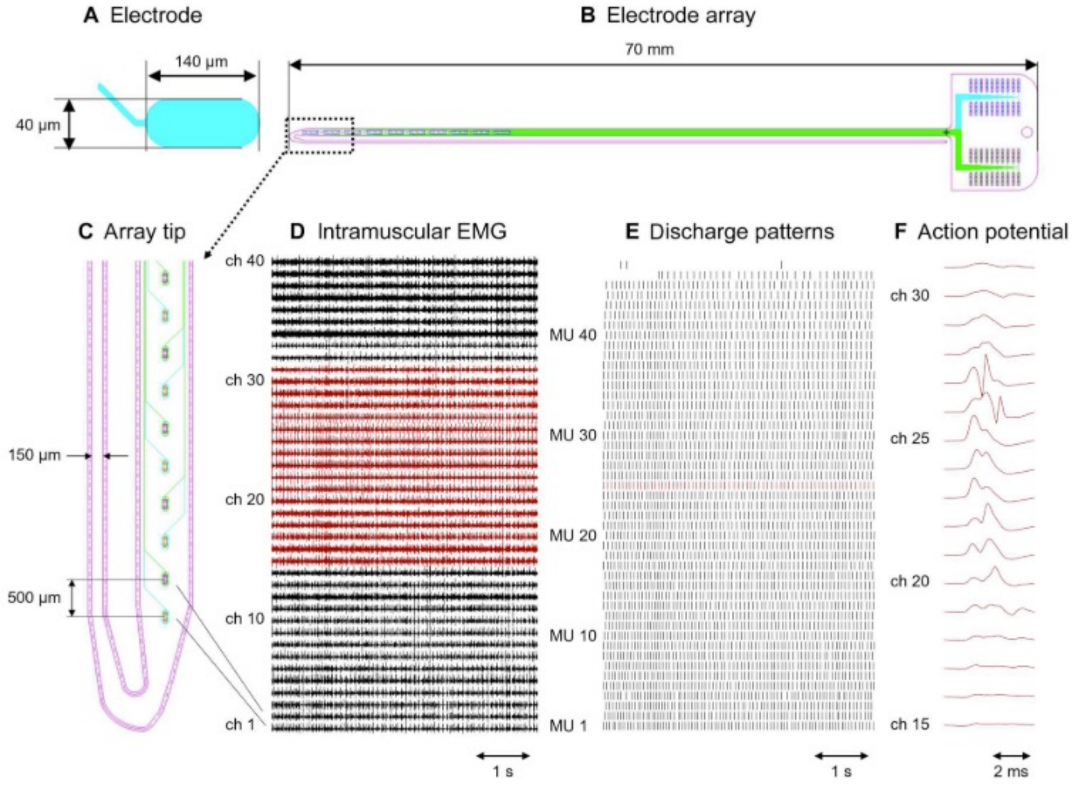
Farina et al., Nature Biomed Eng. 2017

Minimum Invasive Method

❖ Thin-film electrode to overcome sEMG barriers

- ➢ sEMG is difficult to have a connection to the skin
- ➢ sEMG's cross-talk is always the issue

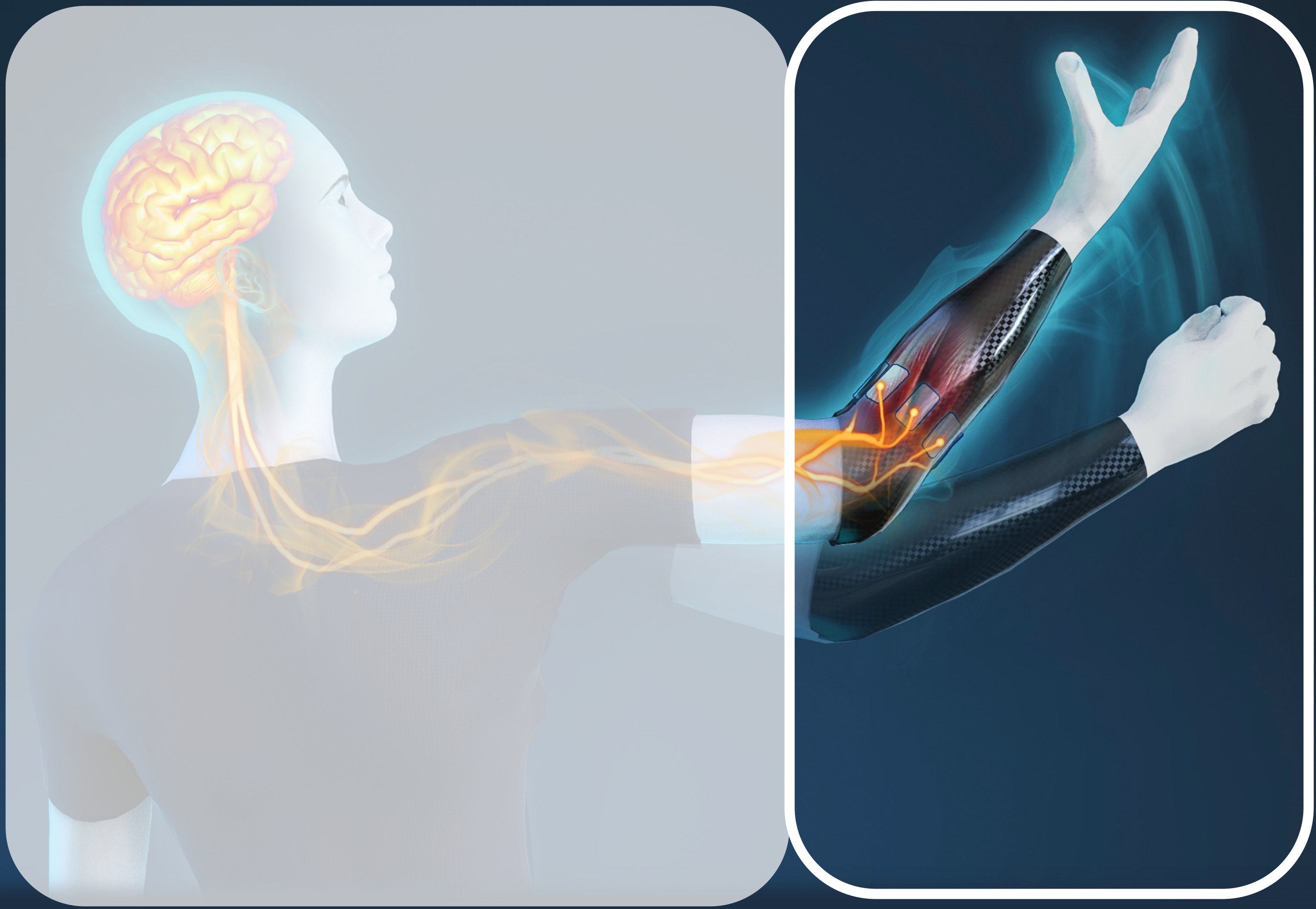




Muceli et al., Journal of Neural Eng., 2019

Muceli et al., Science Advances 2022

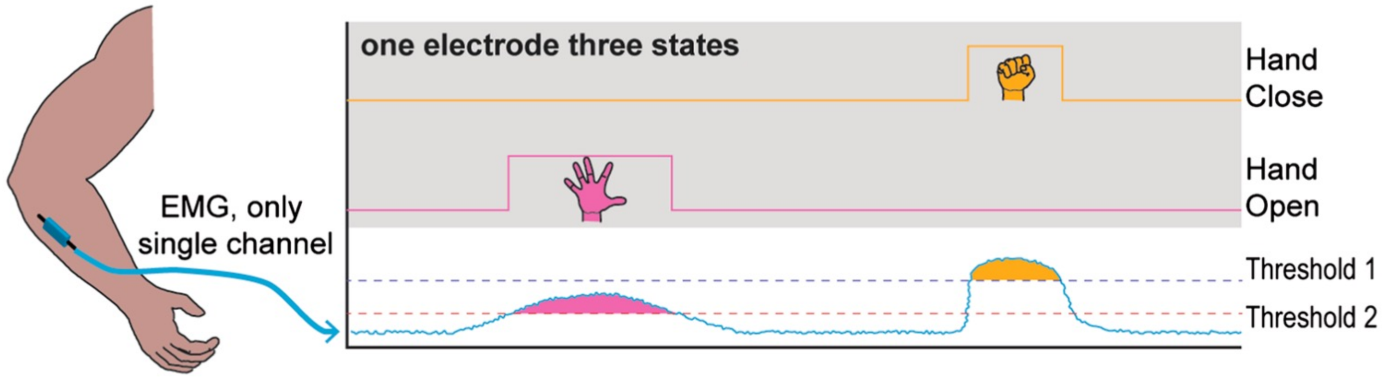
Control strategies



Control strategies – Low-density sEMG (one to two channels)

❖ Early Stage Control Strategies

Single channel threshold based



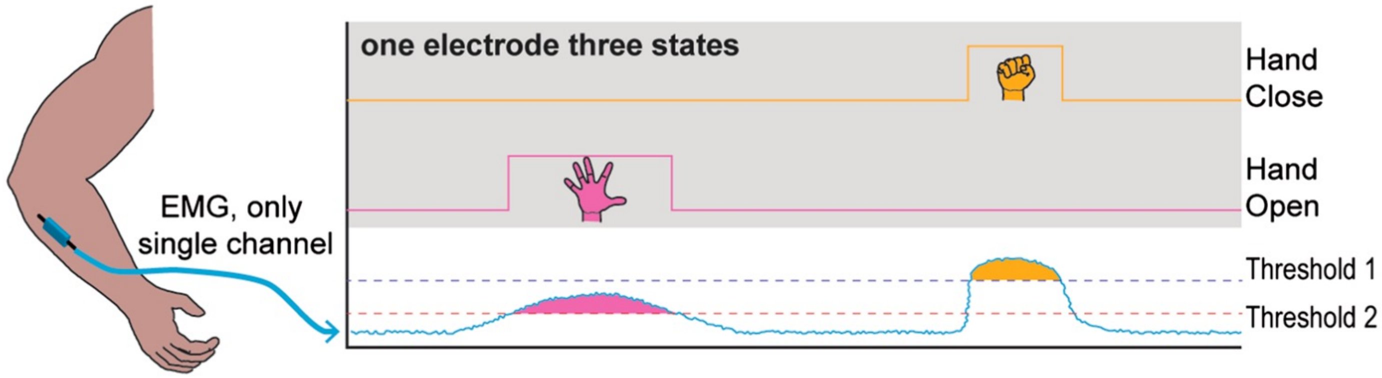
- Using one channel EMG with threshold
  - ✓ Easy to implement
  - ✓ Robust to the environmental change
  - ✗ Number of generated movements is limited

Roche et al., Current Surgery Reports, 2014

Control strategies – Low-density sEMG (one to two channels)

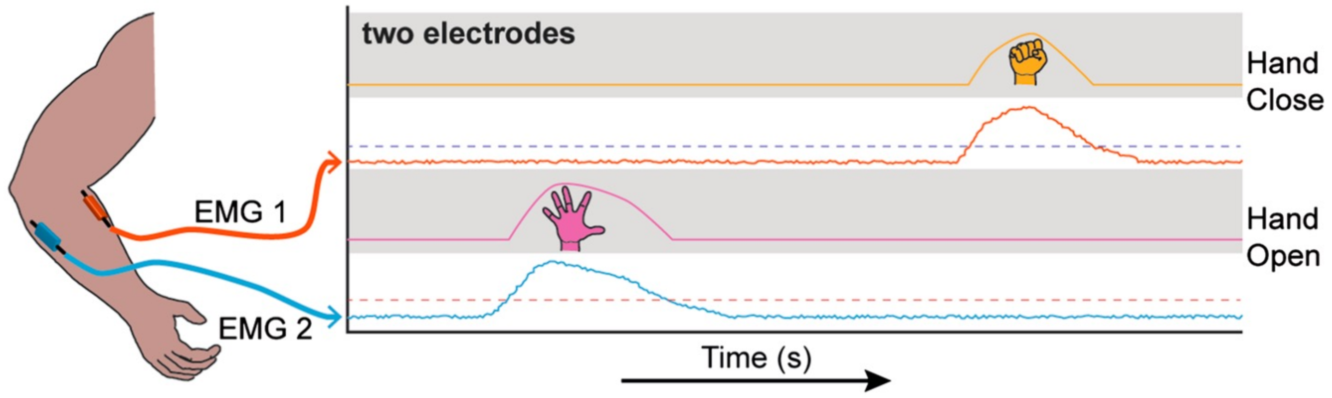
❖ Early Stage Control Strategies

Single channel threshold based



- ➢ Using one channel EMG with threshold
- ✓ Easy to implement
- ✓ Robust to the environmental change
- ✗ Number of generated movements is limited

Two opposing channels



- ➢ Controls mechanical output (force, velocity, position, etc)
- ➢ Used in commercially available hands (Michelangelo, Hero Arm)
- ✓ Increase in gestures
- ✗ Insufficient for finer movement (e.g., individual finger flexion)
- ✗ Far from multifunctional control

Roche et al., Current Surgery Reports, 2014

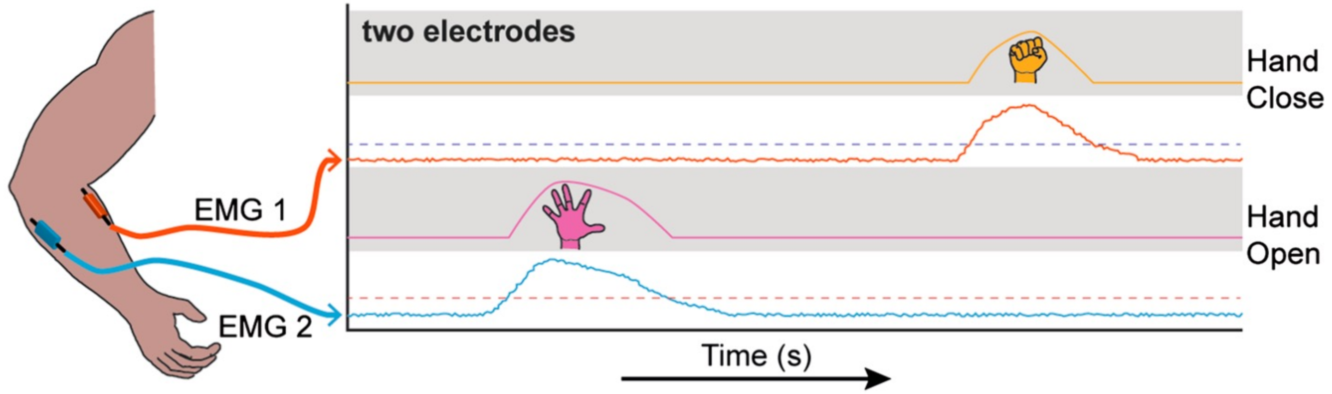
Control strategies – Low-density sEMG (one to two channels)

❖ Early Stage Control Strategies

Single channel threshold based

- ➢ Using one channel EMG with threshold
- ✓ Easy to implement
- ✓ Robust to the environmental change
- ✗ Number of generated movements is limited

Two opposing channels



- ➢ Controls mechanical output (force, velocity, position, etc)
- ➢ Used in commercially available hands (Michelangelo, Hero Arm)
- ✓ Increase in gestures
- ✗ Insufficient for finer movement (e.g., individual finger flexion)
- ✗ Far from multifunctional control

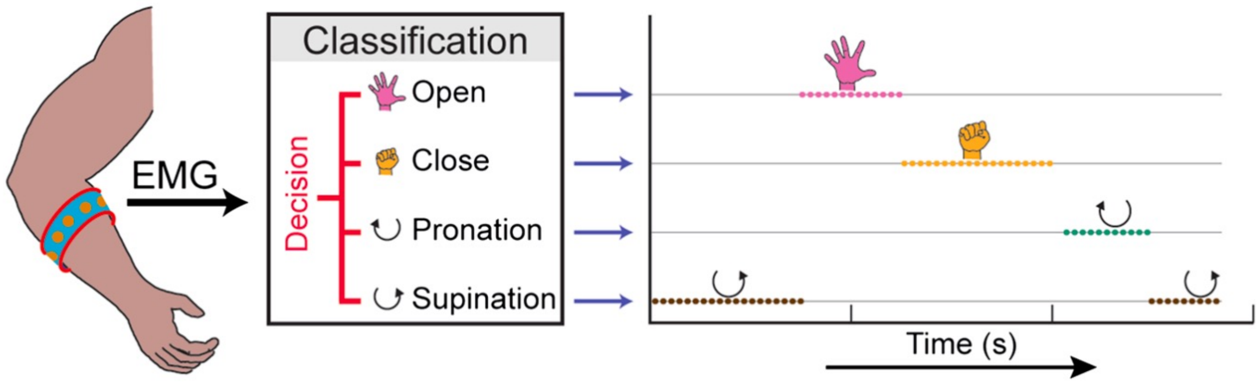
More advanced control strategies have been proposed

Roche et al., Current Surgery Reports, 2014

Control strategies – Low-density sEMG (8 to 16 channels)

❖ Machine Learning Approach

Pattern Recognition Control



- In sequence, not simultaneously & proportionally (on/off basis)

Methods: Support Vector Machine, K-Nearest Neighbor, Artificial Neural Networks

✓ Can achieve multiple gestures

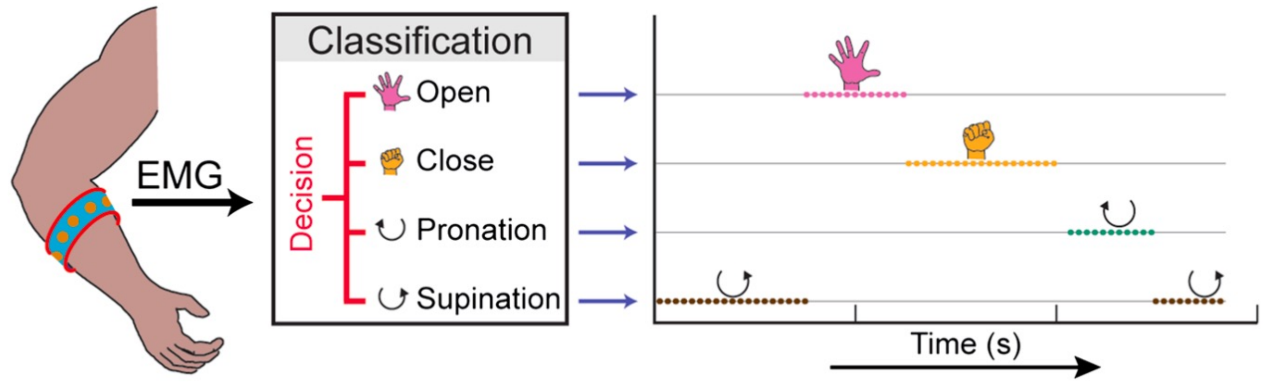
✗ Difficult to translate into real-world situations (muscle contraction patterns change, re-calibrate, classification accuracy, etc)

Roche et al., Current Surgery Reports, 2014

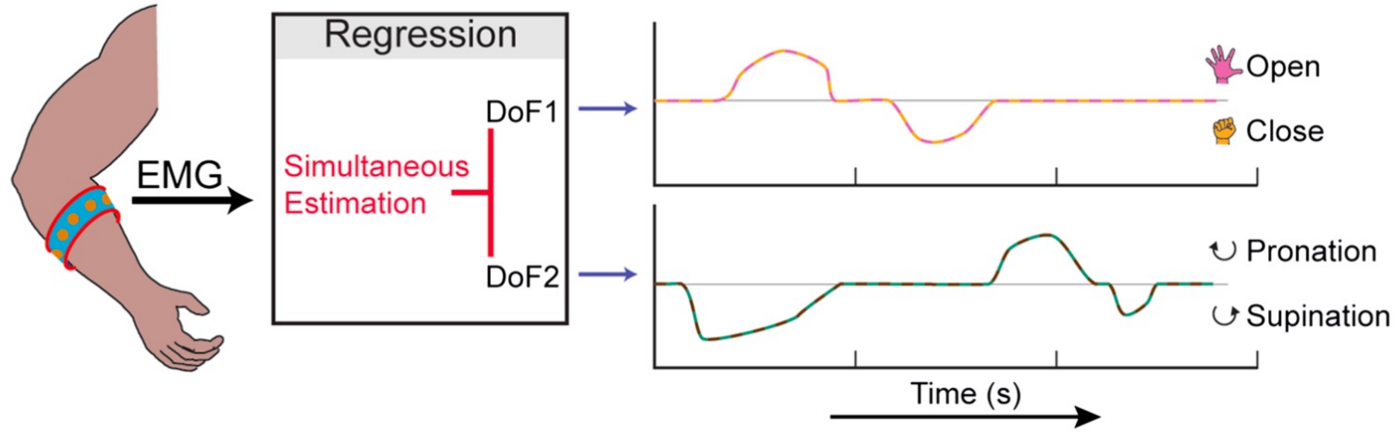
Control strategies – Low-density sEMG (8 to 16 channels)

❖ Machine Learning Approach

Pattern Recognition Control



Regression Control



➢ In sequence, not simultaneously & proportionally (on/off basis)

Methods: Support Vector Machine, K-Nearest Neighbor, Artificial Neural Networks

✓ Can achieve multiple gestures

✗ Difficult to translate into real-world situations (muscle contraction patterns change, re-calibrate, classification accuracy, etc)

➢ Simultaneous & proportional control

Methods: Negative Matric Factorization, Artificial Neural Networks, Linear Regression

✓ Intuitive

✗ Stable control limited to 2DoF due to cross-talk

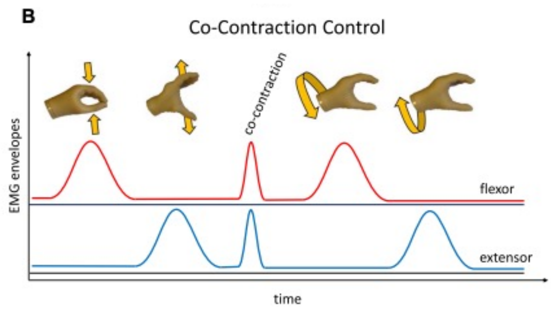
✗ High separation of the DoF matters

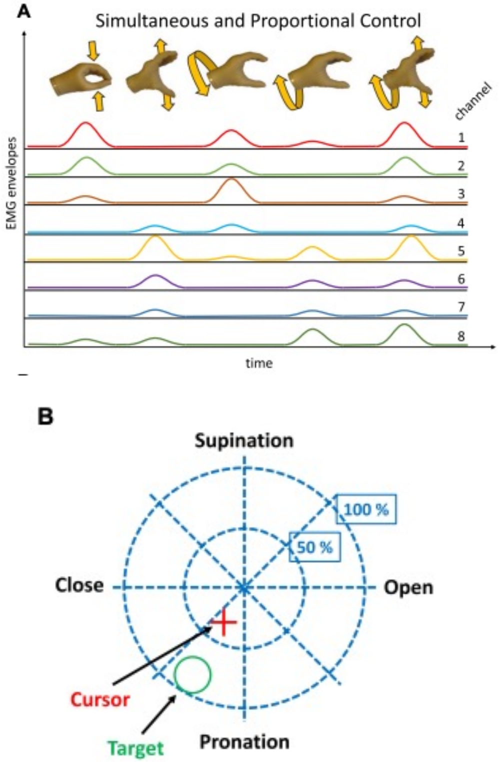
✗ Still limited to the lab phase

Roche et al., Current Surgery Reports, 2014

Control strategies – Low-density sEMG (8 to 16 channels)

❖ Linear Regressor (2DoF) by 8 EMG Channels





Conventional direct control

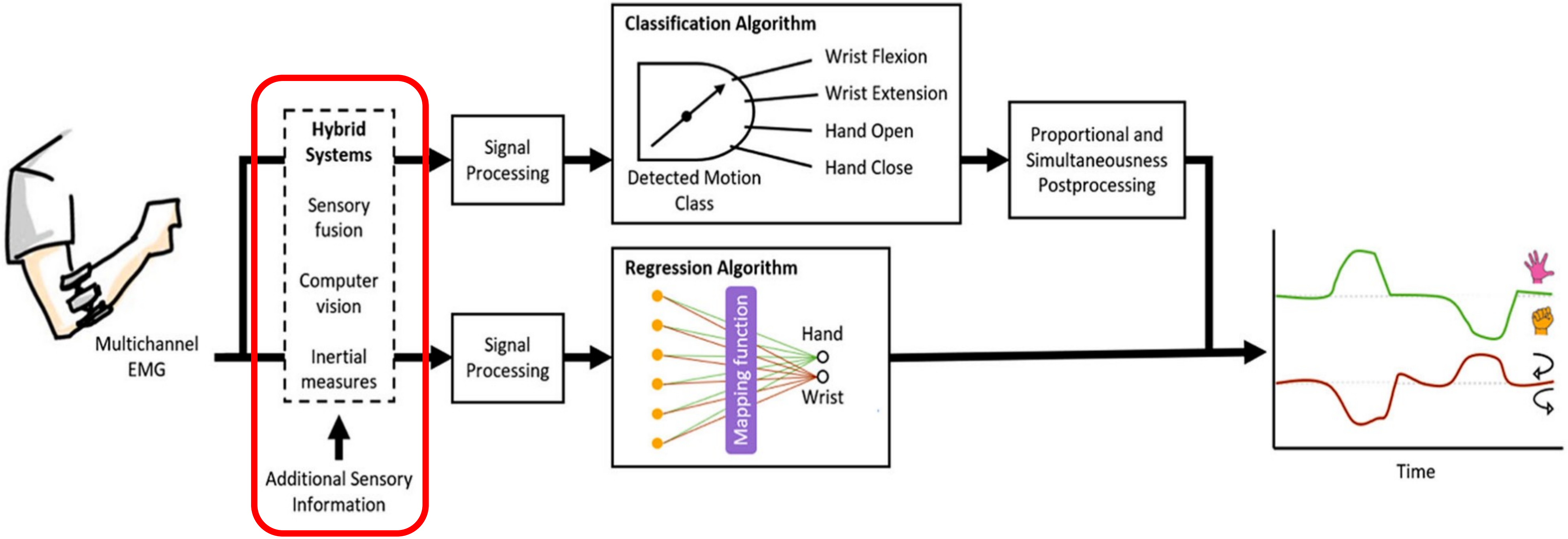
Linear regressor

Intuitive and quicker to complete the task

Hahne et al., *Science Robotics* 2018

Control strategies – Low-density sEMG Hybrid Control

❖ Hybrid Systems to Overcome Limitations of Regression Control



➢ Additional information to improve the control accuracy

Methods: Multi-modal sensory fusion, Computer vision, etc

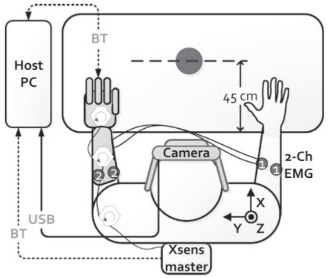
✓ Can acquire information (contextual information) that cannot capture by EMG

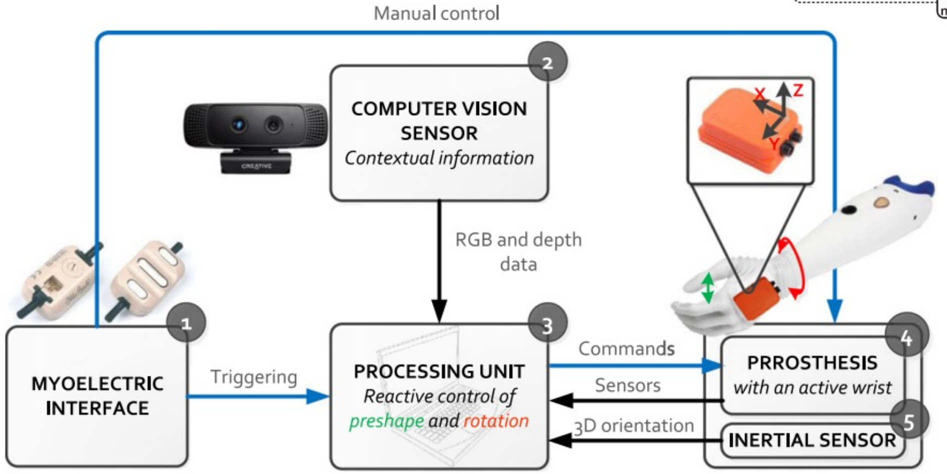
✗ Still limited in the lab environment

Roche et al., Current Surgery Reports, 2019

Control strategies – Low-density sEMG Hybrid Control

❖ Hybrid system – Computer vision





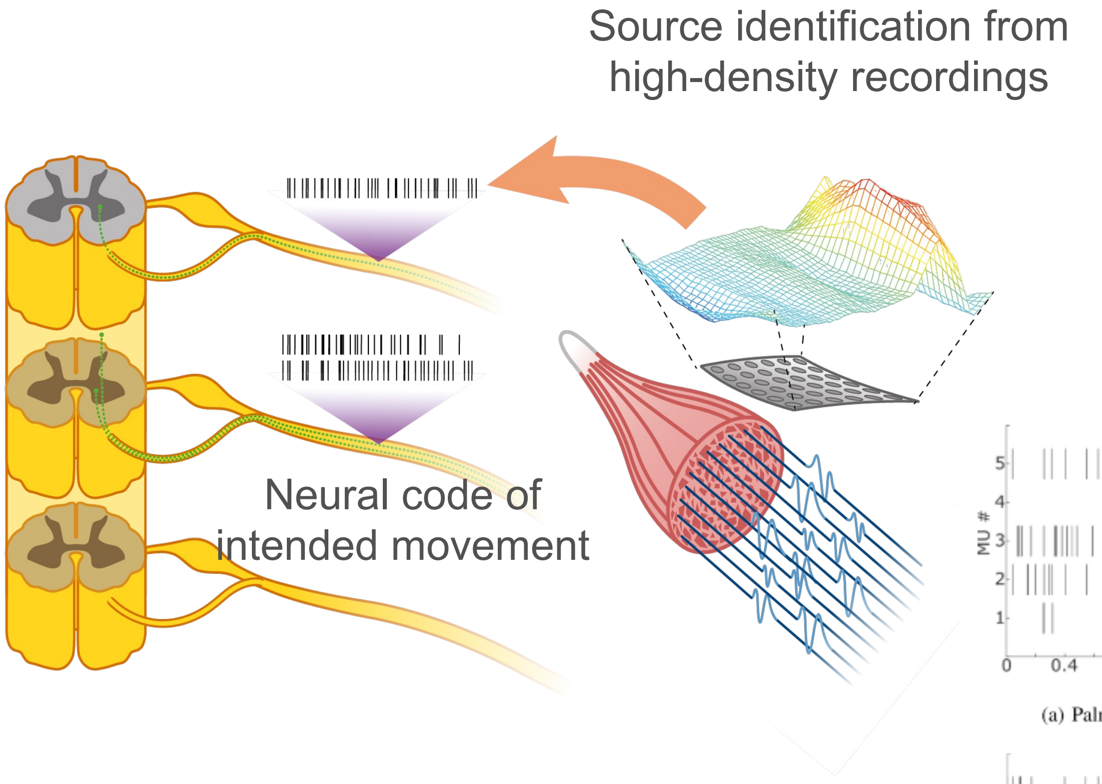
- Simultaneous & proportional control of 3 DoF based on the fusion of computer vision and position/orientation measurements
- Fine-tune the autonomous decisions (semi-automatic control)

Smoother & higher DoF control than conventional method

Markovic et al. *J Neural Eng.* 2015

Control strategies – High-density sEMG

❖ Motor Unit decomposition



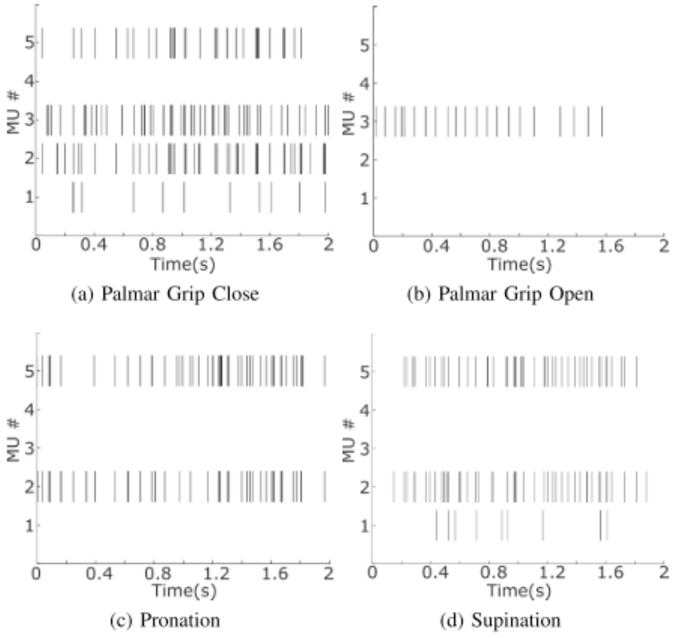
➢ Motor units activity decomposition by HD-EMG

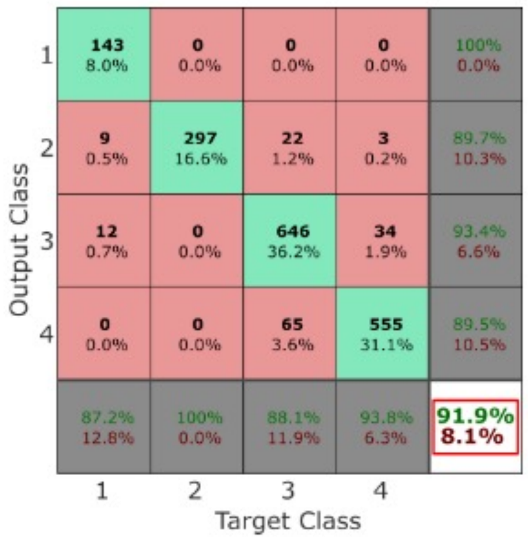
Methods: Blind source separation

✓ Reliable information about neural activity

✗ Increases the computational burden required to the system

Real-time motor unit decomposition
gesture classification (FPGA based)



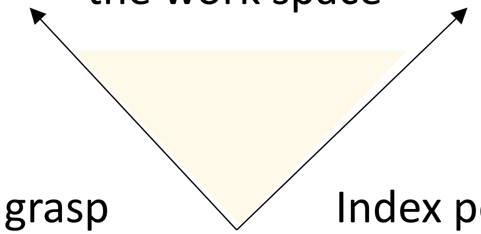


SVM, Class 1: Palmar grip close, Class 2: Palmar grip open, Class 3: Pronation, Class 4: Supination

Barsakcioglu and Farina *IEEE Biomedical Circuits and Systems Conference*. 2018

Control strategies – High-density sEMG

❖ Synergistic approach



Full integration for true bionic interfacing

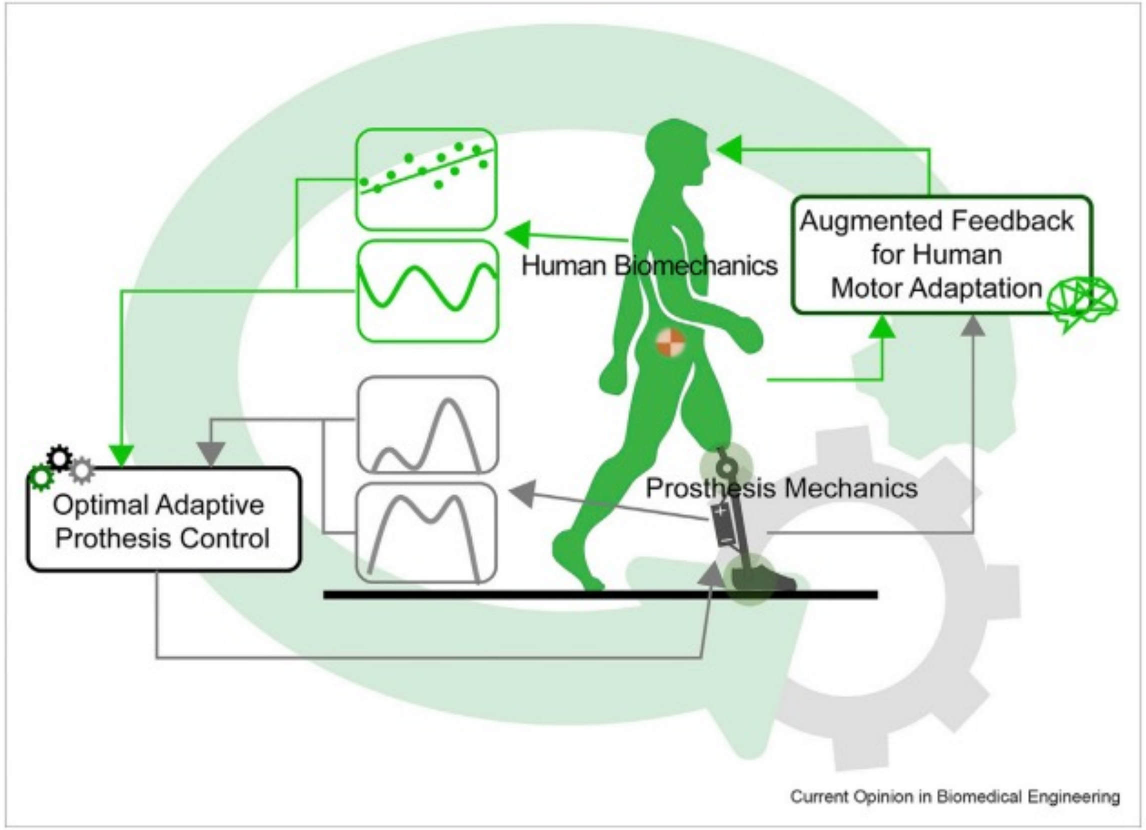
Exploring hand capabilities: Index point (y axis).

(in progress) Barsakcioglu et al,. 2023

Control strategies – Adaptive Technique

❖ Human-prosthesis symbiosis

➢ Human motor control and intelligent prosthesis control function as one system



- Continuous joint coordination

➢ A controller continuously monitors the gait cycle and adjust the joint angles based on the predefined joint coordination

✓ Better adaptability to varied walking speeds and inclination angle

✗ Tuned manually and individually

- Reinforcement learning

➢ Tune the adaptive parameters in real-time

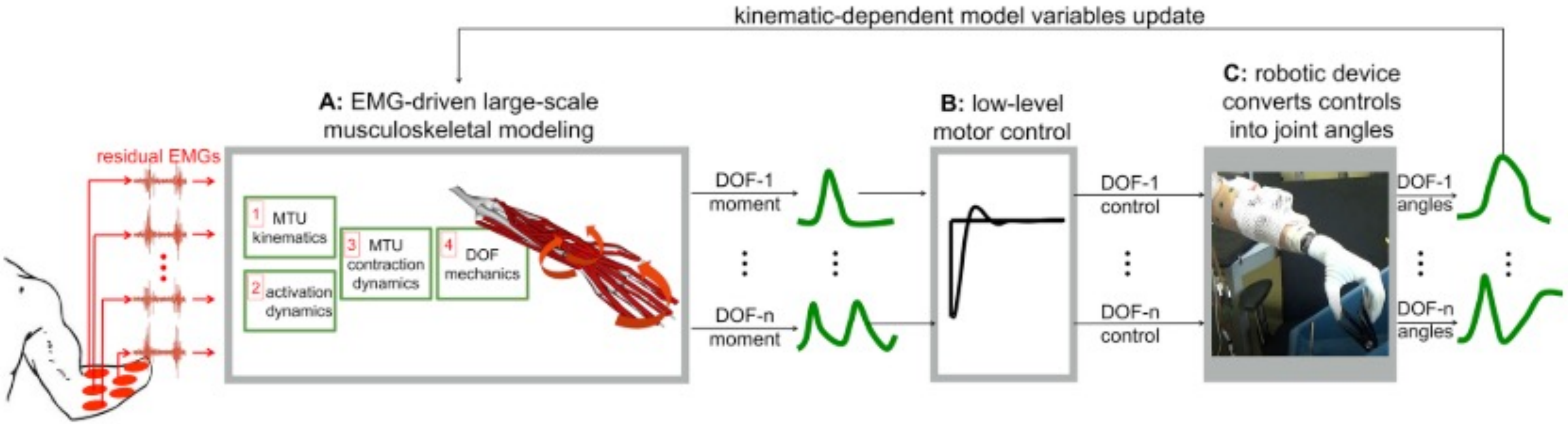
✓ Model free and prior tuning knowledge free

Huang et al. Curr. Opin. Biomed. Eng. 2021

Control strategies – Model-based approach

❖ Biological limbs' joint rotation can be generated by different EMG patterns
❖ Vary across individuals, training conditions, arm postures, or tasks

❖ Forward musculoskeletal model mimics the biological process
➢ Directly incorporates physiological and biomechanical constraints
➢ With TMR, this can reconstruct the internal biomechanical representation of missing limbs



Sartori et al., Journal of Neural Eng., 2018

Future Direction

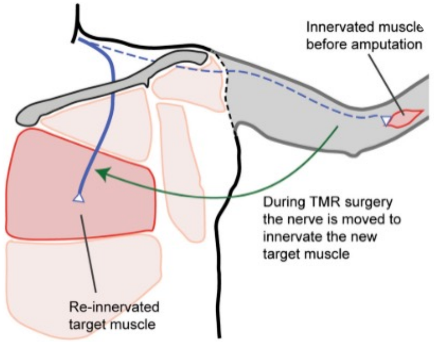
❖ Next generation clinical prostheses' key areas

Farina et al., *Nature Biomed. Eng.* 2021

Future Direction

❖ Next generation clinical prostheses' key areas

Target muscle reinnervation



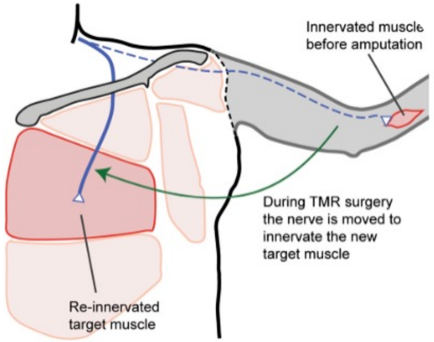
Farina et al., *Nature Biomed. Eng.* 2021

Future Direction

❖ Next generation clinical prostheses' key areas

Chronically implantable myoelectric sensors

Target muscle reinnervation

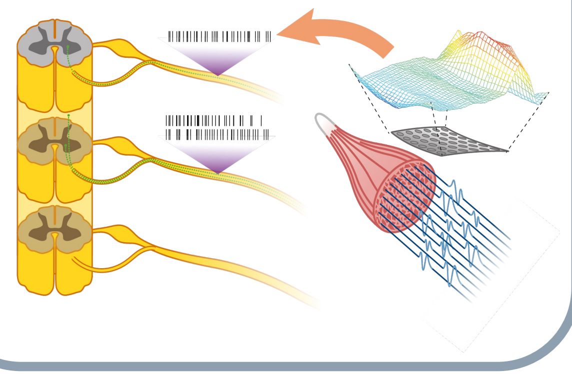


Farina et al., *Nature Biomed. Eng.* 2021

Future Direction

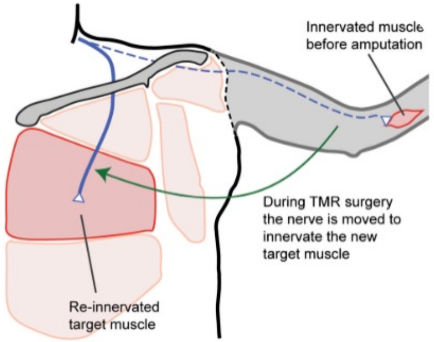
- ❖ Next generation clinical prostheses’ key areas

Advanced control algorithms



Chronically implantable myoelectric sensors

Target muscle reinnervation



Farina et al., *Nature Biomed. Eng.* 2021

Hands on!
- Offline algorithm development-

Mini Project System Overview

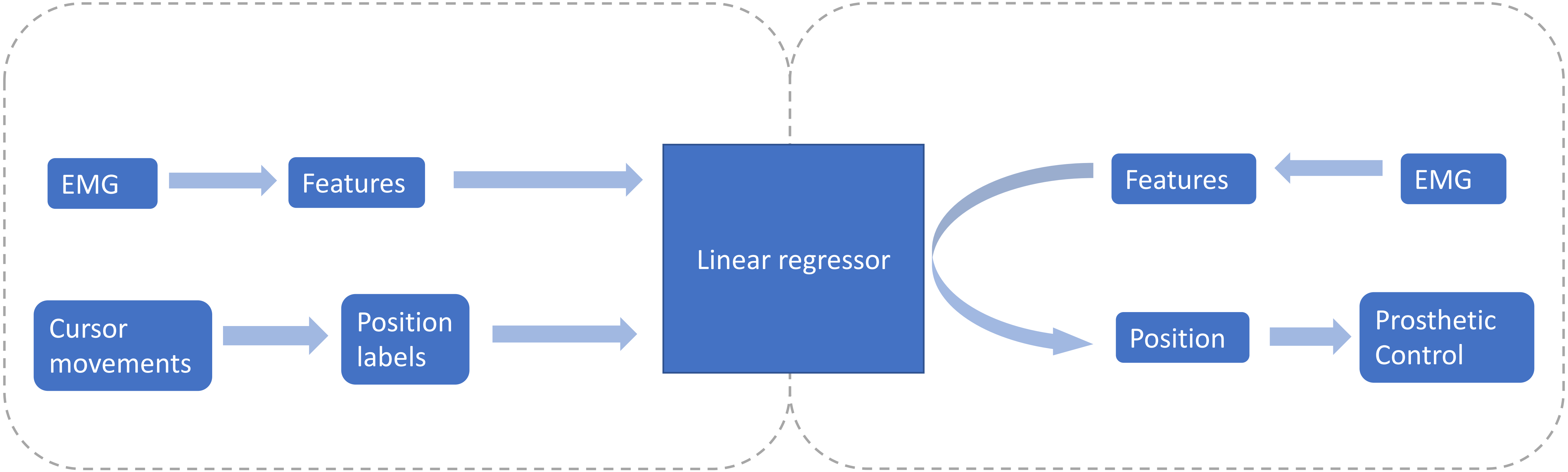
1st Part: Demonstration real time system

Demonstrate with MyoControl.m by organiser:

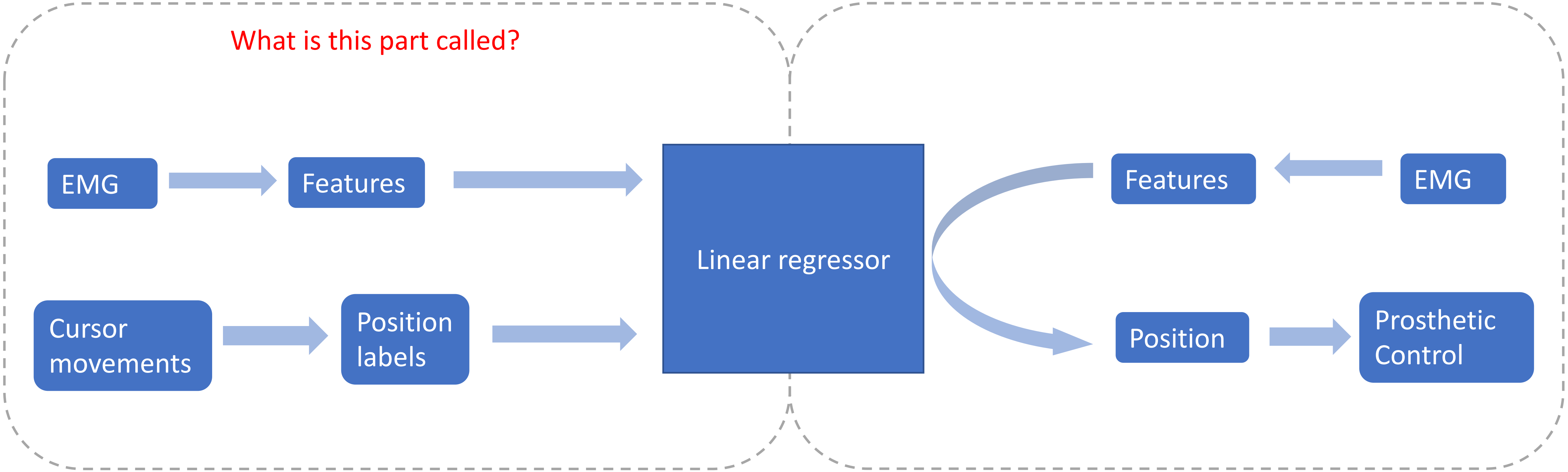
1. App = MyoControl
2. Data collection
3. Training regressor
4. Real time control cursor on virtual cursor

Offline analysis:
Feature extraction and linear regression

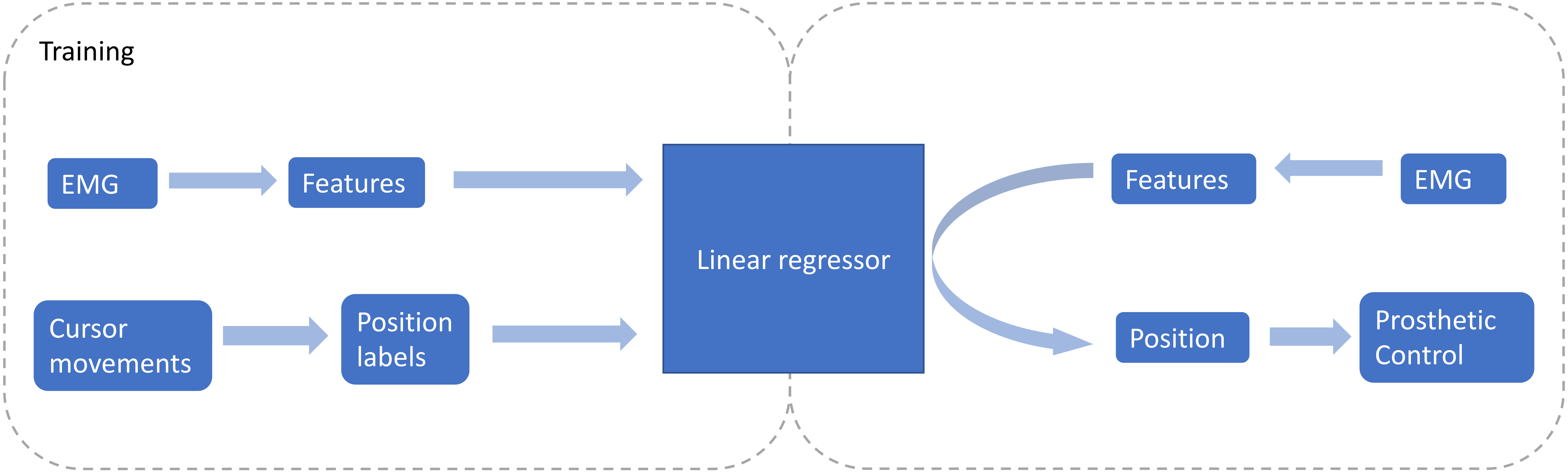
Training the algorithm



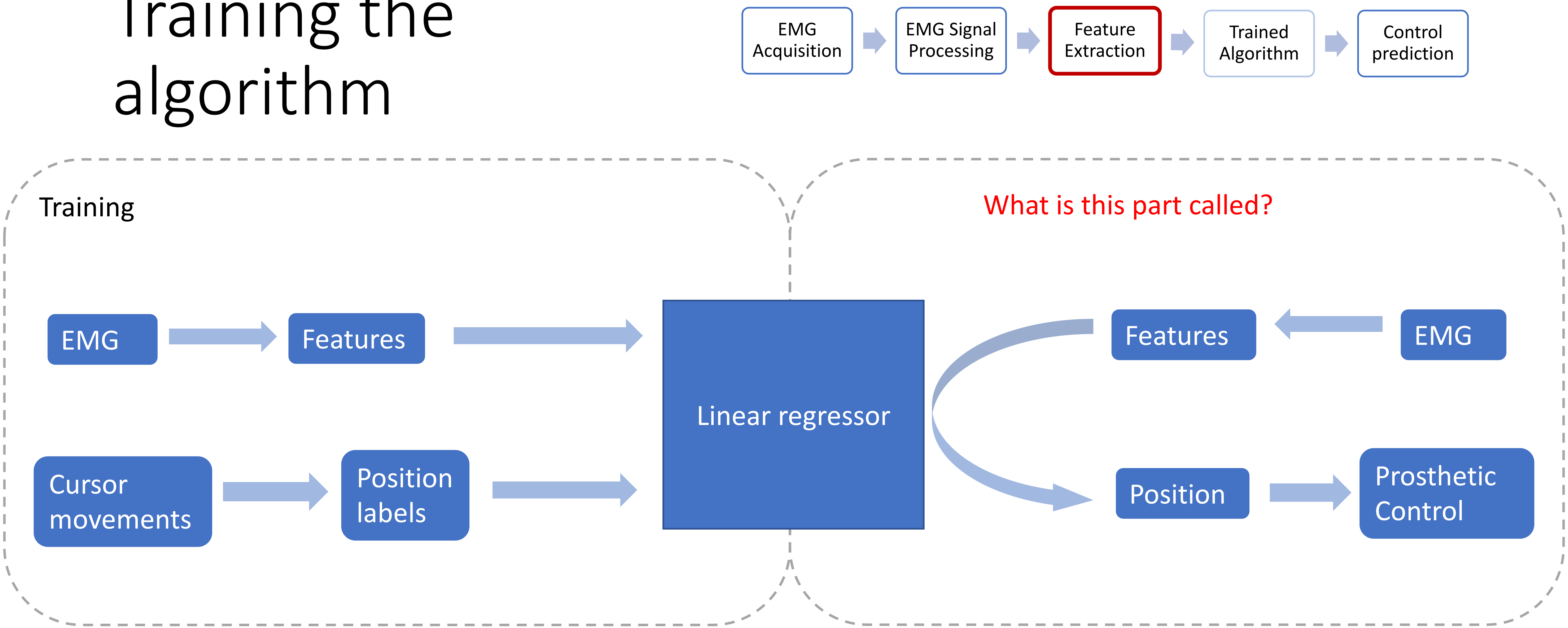
Training the algorithm



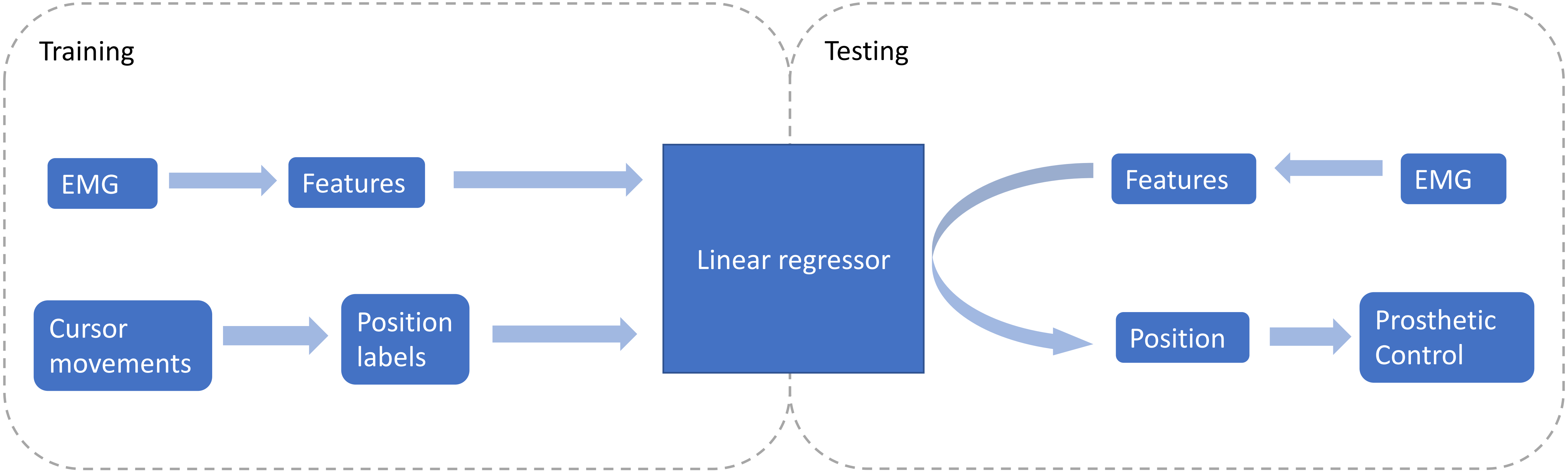
Training the algorithm



Training the algorithm



Training the algorithm



File name: MyoAnalysis.m

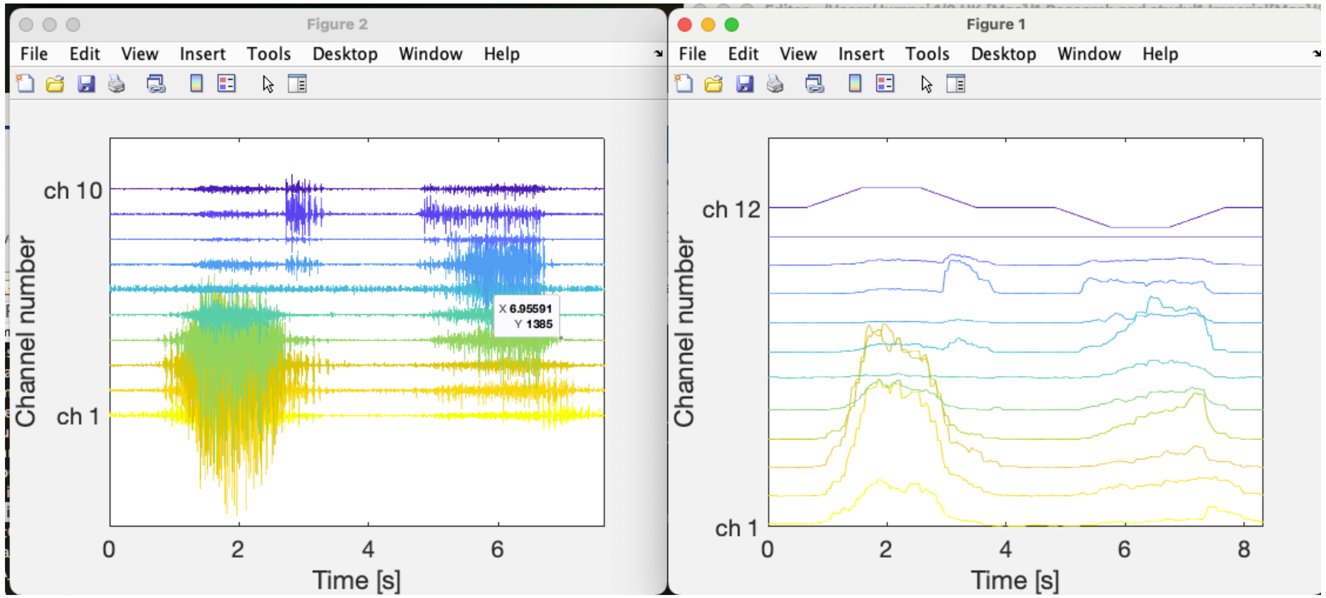
1. Plot Trial

Visualise the sEMG signals

- Check the the sEMG signals on pre-recorded data – rawEMG and rms
- “Demo data” -> “TrainingData” -> any data you want

```
%% 1. Plot trial
%close all;
% Get the file
[nfile,path] = uigetfile();
file         = load(fullfile(path,nfile));

% Input data to test with the regressor
RecInfo      = file.RecInfo;       % Rms data to test

figure(1)
nchan        = size(RecInfo.EMGRMS,2);                        % define the number of channels
% Plotting function
% plotCh(data, number of channels, Sampling frequency)
dataToPlot   = [RecInfo.EMGRMS';RecInfo.Labels'];             % to include in the plot the labels
plotCh(dataToPlot.*10, nchan+2, 25)                        % Plot data and labels, 25 Hz frequency

% Plot of the rms Data
figure(2)
plotCh(RecInfo.EMGraw', nchan, 1000)
```

Result



File name: MyoAnalysis.m

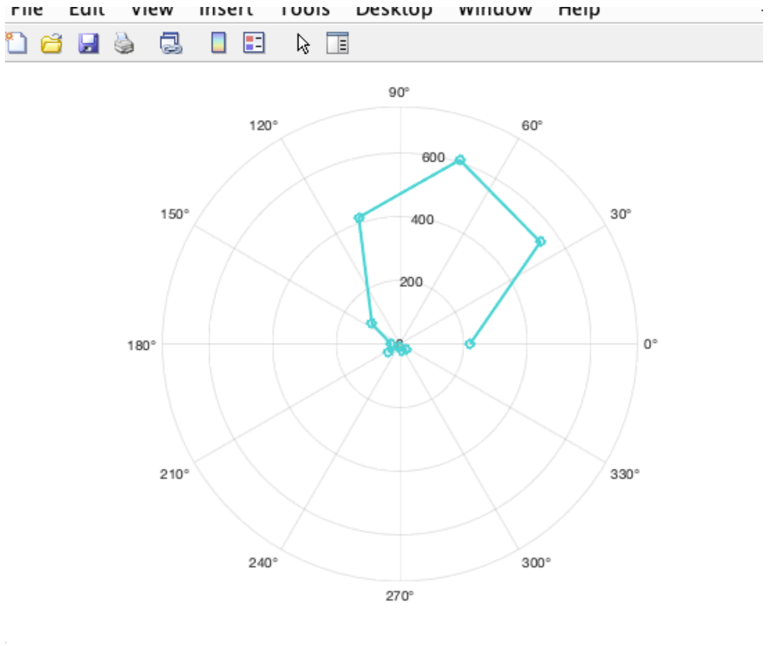
2. Plot over time

Visualise the sEMG signals

- Plotting the EMG RMS value in a polar plot

```matlab
%% 2. Plot over time
%close all;

FP        = figure(3);
FP.Color = [1,1,1];
rlim      = max(max(RecInfo.EMGRMS));
%theta     = deg2rad([(36):36:(360)]);
theta     = deg2rad([0:360/nchan:360]);
for sample = 1 : length(RecInfo.EMGRMS)

    FP = polarplot(theta(1:end-1),RecInfo.EMGRMS(sample,1:nchan),'o-','linewidth',2,'color',[0, 0.8, 0.8]);
    %FP = polarplot(theta,RecInfo.EMGRMS(sample,[2:10,1]),'o-','linewidth',2,'color',[0, 0.8, 0.8]);

    hold off
    FP.Parent.RLim = [0,rlim];

    pause(0.04)
    drawnow;
end
```

Result



File name: MyoAnalysis.m

3.A - Concatenate all data

```
%% 3.A - Concatenate all data

% Concatenate data
path    = uigetdir();
% Scan all the recording files in the folder
ToScan  = fullfile(path,'/*rec*.mat');
files   = dir(ToScan);

% Arrays to concatenate the data
rawEMG      =[];          % Array for the raw EMG
feats_aux   =[];       % Array for the rms EMG
labels_aux  =[];      % Array for the training labels

for nfile = 1: length(files)
    % Load the file - training data
    FileToLoad  = fullfile(path,files(nfile).name);
    load(FileToLoad);

    fprintf(strcat('Loading file : ''', files(nfile).name, '''\n'));

    % Joint data
    feats_aux   = [feats_aux;  RecInfo.EMGRMS];
    labels_aux  = [labels_aux; RecInfo.Labels];
    rawEMG      = [rawEMG;     RecInfo.EMGraw];
end
```

Concatenate the existing data

- ”DemoData” -> ‘’TrainingData”-> open

3.B - Plotting of the training data and labels

```
%% 3.B - Plotting of the training data and labels

% Plot of the Raw Data
figure(2)
plotCh(rawEMG', nchan, 1000)

% Plot of the raw Data and labels
figure(3);
nchan = size(feats_aux,2);
plotCh([feats_aux';labels_aux'].*10, nchan+2, 25)
```

3.C - Implement RMS feature extraction given a raw EMG window – [T1A_ExtractRms.m]

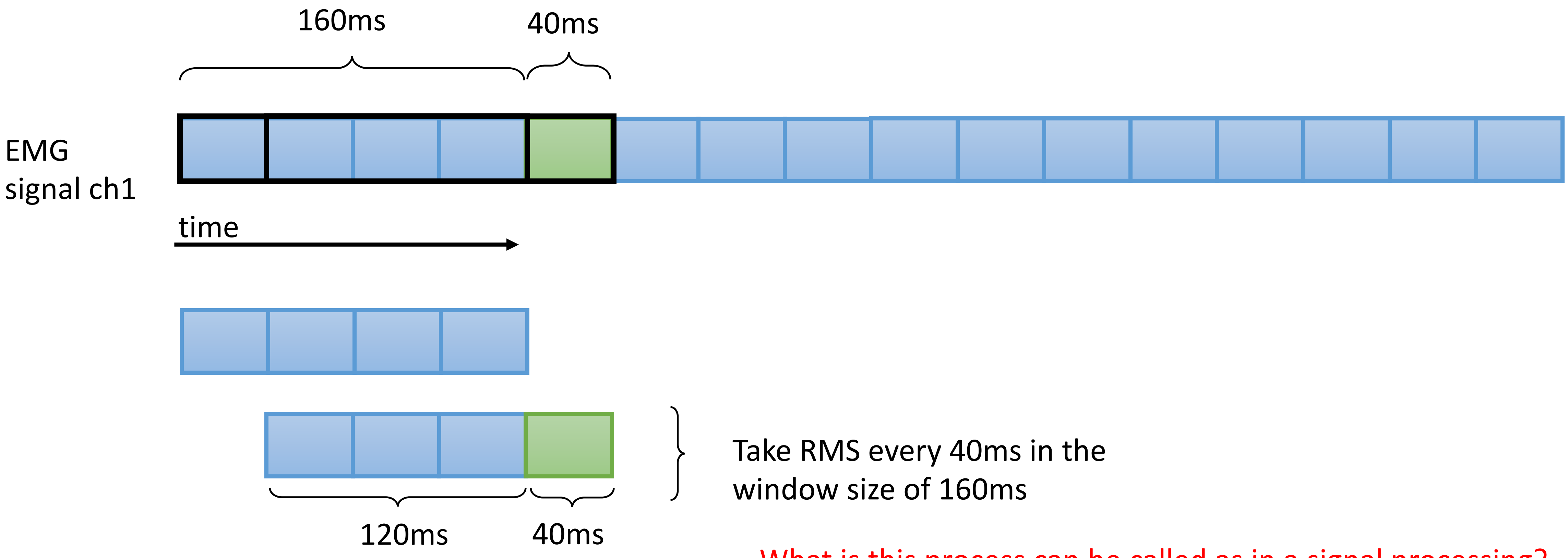
Task 1: Create RMS function – 10min

```
% DataRmsEMG = T1A_ExtractRMS(DataInRawEMG)

dOut(samplect,:) = T1A_ExtractRms(DataI);

% TO COMPLETE BY THE ST
samplect = samplect + 1

end
```

Evaluate Selection in Command Window	⇧F7
Open "T1A_ExtractRms"	⇧⌘D
Help on "T1A_ExtractRms"	F1
Cut	⌘X
Copy	⌘C

```
%% 3.C - Extract features: RMS
Rmswindow    = 160;
BufSize      = 40;
samplect     = 1;
nchan        = size(feats_aux,2);

% RMS data output
dOut         = zeros(round(length(rawEMG)/BufSize) , nchan);


for n = 1:round(length(rawEMG)/BufSize)-(Rmswindow/BufSize)

    if n == 1
        DataI = rawEMG(1:Rmswindow,:);
    else
        DataI = rawEMG((BufSize * (n-1) + 1):(BufSize * (n-1)) + Rmswindow,:);
    end
    % FUNCTION TO IMPLEMENT BY THE STUDENT ------------------------------------
    % DataRmsEMG = T1A_ExtractRMS(DataInRawEMG)

    dOut(samplect,:) = T1A_ExtractRms(DataI);

    % TO COMPLETE BY THE STUDENT ----------------------------------------
    samplect = samplect + 1;

end

plotCh(dOut'.*10, nchan, 25)                 % Plot data and labels, 25 Hz frequency
```

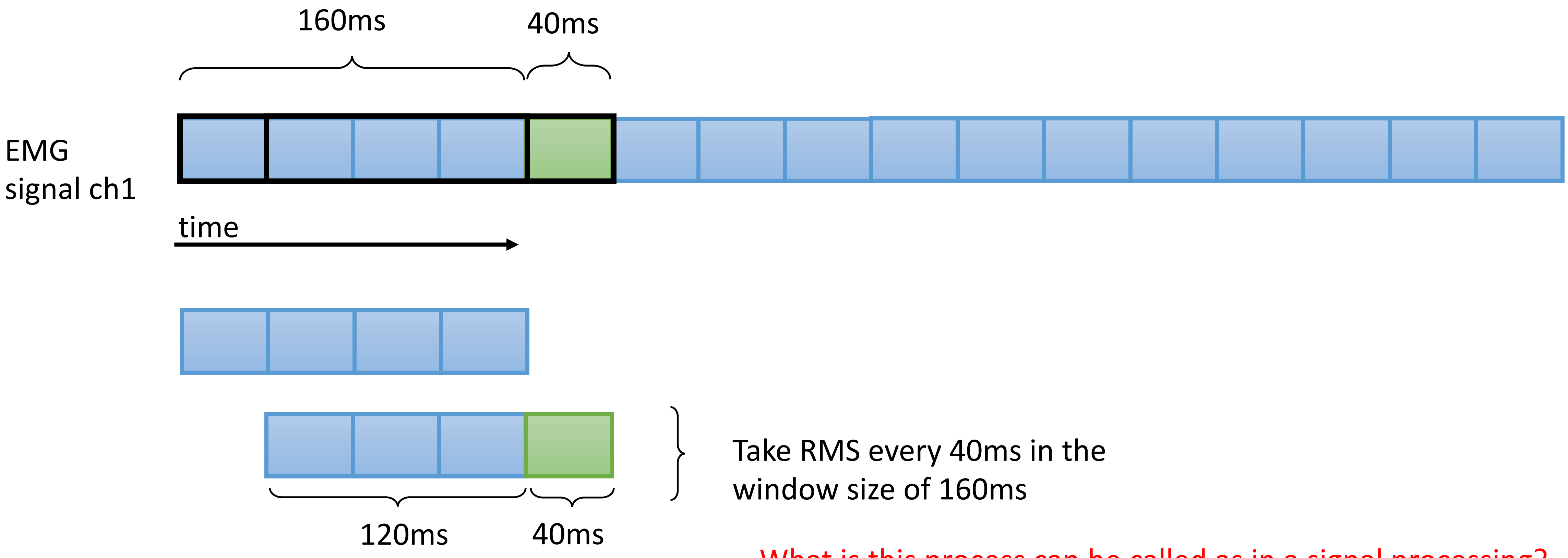
Windowing for feature extraction



Take RMS every 40ms in the window size of 160ms

What is this process can be called as in a signal processing?

Windowing for feature extraction



What is this process can be called as in a signal processing?

Low pass filter, moving average

RMS

How does the RMS formula look like?

- x_i: EMG value
- n : number of samples

- In this case n is the window size (160ms)
- We calculate the RMS for each channel
- => 1 feature for each channel at each windowed timepoint

RMS

EMG Acquisition → EMG Signal Processing → **Feature Extraction** → Trained Algorithm → Control prediction

How does the RMS formula look like?

$$RMS = \sqrt{\frac{1}{n}\sum_{i} x_i^2}$$

- x_i: EMG value
- n : number of samples

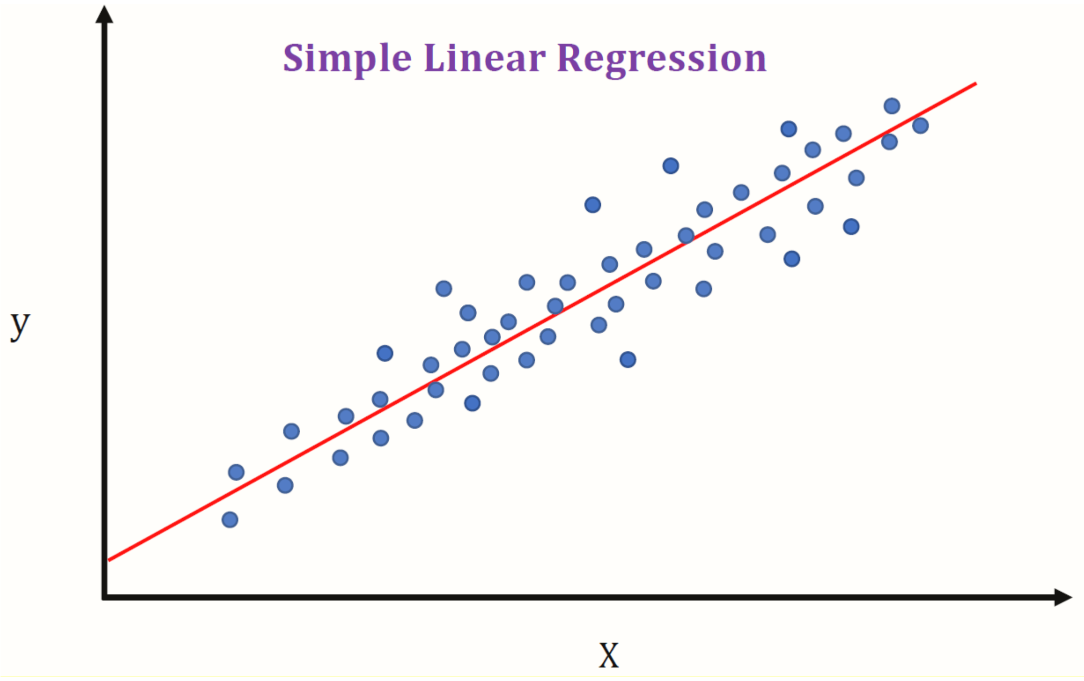
- In this case n is the window size (160ms)
- We calculate the RMS for each channel
- => 1 feature for each channel at each windowed timepoint

Linear Regression

EMG Acquisition → EMG Signal Processing → Feature Extraction → **Trained Algorithm** → Control prediction

How does the linear regression formula look like?

- y: Movement
- X : EMG feature
- m: slope / regression coefficient
- b : intercept



- Simple Linear regression – one angle at a time
- We will calculate the coefficients for each angle
- Functions in Matlab that help you extract the regression coefficients

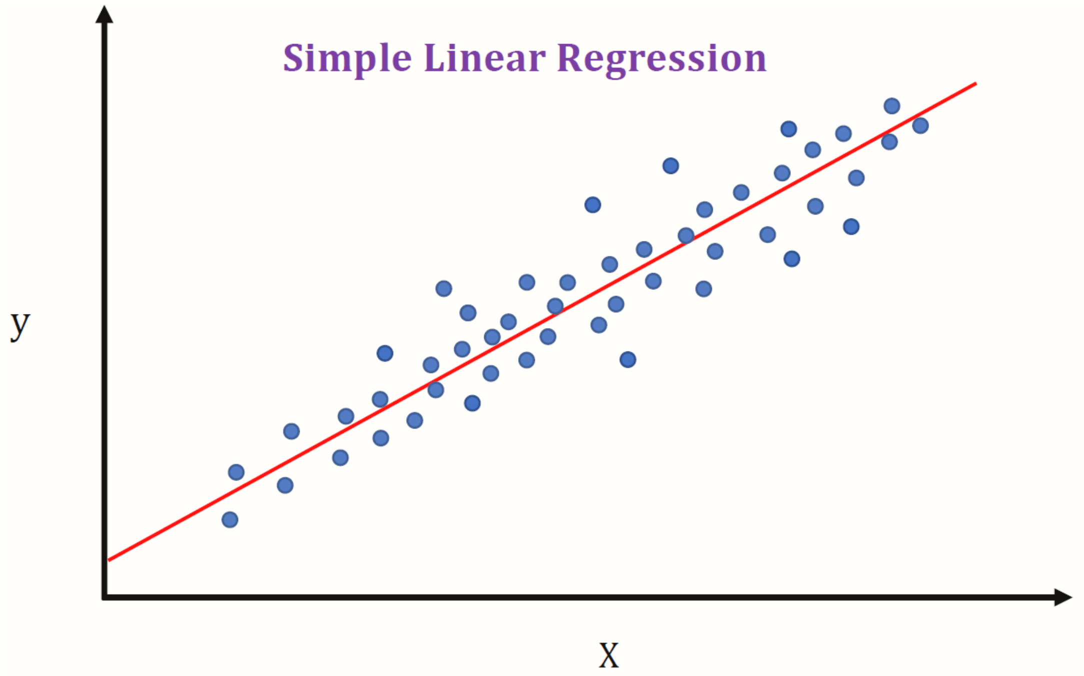
Linear Regression

EMG Acquisition → EMG Signal Processing → Feature Extraction → **Trained Algorithm** → Control prediction

How does the linear regression formula look like?

- $y = mX + b$

- y: Movement
- X : EMG feature
- m: slope / regression coefficient
- b : intercept



- Simple Linear regression – one angle at a time
- We will calculate the coefficients for each angle
- Functions in Matlab that help you extract the regression coefficients

File name: MyoAnalysis.m

3.D – Train regressor

Implement linear regression given pre-recorded data and labels – [T1B_TrainReg.m]

Task 2: Create linear regressor function – 20min

```
%% 3.D - Train regressor  --- To complete by the student
labels  = labels_aux; % [DOFs x length recording] - [2 x Length recording]
feats   = feats_aux;   % [channels x length recording]

 % FUNCTION TO IMPLEMENT BY THE STUDENT ------------------------------------
RegCoef = T1B_TrainReg(feats, labels);

```

File name: MyoAnalysis.m

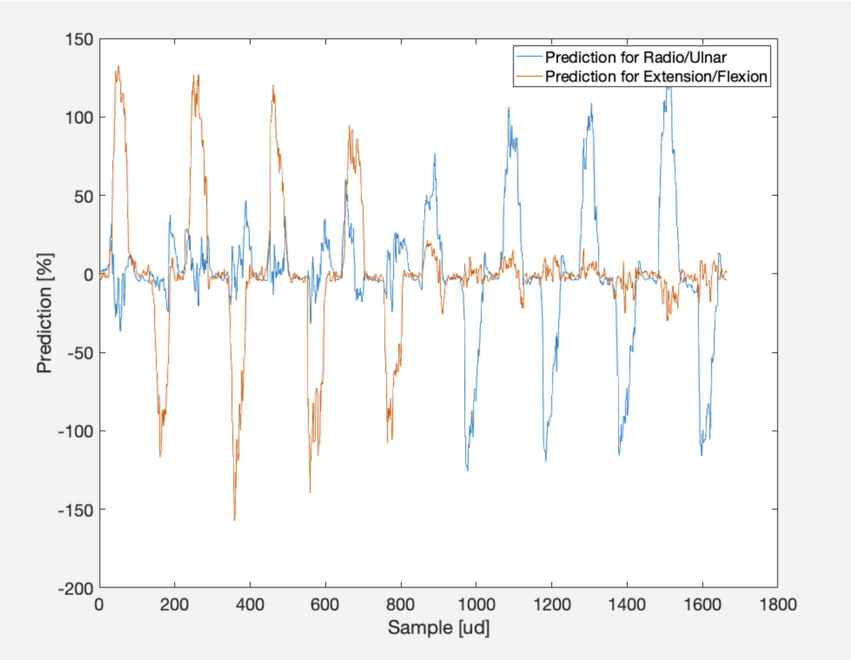
```
%% 3.E Test the regressor output

lengthRecording = length(feats);

% test the data by each sample
for sample = 1:lengthRecording
    trEst(sample,:) = T1C_TestReg([1,feats(sample,:)], RegCoef);
end

% Plot the prediction of the 2 DOF
% Positive side represents Radio/Extension prediciton
% Negative side represents Ulnar/Flexion
plot(trEst)
legend('Prediction for Radio/Ulnar','Prediction for Extension/Flexion');
set(gca,'FontSize',20);
xlabel('Sample [ud]');
ylabel('Prediction [%]');
%% 3.F - Save the regressor
save(fullfile(path,'CoefReg.mat'), 'RegCoef');

fprintf('Regressor completely saved! \n')
```

3.E – Test the regressor output

- Implement testing of the linear regression – [T1C_ TestReg.m]

Task 3: Create the testing linear regressor function – 10min

Result



- Flex/Ext and Radio/Ulnar movement
- Cross talk between two types of movements
    - The limitation of sEMG

File name: MyoAnalysis.m

3.F – Save the regressor

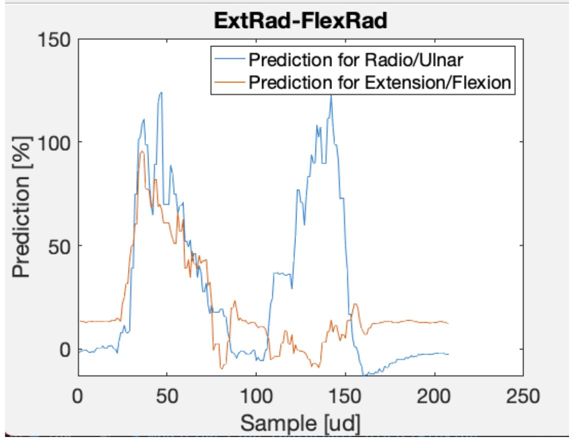
```matlab
%% 3.F - Save the regressor
save(fullfile(path,'CoefReg.mat'), 'RegCoef');

fprintf('Regressor completely saved! \n')
```

3.G – Test the regressor with new Data

- “DemoData” -> ”TestData” -> any data you want to see
- Plot prediction with the pre-recorded training data
- Plot prediction with the pre-recorded testing data
- Save regressor coefficients

```matlab
%% 3.G - Test the regressor with new Data
% Get the file
[nfile,path] = uigetfile();
file         = load(fullfile(path,nfile));

% Input data to test with the regressor
inputData     = file.RecInfo.EMGRMS;       % Rms data to test
lengthRecording = length(inputData);

% test the data by each sample
for sample = 1:lengthRecording
    OutputPred(sample,:) = T1C_TestReg([1,inputData(sample,:)], RegCoef);
end

% Plot the prediction
plot(OutputPred)
% Positive side represents Radio/Extension prediciton
% Negative side represents Ulnar/Flexion
title(nfile(3:end-4))
legend('Prediction for Radio/Ulnar','Prediction for Extension/Flexion');
set(gca,'FontSize',20);
xlabel('Sample [ud]');
ylabel('Prediction [%]');
```

Result



Real time implementation

On MyoControl.m:

1. Explain the real time system GUI – [MyoControl.m]
2. Run

   App = MyoControl

Real time on VR or prosthesis

1. Collect Data new data by student – [MyoControl.m]
2. Train regressor with student data – [MyoAnalysis.m]
3. Load regressor on real time – [MyoControl.m]
4. Test with the hand or virtual cursor