



rijksuniversiteit
 groningen



umcg

EMG FEATURES EXTRACTION AND PERFORMANCE INDICATORS FOR UPPER LIMB PROSTHETICS

Ali Huessin Ahmed Eissa

S3407470

Dept. of human movement sciences

UMCG

the Netherlands

Period: 16/04/2018 – 20/06/2018

Internship

Supervisors: A.W.Franzke, Dept. of human movement sciences (UMCG)

M.B. Kristoffersen, Dept. of human movement sciences (UMCG)

Mentor: prof. dr. ir. G.J. (Bart) Verkerke, University of Groningen, Dept. of
Biomedical engineering

Contents

Introduction	3
Methods.....	4
Original data structure.....	4
Data adjustment	5
Extraction of RAW data.....	5
Data formatting.....	7
Feature extraction.....	12
Performance	15
Statistical analysis	16
Results.....	17
Discussion and conclusion	21
References	23

Introduction

The control of upper body prosthesis through the use of myoelectric signal has been present in concept for the past 60 years [1]. For that purpose, surface electromyography (EMG) electrodes are used to record the muscle fiber action potentials of muscle bundles underneath the skin. Surface electrodes offer a non-invasive solution to EMG signal extraction, but they lack behind intramuscular electrodes, especially with regards to recordings from deep muscles and subsequent EMG crosstalk. Surface electromyography is constantly under development to overcome issues such as the displacement of the electrodes [2].

Myoelectric control of prosthesis includes many types of control methods, such as threshold, proportional and pattern recognition control. In the case of less complex controls, such as threshold control systems, the system toggles between movements, such as hand opening/closing, where the force and/or velocity of the movement is controlled by the intensity of the extracted EMG signal from surface electrodes. The type of movement is selected by contracting muscle bundles assigned to a movement, with EMG surface electrodes attached to these muscles [3].

Myoelectric pattern recognition MEPR is based on the idea that individuals are able to produce repeatable patterns of muscle signal, which gives MEPR its biggest advantage over other control methods. This means that it offers users the ability to control their prosthesis as they will [4]. However, MEPR is still not a popular choice. This is mainly attributed to the complexity of the training process for new users, which increases with added degrees of freedom, and users are mostly unable to reliably produce robust patterns [5].

The process of pattern recognition generally relies on derived values from the initially acquired data, called features. These various feature sets represent different aspects of interest in the original data, for example the color value of pixels in an image or the voltage amplitude of a signal. It is important to select relevant features to be extracted for the purpose of the pattern recognition system. These features are then used to train classifier algorithms to make predictions about the classes of new, but similar, data sets. When these new data sets are not similar to the data used to train the algorithms, the chances to have a correct prediction of the class of the data is low. [6].

After the desired features are selected, these features are extracted from segmented raw EMG data; by segmenting the data into overlapping windows, and these windows are divided into training, validation and testing sets [7].

Feature extraction can be implemented in the open-source myoelectric control platform, BioPatRec [8]. The system learns about the patterns in the data and how to classify them using

the training and validation sets, while using the testing set to check the accuracy of the system. The classification process is carried out by a classifier algorithm, and one of the advantages of using BioPatRec is that it has the option to choose from many different classifiers, or even the ability to add new ones [8].

The EU funded project INPUT [9] aims towards creating upper limb prosthesis that users can reliably utilize in their everyday activities. The function of the prosthesis is to use surface electrodes to extract EMG signal from the arm above the amputation region, and extract a predetermined set of features to train the MEPR system.

In a test carried out by members of project INPUT, EMG readings from surface electrodes were collected from 38 able bodied participants. The participants were given movement tasks to train the pattern recognition algorithm used in the project, and to test the ability of the algorithm to classify the movements from the extracted EMG signal, using the motion test [10].

In this project, with the help of the platform BioPatRec, data collected from participants can be further studied. This could be done by extracting different features, and checking the classifiers interaction with these features, as well as the possibility to use different classifiers. However, the data was collected and stored in a format that was incompatible with the required data format for BioPatRec.

Therefore, the primary goal of this study is to use Matlab to manipulate the structure of the data set used in project INPUT. This is expected to allow the use of the data on the platform BioPatRec. Second goal is to check for a possible correlation between the feature data sets and the ability of able bodied users to use the classifier algorithm. With the help of some statistical analysis, a further understanding of the effect of feature changes on performance could be acquired.

Methods

Original data structure

Project INPUT collected EMG data from 38 subjects using an 8 channel input from electrodes attached to the subjects' forearm. The first objective was to extract the raw data for each participant, with a total of 15 sessions per participant, and save it in a single file.

The original data was stored in a (.mat) file that contained three main divisions called structs. These 3 structs contained groups of participants that were divided according to the type of feedback the participants received; extended feedback (EF struct), medium feedback (MF struct) and no feedback (NF struct). Each one of these 3 structs contained fields corresponding to the participants for each group; 12 participants in the EF struct, 13 in the MF struct and 12 in the NF struct. Each participant field had 15 structs representing each session of

their participation in the trial. Finally, each one of these 15 structs contained the EMG recording session, as well as two performance indication fields; a field with the values of the number of correct classifications and another for accuracy (Fig.1).

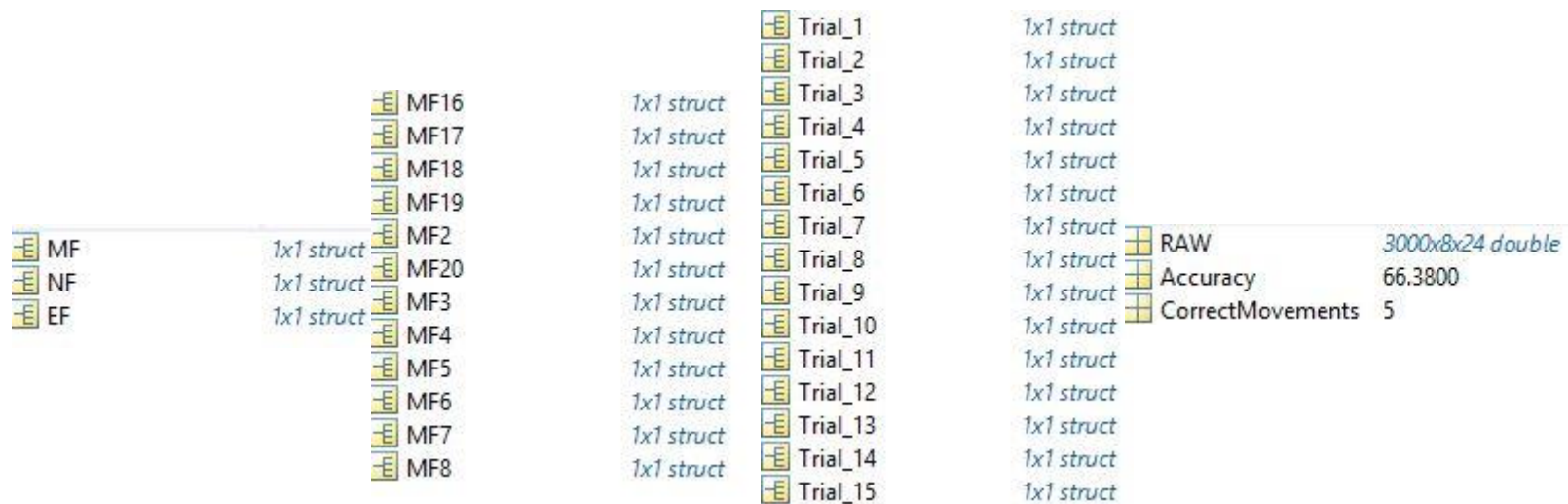


Figure 1: Original data structure

The original format for the data contained the bulk of recording session for all participants, but could not be directly imported into BioPatRec, since BioPatRec only accepts data files containing a single struct with eleven fields. The fields should have identical names to the ones used by BioPatRec, and it has the EMG recordings for a single session, as well as other fields such as the sampling frequency, the name of movements used and the number of channels.

Data adjustment

Extraction of RAW data

The first step was writing a script using Matlab, which loaded the original data file and accessed each individual raw data field, labeled it with the participant's category – MF/NF/EF- and the trial number, then saved it as a (.mat) file for further use (fig.2).

```

1 - load('ExpData.mat');
2 - names_MF = fieldnames(DataAll.MF);
3 - names_NF = fieldnames(DataAll.NF);
4 - names_EF = fieldnames(DataAll.EF);
5 - %allocating MF feedback entries
6 - for i = length(names_MF):-1:1
7 -     Current(i) = deal(DataAll.MF.(names_MF{i}));
8 -     Trial_number = fieldnames(Current(i));
9 -     for j = 1 : length(Trial_number)
10 -         k = ((i-1)*15) + j;
11 -         MF_entries(k) = deal(Current(i).(Trial_number{j}));
12 -     %buffer
13 -         x = MF_entries(k);
14 -         save(sprintf('MF_entries_%02d',k), 'x');
15 -     end
16 - end
17 - %allocating NF feedback entries
18 - for i = length(names_NF):-1:1
19 -     Current_2(i) = deal(DataAll.NF.(names_NF{i}));
20 -     Trial_number = fieldnames(Current_2(i));
21 -     for j = 1 : length(Trial_number)
22 -         k = ((i-1)*15) + j;
23 -         NF_entries(k) = deal(Current_2(i).(Trial_number{j}));
24 -     %buffer
25 -         x = NF_entries(k);
26 -         save(sprintf('NF_entries_%02d',k), 'x');
27 -     end
28 - end
29 - %allocating EF feedback entries
30 - for i = length(names_EF):-1:1
31 -     Current_3(i) = deal(DataAll.EF.(names_EF{i}));
32 -     Trial_number = fieldnames(Current_3(i));
33 -     for j = 1 : length(Trial_number)
34 -         k = ((i-1)*15) + j;
35 -         EF_entries(k) = deal(Current_3(i).(Trial_number{j}));
36 -     %buffer
37 -         x = EF_entries(k);
38 -         save(sprintf('EF_entries_%02d',k), 'x');
39 -     end
40 - end

```

Figure 2: Data extraction code

A shorter version of the script was originally used, which used a single loop for all three participant types – MF/EF/NF-, but was later expanded into three distinct parts to debug possible issues. One of the issues encountered was that there was a recording session missing for one of the participants, which initially caused a labeling error for the saved files. The issue

was mitigated by using the number of structs for each participant as a variable that can be used for numbering in the formula $[k = ((i-1)*15) + j]$. The saved files had a title containing the type of the participant –MF/EF/NF- and the numbering for the trials for each type from 1 to the total number of trials of the same type. For example, the fifth EF Participant would have their 10th trial's raw data saved as (EF_entries_50.mat). This formatting allowed for using simpler counters in further parts of the data processing, and could easily be recovered into the original trial numbering from 1 to 15.

Data formatting

The format that needed to be changed was the format of the EMG [RAW] data within each recording session, which is why the RAW data was extracted from the original data file into individual files. The recordings in the original data were divided into 21 movements for each trial, where 7 movements were used to create the 21 sets; such that each movement is made into sets of 3 consecutive 3 second contractions with 30%, 60% and 90% contraction power respectively. Rest was treated as a separate extra movement. The original RAW data files contained a 3 dimensional matrix, where one dimension was for each of the 8 channels, one dimension represented one of the 24 movements (3 repetitions of 7 movements + 3 rest) and one dimension for the raw data for each movement for each channel.

For the data to be imported into BioPatRec, recordings had to be in a specific format of 3 repetitions of 3 seconds of contraction followed by 3 seconds of rest for each movement. Which means that each one of the 21 movements (as categorized in the original data) needs to have its contractions extracted individually, and then having its corresponding rest data added between each 3 second contraction of the same movement type, such that the data would be in the form: [3 seconds 30% contraction (movement (x)) → 3 seconds rest → 3 seconds 60% contraction (movement(x)) → 3 seconds rest → 3 seconds 90% contraction (movement(x)) → 3 seconds rest. There was only 1 set of rest data in the original recordings consisting of 9 seconds of rest, and it was used to fill in the rest data for all movements. This was done after making sure that BioPatRec wouldn't be using the rest intervals in the feature extraction, and is exclusively for entering the data in an acceptable format, as they are cut out by BioPatRec in the phase where the contractions are extracted from the data file.

Using MATLAB, 555 individual RAW data files were created in the extraction of RAW data step, containing the individual recordings for each participant for each day. These data files were read into the data formatting script (fig.3), which put the data in the correct format, and created a struct containing the data. The script also created the other fields required by BioPatRec that define the number of channels and the name of movements, as well as other fields that are used in the feature extraction phase (fig.4).


```

1 - name_counter = 1;
2 - Number = 195;
3 - for n = 1:Number
4 -     Trial_num = rem(n,15);
5 -     if Trial_num == 0
6 -         Trial_num = 15;
7 -     end
8 -     if Trial_num == 1
9 -         name_counter = (name_counter + 1) ;
10 -    end
11
12 -    x = load(sprintf('NF_entries%d.mat', n));
13 -    % main loop through 8 channels
14
15
16 -    for r = 8:-1:1
17
18 -        % loop through the movement types (21 in total) with a step size of 3, include 30%,
19 -        % 60% and 90% for each iteration
20 -        for i = 8:-1:1
21 -            k1 = ((3*(i-1))+1);
22 -            %extract the 3 data sets for each movement
23 -            for j = k1+2:-1:k1
24 -                movement(:,j) = x.x.RAW(:,r,j);
25 -                k2 = ((k1+3)-j);
26 -                rest(:,j) = x.x.RAW(:,r,k2);
27 -                %create a small block containing a contraction plus rest
28 -                block_1(:,j) = vertcat(movement(:,j),rest(:,j));
29 -                %if all 3 contraction forces are conc. to their respective rest
30 -                %data, create a bigger block containing all 3 contractions and
31 -                %rest
32 -                if j == k1
33 -                    block_3(:,r,i) = vertcat(block_1(:,j), block_1(:,j+1), block_1(:,j+2));
34 -                end
35 -            end
36 -        end
37 -    end
38 -    tdata = block_3(:, :, 2:8);
39 -    date = [2016 12 15 12 08 20];
40 -    dev = char('13E200');
41 -    temp_field = names_NF(name_counter -1);
42 -    %create the modified data
43 -    recSession = struct('sF',1000,'sT',18,'cT',3,'rT',3,'nM',7,'nR',3,'nCh',8, 'mov',{'Pronation'; 'Supinatio
44 -    save(sprintf('adjusted%s_trial%d.mat',temp_field, Trial_num ),'recSession');
45 -    end

```

Figure 3: Data formatting code

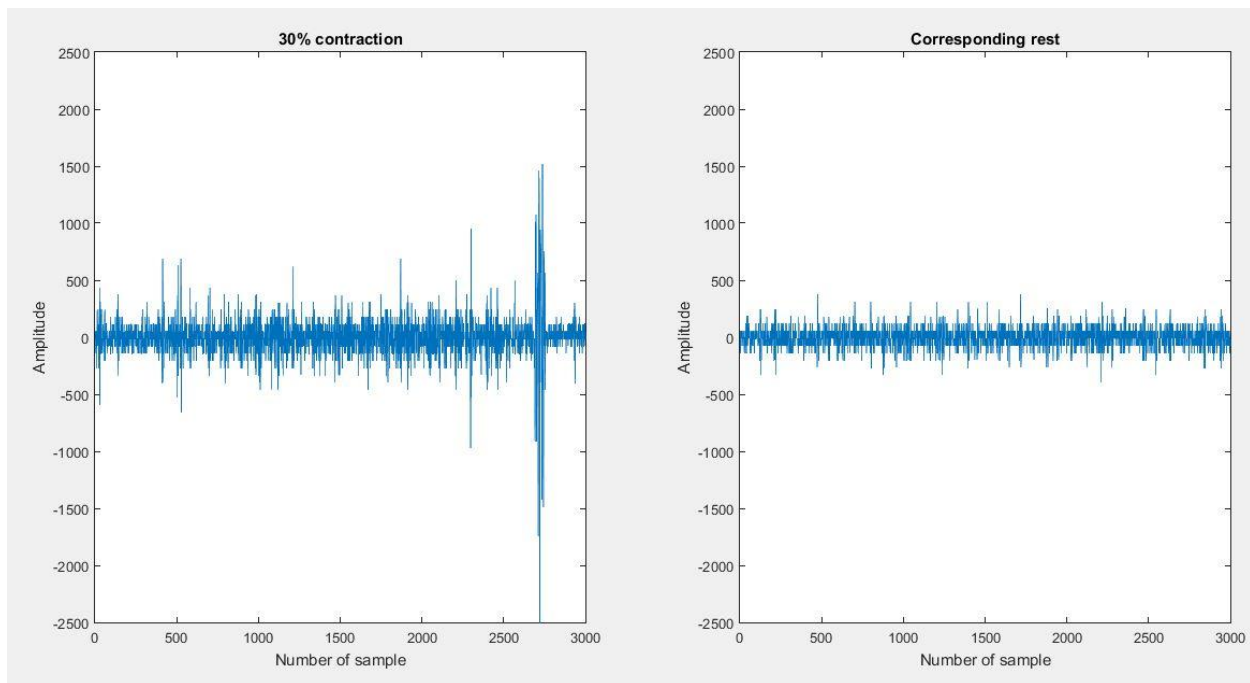
1x1 struct with 17 fields

Field ▲	Value
sF	1000
tW	0.2000
nCh	[1,2,3,4,5,6,7,8]
mov	8x1 cell
scaled	NaN
noise	0
wOverlap	0.0500
dev	'13E200'
comm	'N/A'
fFilter	1x1 cell
sFilter	1x1 cell
trSets	48
vSets	24
tSets	49
trFeatures	48x8 struct
vFeatures	24x8 struct
tFeatures	49x8 struct

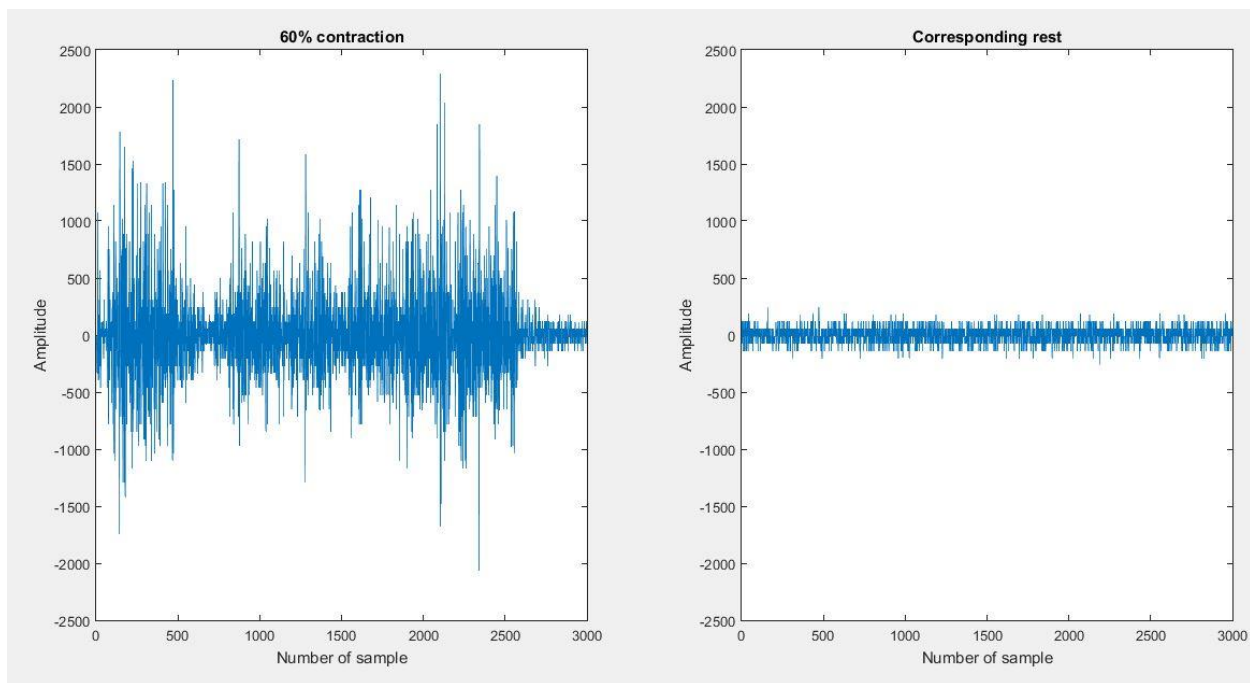
Figure 4: Format of adjusted data

The script for this phase was initially made to loop through all 3 types of data (NF/MF/EF), then a single script for each data type was used instead to simplify the code and it could be easily adjusted to all three types. The script shown in figure (3) was used for (NF) titled part of the data, and the concept for data formatting is the same for the other two types. With knowledge of the sampling frequency of the original data, and the arrangement of the movement types, the code starts looping through each of the 8 channels from the recording while taking in consideration the size of each contraction. The code saves the rest data aside, separates the first contraction power for the first movement and concatenates it with the corresponding part of the rest data. Then the code does the same for the second and third contraction parts of the same movement, and saves the created block of data which contains the consecutive contractions and rest. The code then proceeds to the next movement and repeats the process, and does so for all the movements, before proceeding to the next channel, until all the data is in the desired format. Afterwards, the script creates the fields shown in figure (4), and saves the data in a new file with the number of the trial.

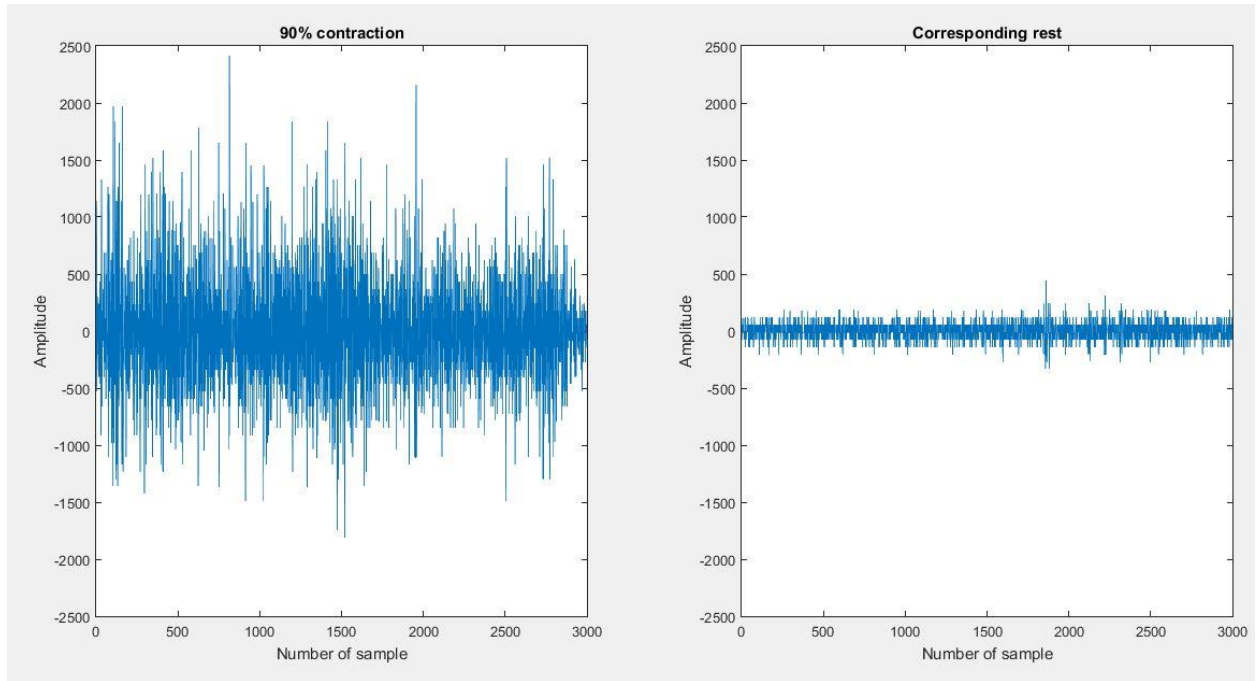
A visualization of how the data looked like before and after adjustment can be noted by plotting the data (fig. 5). The figure illustrates how a single iteration of the code concatenates a movement to its corresponding rest data.



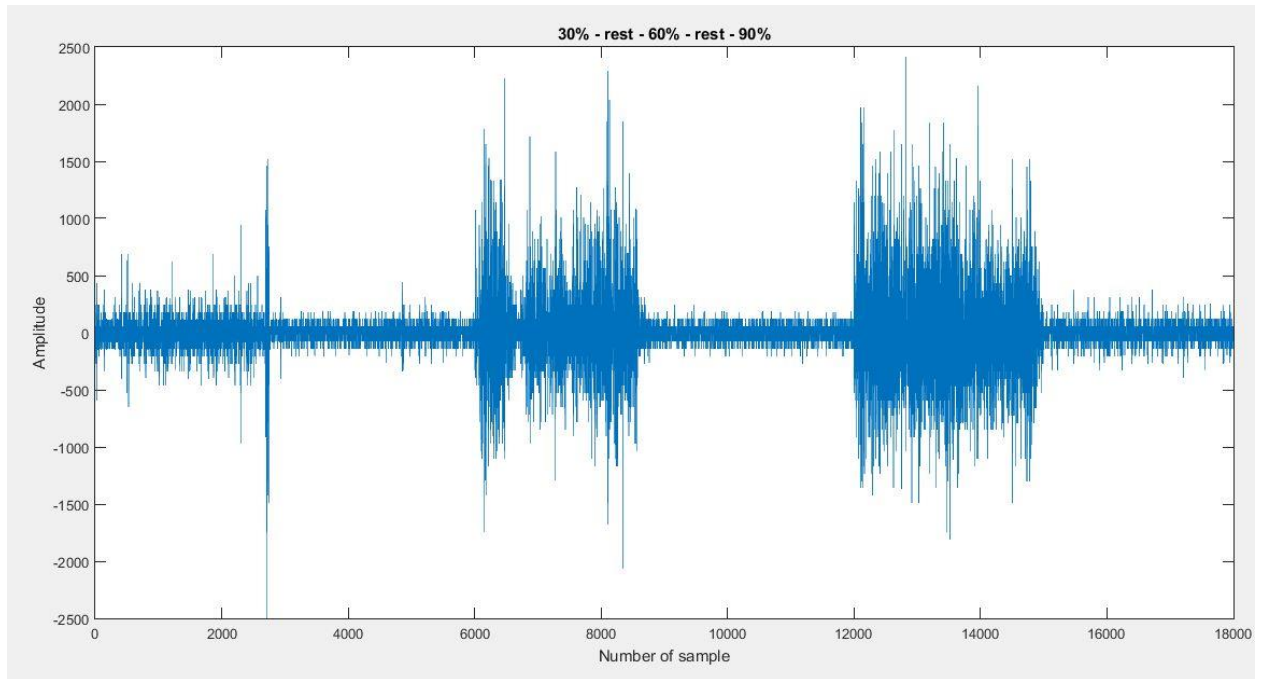
(a)



(b)



(c)



(d)

Figure 5: Extraction of the data of individual contraction powers and corresponding rest from the original format (a,b and c).
Adjusted data of contraction powers and corresponding rest, making 1 move (d)

As a last adjustment before using the data, the file naming was changed to include the original naming of the data from project INPUT (fig. 6).

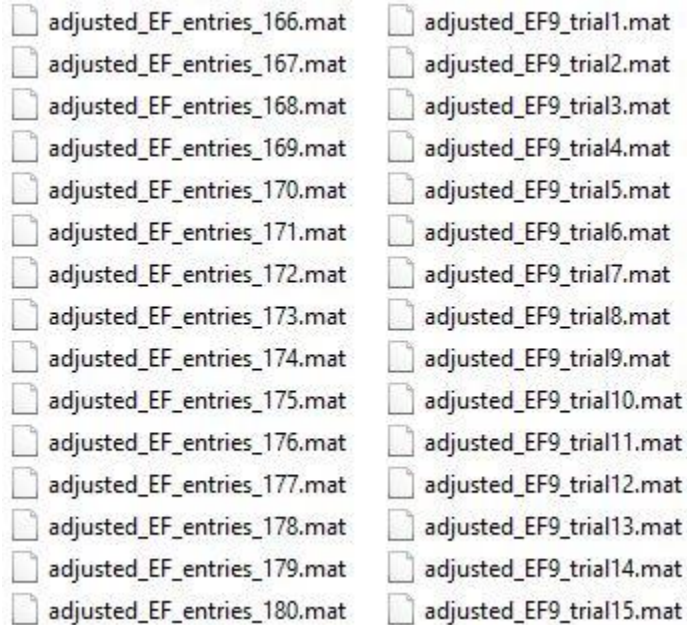


Figure 5: Saved data and naming; Initial naming (left) and new naming (right)

Feature extraction

Four features were chosen to be extracted from the signal, according to their prominent use in literature, as a non-exhaustive compilation of myoelectric signal features employed in pattern recognition for prosthetic control. These features are: Absolute mean value, zero crossing, wave length and root mean square [8].

The feature absolute mean value is referred to as (tmabs) in BioPatRec, and it is calculated from the mean of the sum of the absolute data values in the segmented window. The root mean square is referred to as (trms), and is calculated from the square root of the sum of the squared values divided by the number of data points in the window. The zero crossing feature referred to as (tzc) represent points in the data where the value changes from positive to negative, or vice versa. The calculation of the wavelength feature, as an accumulative change in the length between consecutive points in the same phase, is referred to as (twl). The calculation of all four features is carried out within the (GetSigFeatures) function in BioPatRec through formulas dedicated to different feature types (fig.7).

```

85 function pF = GetSigFeatures_tmabs(pF)
86 % 2011-07-27 Max Ortiz / Creation
87 -   pF.f.tmabs = mean(pF.absdata);
88 - end

122 function pF = GetSigFeatures_trms(pF)
123 % 2011-07-27 Max Ortiz / Creation
124 -   pF.f.trms = sqrt(sum(pF.absdata.^2)/pF.sp);
125 - end

113 function pF = GetSigFeatures_twl(pF)
114 % Waveform Length (acumulative changes in the length)
115 % 2011-07-27 Max Ortiz / Creation
116 -   mdata = [zeros(1,pF.ch) ; pF.data(1:pF.sp-1,:)];
117 -   pF.f.twl = sum(abs(pF.data - mdata));
118 - end

129 function pF = GetSigFeatures_tzc(pF)
130 % 2011-07-27 Max Ortiz / Creation
131 % check if tmabs is available
132 - if ~isfield(pF.f,'tmabs')
133 -   pF = GetSigFeatures_tmabs(pF);
134 - end
135
136 - tmp = repmat(pF.f.tmabs,[size(pF.data,1),1]);
137 - zc = (pF.data >= tmp) - (pF.data < tmp);
138 - pF.f.tzc = sum( (zc(1:pF.sp-1,:) - zc(2:pF.sp,:)) ~= 0);
139
140 % % Zero Crossing / using the abs mean as threshold
141 % for i = 1 : pF.ch
142 %   zc = (pF.data(:,i) >= pF.f.tmabs(i)) - (pF.data(:,i) < pF.f.tmabs(i));
143 %   pF.f.tzc(i) = sum((zc(1:pF.sp-1) - zc(2:pF.sp)) ~= 0);
144 % end
145
146 - end

```

Figure 6: Mean absolute average (Lines 86-88), root mean square (lines 122-125), wave length (lines 113-118) and zero crossing (lines 129-146)

Since the feature data is extracted for every one of the 8 channels used for recording, and divided into training, validation and testing sets, the extracted features needed to be laid out in proper format to collect all the fragments in one data set. Through the combined use of BioPatRec and Matlab, the feature sets of each participant could be calculated. The script could continuously interact with the features produced by BioPatRec and transform them into the desired format (fig.8).


```

1 - load('Channel_test_1.mat');|
2 - %acquire structs for the training, validation and testing
3 - tr = deal(sigFeatures.trFeatures);
4 - v = deal(sigFeatures.vFeatures);
5 - te = deal(sigFeatures.tFeatures);
6 - %initializing feature sets
7 - tr_mabs = zeros((8*48),8);
8 - tr_zc = zeros((8*48),8);
9 - tr_wl = zeros((8*48),8);
10 - tr_rms = zeros((8*48),8);
11 - v_mabs = zeros((8*24),8);
12 - v_zc = zeros((8*24),8);
13 - v_wl = zeros((8*24),8);
14 - v_rms = zeros((8*24),8);
15 - te_mabs = zeros((8*49),8);
16 - te_zc = zeros((8*49),8);
17 - te_wl = zeros((8*49),8);
18 - te_rms = zeros((8*49),8);
19 - %create a step count to fill in the data: 8 data points per step for each
20 - %of the 8 channels starting with the training set
21 - for j = 1:8
22 -     for i = 1:48
23 -         x = (i*8);
24 -         y = (x - 7);
25 -         tr_mabs((y:x),j) = tr(i,j).tmabs;
26 -         tr_zc((y:x),j) = tr(i,j).tzc;
27 -         tr_wl((y:x),j) = tr(i,j).twl;
28 -         tr_rms((y:x),j) = tr(i,j).trms;
29 -     end
30 - end
31 - % Validation set
32 - for j = 1:8
33 -     for i = 1:24
34 -         x = (i*8);
35 -         y = (x - 7);
36 -         v_mabs((y:x),j) = v(i,j).tmabs;
37 -         v_zc((y:x),j) = v(i,j).tzc;
38 -         v_wl((y:x),j) = v(i,j).twl;
39 -         v_rms((y:x),j) = v(i,j).trms;
40 -     end
41 - end
42 - %test set

```

```

43 - for j = 1:8
44 -     for i = 1:49
45 -         x = (i*8);
46 -         y = (x - 7);
47 -         te_mabs((y:x),j) = te(i,j).tmabs;
48 -         te_zc((y:x),j) = te(i,j).tzc;
49 -         te_wl((y:x),j) = te(i,j).twl;
50 -         te_rms((y:x),j) = te(i,j).trms;
51 -     end
52 - end
53 - %concatenate values from all the sets
54 - mabs = cat(1,tr_mabs,v_mabs,te_mabs);
55 - zc = cat(1,tr_zc,v_zc,te_zc);
56 - wl = cat(1,tr_wl,v_wl,te_wl);
57 - rms = cat(1,tr_rms,v_rms,te_rms);
58 - save('Channel_test_features_1','mabs','zc','wl','rms');

```

Figure 7: Feature extraction and formatting code

Biopatrec divides the extracted features into training, validation and testing sets. All three sets are required in order to construct an array of all features for each channel. The code extracts the four specified features and concatenates all three types of feature sets, and then it saves the features with the corresponding trial number. Another script was used to collect all features extracted from all 15 trials for each participant that simply puts the extracted features from individual trials together in one field.

Performance

The original data collected from participants included performance indicators. A classification was made using Matlab, according to the overall performance enhancement throughout the trial, and the participants that showed the most improvement were the ones further studied.

The goal was to select participants that showed positive and consistent change in their performance throughout the trials. The classification was carried out by plotting both accuracy and number of correct movements for all 15 sessions of each participant. The data in the performance plots were analyzed by calculating the number of positive changes between consecutive points in the graph, and the value of change. Negative change contributed negatively to the performance enhancement according to how much the accuracy or number of correct movements dropped (fig.9).

The decision on the participant that showed most improvement was made by choosing the participant with the highest change value; such that the participant would show improvement

in performance from the initial trial, and consistently kept the performance high with little negative retrace.

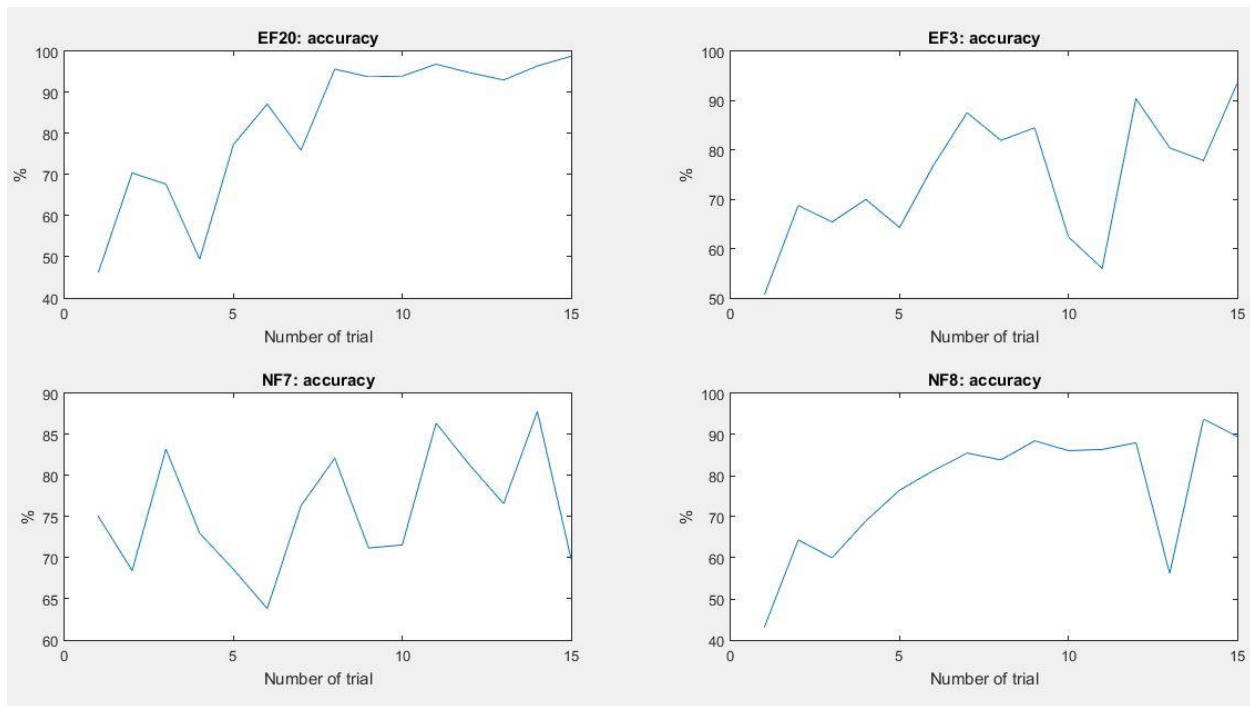


Figure 8: Performance plots of some participants

Performance tracking by the INPUT team in the original data was indicated by two fields: accuracy and number of correct movements. Accuracy is a measure of the percentage of the time the performed movement matches the prompted movement for the participant, and the number correct movements are the number of movements that were performed without misclassifications for two seconds. A profile containing the 4 chosen features was created, containing data for the participant that showed the most enhancements in performance throughout the 15 trials, which is participant EF_20. Using this profile, the feature data could be put up against the performance indication data set for that participant, and further statistical study of a possible systematic change in the features was possible.

Statistical analysis

In order to study a possible systematic change in the feature data sets corresponding to performance indicators for participants, four spearman correlation tests were carried out. The aim of carrying out multiple tests was to test the correlation coefficient value between equally high, equally low and between high and low performance trials from the same participant, as well as between equally high performance trials from different participants. These four tests were made to check all possible correlations between performance values. Spearman

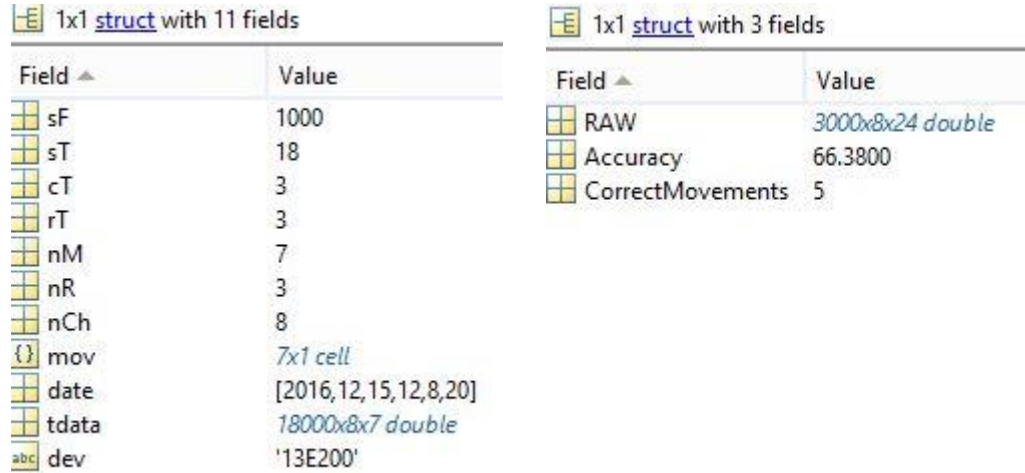
correlation was used instead of pearson correlation, since the data is non-linear [11].

The feature data from participant EF_20 was used for the first three tests, which are those carried out on the data from the same participant: Test number one was between two high accuracy trials, trial 9 (accuracy: 93.8%) and trial 10 (accuracy: 94%). The second test was between two low accuracy trials, trial 1 (accuracy: 46.24%) and trial 4 (accuracy: 49.4%). The third test was between one low and one high accuracy trial, trial 1 (accuracy: 46.24%) and trial 15 (accuracy: 98.84%).

The fourth correlation test, which was between equally high performance trials from different participants, was carried out on the extracted features from participant EF_20 – trial 9 (accuracy: 93.8%), and participant EF_9 – trial 11 (accuracy 92.7%).

Results

The first part of the study, which aims to extract and change the format of the data from project INPUT into the format used by BioPatRec, was successful for all recording sessions (Fig.10), and the data could be loaded into BioPatRec to extract the feature data sets.



Field	Value
sF	1000
sT	18
cT	3
rT	3
nM	7
nR	3
nCh	8
mov	7x1 cell
date	[2016,12,15,12,8,20]
tdata	18000x8x7 double
dev	'13E200'

Field	Value
RAW	3000x8x24 double
Accuracy	66.3800
CorrectMovements	5

Figure 10: Adjusted data (left) and original data (right)

For the selection of the participant showing the most prominent improvement in performance, the performance indication in the form of a plot for both accuracy and number of correct movement was created with Matlab, which allowed to visually and numerically check for the participant with the highest performance improvement. This helped selecting participant (EF_20) as the most prominent enhancement in performance (Fig. 11).

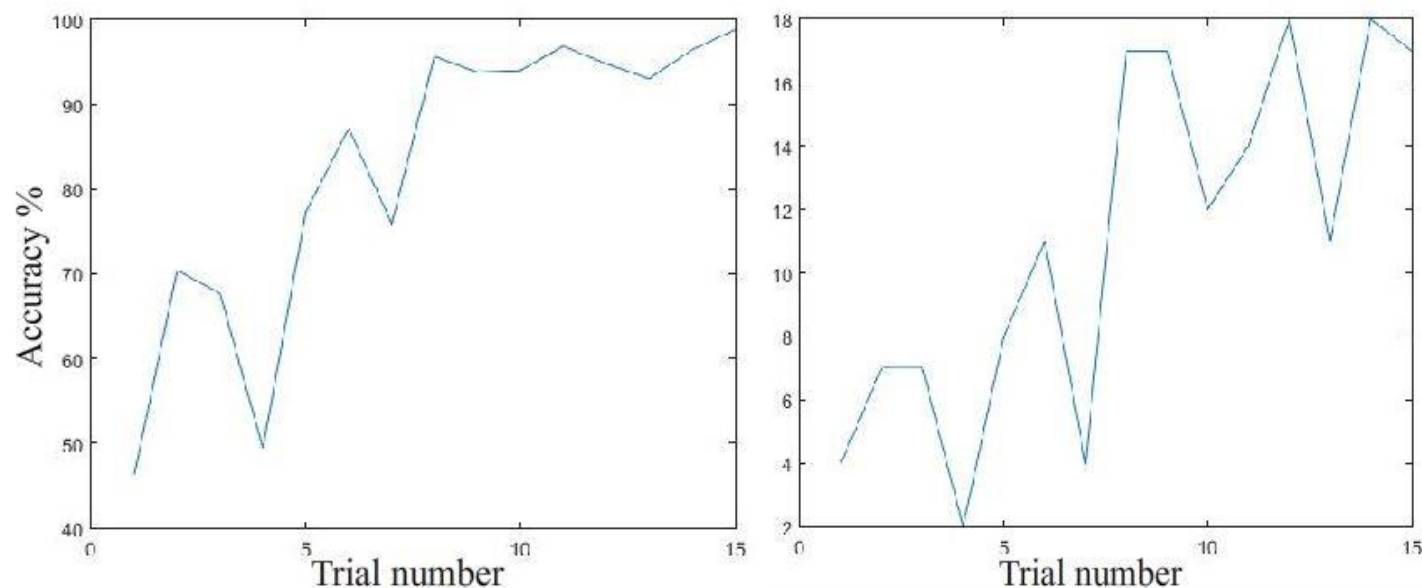


Figure 11: Performance of participant EF_20 - accuracy (left) and number of correct movements (right)

In terms of feature extraction, the four chosen features were extracted and plotted for each of the 8 channels. For each participant, the full feature points collected throughout their 15 trials could be presented both numerically (as 15,000 points) and graphically, as show in (Fig. 12).

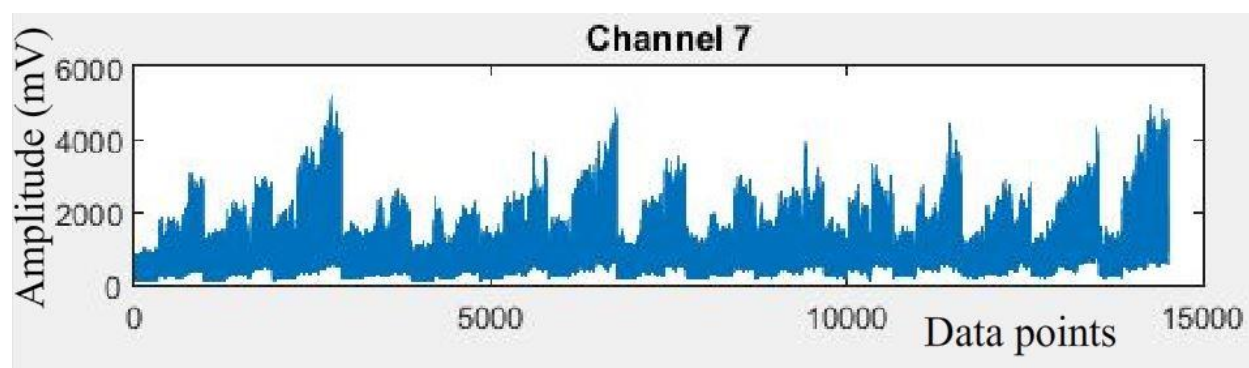


Figure 12: (Mabs features for participant EF_20) - channel 7

As for the statistical study of a possible correlation between the feature data points, with regards to the achieved accuracy in trials, Graphpad Prism was used; pearson correlation test was performed using GraphPad Prism version 7.04 for Windows [12]. Where the correlation coefficient (r) has a value between $[-1,1]$, where an r value of (-1) means that the data sets are completely inversely correlated, (0) means that there is no correlation between the data, and (1) means that the data is completely correlated. The tests were carried out on the data from all four extracted features, and for each of the 8 channels.

Color coding:

R value	0.85→1	0.75→0.85	0.65→0.75	0.5→0.65	0.1→0.5	<0.1
color						

Each row in the tables represents the correlation (r) value between two different trials; where the number of the channel in comparison is mentioned in the first row, while the title of the compared feature is mentioned in the first column.

R values between high accuracy trials for participant EF_20, between [trial 9 (93.8%) and trial 10 (94%)]:

	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
Mabs	0.919 (0.9083 to 0.9286)	0.8727 (0.8563 to 0.8873)	0.9569 (0.951 to 0.962)	0.9007 (0.8877 to 0.9122)	0.8945 (0.8807 to 0.9068)	0.7903 (0.7646 to 0.8134)	0.8613 (0.8436 to 0.8772)	No data
ZC	0.5929 (0.5492 to 0.6334)	0.281 (0.2202 to 0.3397)	0.1336 (0.06936 to 0.1968)	0.4076 (0.352 to 0.4603)	0.3798 (0.3228 to 0.4339)	0.2311 (0.1687 to 0.2916)	0.2037 (0.1407 to 0.2651)	No data
WL	0.9091 (0.8971 to 0.9197)	0.8733 (0.857 to 0.8879)	0.9628 (0.9578 to 0.9673)	0.9048 (0.8922 to 0.9159)	0.9007 (0.8876 to 0.9122)	0.7915 (0.766 to 0.8146)	0.8613 (0.8435 to 0.8771)	No data
RMS	0.9207 (0.9102 to 0.9301)	0.8698 (0.8531 to 0.8848)	0.9592 (0.9537 to 0.9641)	0.8955 (0.8818 to 0.9076)	0.8908 (0.8766 to 0.9035)	0.785 (0.7588 to 0.8087)	0.8426 (0.8227 to 0.8604)	No data

R values between low accuracy trials for participant EF_20, between [trial 1 (46.24%) and trial 4 (49.4%)]:

	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
Mabs	0.5021 (0.452 to 0.5491)	0.3979 (0.3418 to 0.4511)	0.5441 (0.4967 to 0.5882)	0.8997 (0.8866 to 0.9114)	0.7768 (0.7497 to 0.8013)	0.7451 (0.7148 to 0.7727)	0.7426 (0.712 to 0.7704)	No data
ZC	0.1539 (0.0899 to 0.2166)	0.1325 (0.06825 to 0.1957)	0.3682 (0.3108 to 0.423)	0.2422 (0.1802 to 0.3023)	0.2956 (0.2352 to 0.3536)	0.07387 (0.009048 to 0.1381)	0.1821 (0.1187 to 0.2441)	No data
WL	0.5784 (0.5335 to 0.62)	0.4534 (0.4003 to 0.5035)	0.5026 (0.4525 to 0.5496)	0.911 (0.8993 to 0.9214)	0.803 (0.7787 to 0.8249)	0.775 (0.7478 to 0.7997)	0.771 (0.7433 to 0.796)	No data
RMS	0.4797 (0.4282 to 0.5282)	0.4035 (0.3477 to 0.4564)	0.5539 (0.5072 to 0.5973)	0.9014 (0.8885 to 0.9129)	0.7622 (0.7336 to 0.7881)	0.7386 (0.7077 to 0.7667)	0.7369 (0.7058 to 0.7652)	No data

R values between high and low accuracy trials for participant EF_20, between [trial 1 (46.24%) and trial 15 (98.84%)]:

	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
Mabs	0.5483 (0.5012 to 0.5921)	-0.05602 (-0.1204 to 0.00888)	0.6615 (0.6234 to 0.6965)	0.862 (0.8444 to 0.8778)	0.4083 (0.3527 to 0.4609)	0.7313 (0.6996 to 0.7601)	0.7898 (0.7641 to 0.813)	No data
ZC	0.05937 (-0.005517 to 0.1238)	0.1178 (0.05338 to 0.1813)	0.09138 (0.02667 to 0.1553)	0.04523 (-0.01969 to 0.1098)	-0.1809 (-0.2429 to -0.1174)	0.1556 (0.0917 to 0.2183)	0.07174 (0.0069 to 0.136)	No data
WL	0.5542 (0.5076 to 0.5976)	-0.05703 (-0.1215 to 0.007864)	0.6429 (0.6032 to 0.6794)	0.8674 (0.8504 to 0.8826)	0.3717 (0.3144 to 0.4263)	0.7501 (0.7203 to 0.7771)	0.817 (0.7942 to 0.8375)	No data
RMS	0.5326 (0.4845 to 0.5807)	-0.06572 (-0.13 to -0.0008543)	0.6637 (0.6258 to 0.7016)	0.8606 (0.8428 to 0.8784)	0.3877 (0.3312 to 0.4442)	0.7711 (0.7434 to 0.7988)	0.7798 (0.7531 to 0.8065)	No data

	0.5775)		0.6985)	0.8765)	0.4415)	0.7961)	0.804)	
--	---------	--	---------	---------	---------	---------	--------	--

R values for high accuracy trials between participant (EF_20) - trial 9 (93.8%), and participant (EF_9) - trial 11 (92.7 %)]:

	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
Mabs	0.7517 (0.722 to 0.7786)	0.5023 (0.4521 to 0.5492)	0.8843 (0.8693 to 0.8976)	0.8814 (0.8661 to 0.8951)	0.6149 (0.5729 to 0.6537)	0.8658 (0.8486 to 0.8811)	0.819 (0.7964 to 0.8393)	No data
ZC	0.236 (0.1738 to 0.2963)	0.1483 (0.0842 to 0.2111)	0.1865 (0.1231 to 0.2483)	0.03605 (- 0.02888 to 0.1007)	0.04676 (- 0.01816 to 0.1113)	0.1288 (0.06449 to 0.1921)	0.01297 (- 0.05194 to 0.07777)	No data
WL	0.7744 (0.7471 to 0.7992)	0.5253 (0.4766 to 0.5707)	0.8855 (0.8706 to 0.8987)	0.9016 (0.8887 to 0.9131)	0.6526 (0.6137 to 0.6883)	0.8688 (0.852 to 0.8839)	0.8026 (0.7782 to 0.8245)	No data
RMS	0.7501 (0.7203 to 0.7771)	0.485 (0.4338 to 0.5331)	0.8824 (0.8672 to 0.896)	0.8753 (0.8592 to 0.8896)	0.6187 (0.577 to 0.6572)	0.8648 (0.8475 to 0.8803)	0.8121 (0.7888 to 0.8331)	No data

Discussion and conclusion

The extraction and reformatting of data from project INPUT allowed for the data from project INPUT to be used in the pattern recognition platform BioPatRec. This opened the opportunity to further investigate the EMG signal collected from the participants in the project, and the ability to study extracted features from the signal.

Many scripts were designed using Matlab, most of which are explained in this paper. The use of Matlab and the implementation of basic logic and programming offered a simple and effective approach to transform the data into the desired format, and to interact with the pattern recognition platform.

The statistical analysis of the extracted features was carried out using Graphpad prism, after choosing a participant from project INPUT that exhibited the most noticeable enhancement in performance, in both accuracy and number of correctly classified movements. The statistical

test carried out was planned to give an idea about the possible correlation between feature data sets. The tests were carried out between trials of similar accuracy values, both high and low, and between the highest and lowest accuracy values. This was done to test all possible scenarios in a non-exhaustive manner, in order to test the hypothesis about the existence of a correlation.

The first test was between two trials from the same participant, who had a high accuracy on both chosen trials. The result of the correlation test shows a very high correlation between the trials for the Mabs, WL and RMS features. This high correlation might point to a systematic change in the feature data corresponding to the high accuracy achieved in the trials. The second test was for the same participant, but between two low accuracy trials. The result of the correlation test shows poor correlation between the trials, which points to the low repeatability of the feature data. The third test for the same participant was between a low accuracy trial and a high accuracy trial. The results also show a poor correlation between the trials, which supports the idea that the accuracy of classification is reflected in the feature data, such that different spectra of accuracy would exhibit differently correlated feature data, and only highly accurate trials would exhibit a high correlation. The last test was between two trials of similar accuracy, but for two different participants. The idea of this last test was to check if different participants exhibit a specific level of repeatability in the feature space that can be traced via ranking correlation. The result of the test showed a moderately strong correlation between the two trials, which might point to a systematic change in the feature space that reflects the performance of participants.

Throughout the tests, all 8 channels of the data were used, but it appears that the data collected from channel 8 was of much lower amplitude than the rest, and was considered an outlier since it offered an almost constant/non-varying read out values. However, the 7 other channels were all considered, some of which consistently offered a higher correlation than others as seen in channel (4). Also, the nature of the chosen features can offer a higher correlation based on the algorithms used in acquiring them, which is why the color coding was used to determine variability in correlation instead of using (r) values as they are conventionally used. It can also be noticed that the zero crossing feature offered very little contribution to the correlation on the scale, but variations can be noticed nonetheless between strongly and poorly correlated trials.

A first conclusion from these results is that there seems to be a systematic change in the feature data that reflects the performance of participants. However, the signal acquisition and processing contributes to a high extent in such a correlation. It is also important to point out that the correlation tests were carried out on only two participants. Therefore, a complete analysis for other random participants will be necessary to have a more concrete resolution about the existence of a correlation.

References

- [1] Parker, P., Englehart, K. and Hudgins, B. (2006). Myoelectric signal processing for control of powered limb prostheses. *Journal of Electromyography and Kinesiology*, 16(6), pp.541-548.
- [2] Navarro, X., Krueger, T., Lago, N., Micera, S., Stieglitz, T. and Dario, P. (2005). A critical review of interfaces with the peripheral nervous system for the control of neuroprostheses and hybrid bionic systems. *Journal of the Peripheral Nervous System*, 10(3), pp.229-258.
- [3] Asghari Oskoei, M. and Hu, H. (2007). Myoelectric control systems—A survey. *Biomedical Signal Processing and Control*, 2(4), pp.275-294.
- [4] Bunderson, N. and Kuiken, T. (2012). Quantification of Feature Space Changes With Experience During Electromyogram Pattern Recognition Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(3), pp.239-246.
- [5] Powell, M., Kaliki, R. and Thakor, N. (2014). User Training for Pattern Recognition-Based Myoelectric Prostheses: Improving Phantom Limb Movement Consistency and Distinguishability. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(3), pp.522-532.
- [6] Blum, A. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2), pp.245-271.
- [7] Parker, P. and Scott, R. (2018). Myoelectric control of prosthesis. *Critical reviews of biomedical engineering*, 13(4), pp.143-151.
- [8] Ortiz-Catalan, M., Brånemark, R. and Håkansson, B. (2013). BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. *Source Code for Biology and Medicine*, 8(1), p.11.
- [9] Input-h2020.eu. (2018). input-h2020.eu – INPUT EU Project H2020. [online] Available at: <http://www.input-h2020.eu/> [Accessed 31 Aug. 2018].
- [10] Kuiken, T. (2009). Targeted Muscle Reinnervation for Real-time Myoelectric Control of Multifunction Artificial Arms. *JAMA*, 301(6), p.619.
- [11] Motulsky, H. (n.d.). Intuitive biostatistics.
- [12] Graphpad.com. (2018). Home - graphpad.com. [online] Available at: <https://www.graphpad.com/> [Accessed 31 Aug. 2018].