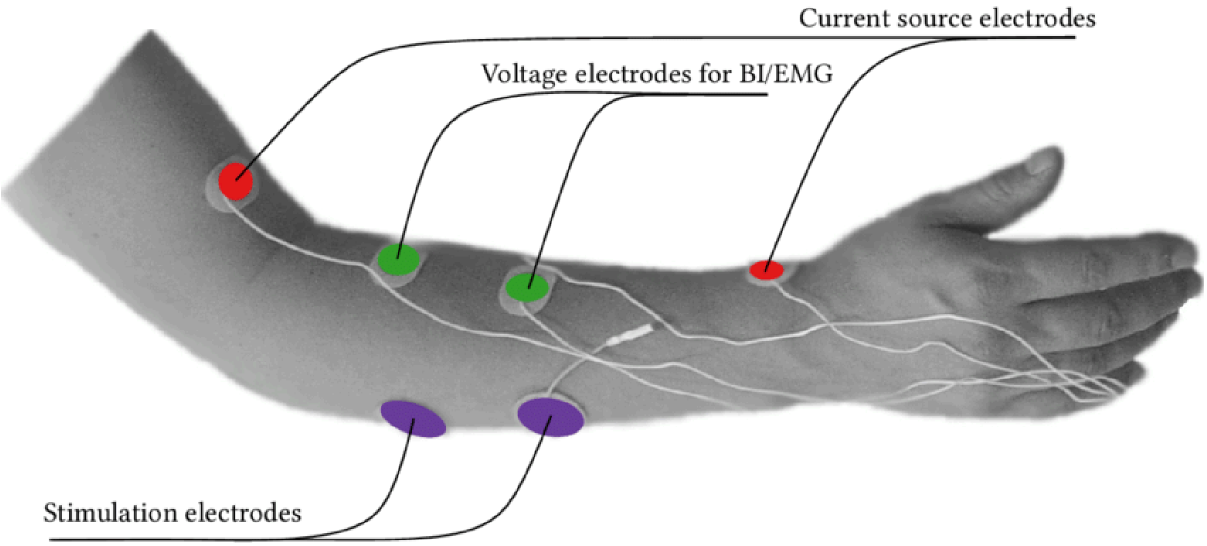
SORBONNE UNIVERSITÉ
FACULTÉ DES SCIENCES ET INGÉNIERIE

SCIENCES SORBONNE UNIVERSITÉ



Classification of experimentally acquired EMG signals

UE : MU5EEB12
Professeur: Nathanaël Jarrassé
Étudiants: Evan BOKOBZA Laurent LIN

PARIS, Décembre 2023

Resume

The practical work allowed us to observe the usefulness of electrodes to understand how muscles are involved during exercise. It particularly enabled us to learn how to label and extract relevant parts of the signal, demonstrating the reliability of processing tools for training data. Our results show an amount of error, this is because certain sequences are longer than expected, leading to a decrease in signal quality. Consequently, our sequence 6 demonstrated poor quality because of a misunderstanding of the task, so we experimented with both the training sequence and sequence 6 from a colleague in an attempt to correct the issue.

Table of contents

Introduction

The human body can perform countless movements, especially within the domain of arm movements. The analysis of these actions, in particular the signals produced to identify the muscles required for a particular type of movement.

The experiment carried out with the Noraxon DTS system and electrodes provide a way to identify the muscles used during different tasks done in sequences. Using a toolbox, we are going to develop process systems to recover the signals and analyse the rate of correlation between the actual movements executed and the signals detected.

By understanding the relationship between muscle activation and specific movements allow us to understand not only human behavior but also to improve the robotic system. Such improvement has the potential to improve the daily life of numerous people.

I Experimental recording of surface electromyograms (EMG)

The EMG signals are recorded by the Noraxon DTS system. These EMG signals are used to measure various actions performed by the user. As a result, we obtain sequences of movements in the (.mat) format, enabling us to analyze them using MATLAB. In order to classify the movements of the user, we use the standard class code such as :

Standard class code to use to label the EMG data

- Rest: 0
- Pinch (Thumb/Index only) Closing / Opening: 1 / 2
- Hand Closing / Opening: 3 / 4
- Wrist Flexion / Extension: 5 / 6
- Wrist Pronation / Supination : 7 / 8
- Elbow flexion / Extension : 9 / 10

Figure 1. Standard class code for EMG data

In order to execute the motion, six electrodes were placed on :

- Biceps
- Triceps
- Wrist flexor
- Wrist extensor

These recording placement supply the necessary informations to evaluate :

- Elbow (flexion/extension)
- Wrist (flexion/extension and pronation/supination)
- Hand (closing/opening)
- Thumb-index pinch (closing/opening)

The experiment adheres to a pattern of 10 seconds of effort followed by a 10-second rest period to complete all movements twice, in the aim of creating a sequence (fig.2). At the end, six sequences were created, which each one has a specificity: normal distributions, random movements, simultaneous movements, different placement of arm, faster actions and without one electrode.

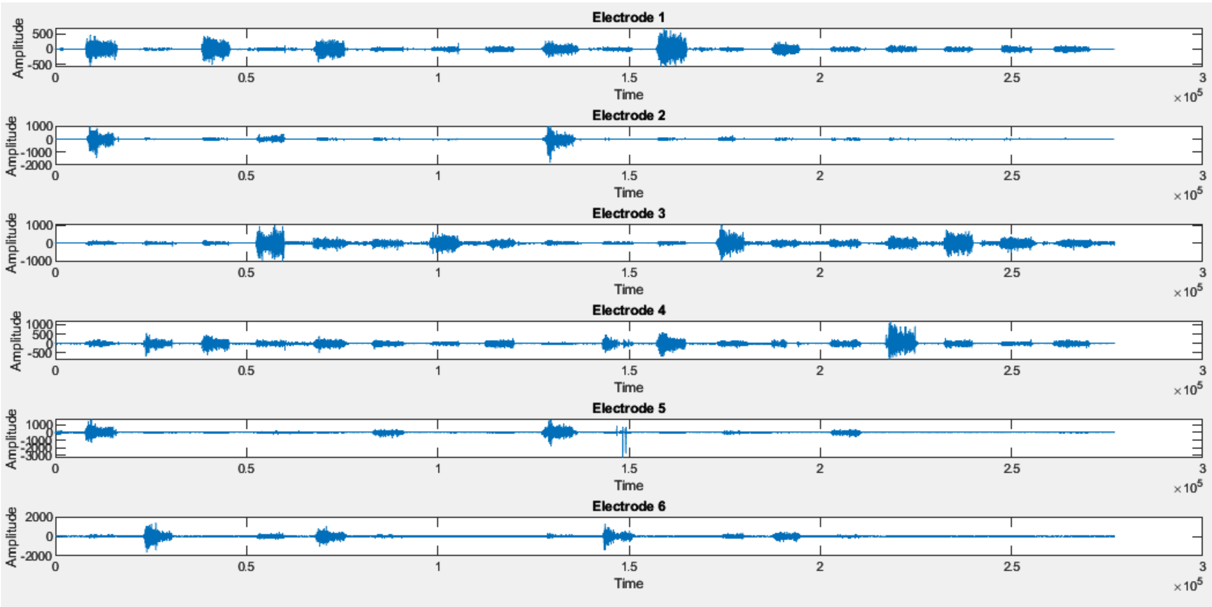


Figure 2. Amplitudes received by electrodes

II EMG signal classification toolbox

a . Toolbox

The experiment is used to recover the .mat files for the different sequences. Functions are used to extract the features and label the sequences, and to use this labeling to incorporate it into the other sequences. This will demonstrate the reliability of the classification.

The load_data function is designed to retrieve values from a .mat file. Upon reading the file, a Butterworth filter is applied to limit the signal between 10 and 400 Hz. Additionally, it resamples the signal to fall within the frequency range of 1000 to 3000 Hz. The function also deletes values corresponding to rest time. The file 's4t1data.daq' contains data acquisition for the signal s4t1. The 's4t1index.mat' file provides columns of motion (from 1 to 7) and the starting index for each motion. Motion refers to various types of actions analyzed through EMG signals, and the start index indicates the time interval occupied by different types of movements.

The purpose of the extract_feature function is to compute the RMS and AR on the data in order to extract features of a .daq file which concatenated two types of extraction such as RMS and Auto Regressive features (fig.3). ‘getrmsfeat’ and ‘getarfeat’ functions are tools to extract features. They take the data, window size and window incrementation to return a 2-dimensional matrix representing by window and features for each signal. The dimensions of the input data and the output data is determined by time interval between each feature.

Feature 1: RMS value

feat1(signal_win) = sqrt(mean(signal_win.^2));

Feature 2: autoregressive coefficients

An autoregressive model is a representation of a time-varying pseudorandom signal. This model defines the output value as a linear combination of past output values.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

with c a constant and epsilon_t white noise.

Figure 3. Equation of each type of feature

The 'getclass' function contains data, movement, index movement, window_size, and window incrementation as input arguments. After viewing the content of getclass, the output is a 1-D matrix which shows the motion class for each second.

After computing the difference betweens classification_testing and class_testing, it is showing a column mostly with value 0 if the prediction is correct, otherwise they would be another value. The function classification_timeplot explore each class which is looking for the correct or incorrect index and it plot the the result (dot/lines are the correct ones and cross are the wrong ones)

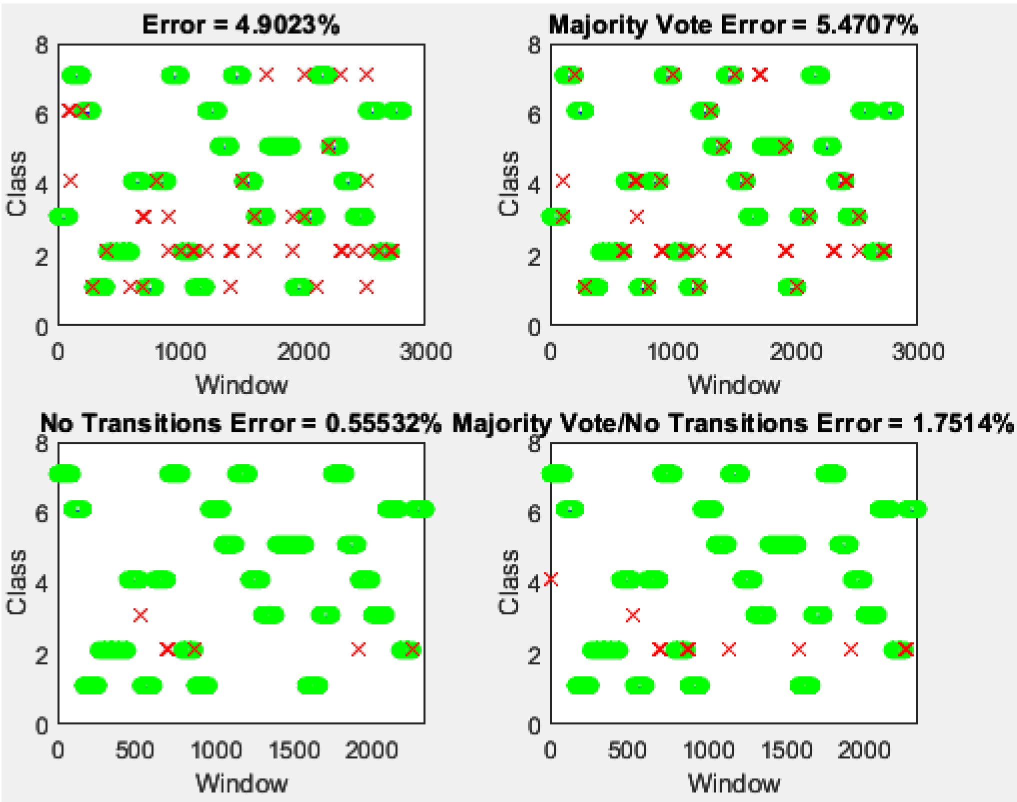


Figure 4. Classification_timeplot

'remove_transition' function helps to remove data from transition state, it takes a series of feature vectors with class labels and removes transition values in order to return a new series of feature vectors with class labels. Majority vote is used to remove noise from the signals. The addition of remove_transition and majority vote allow the signals to be more accurate and compute less error during the training.

Confusion matrix (fig.5) is a 3D visualization which helps to quantify the quantity of correct and incorrect behavior for our class between inputs and outputs.

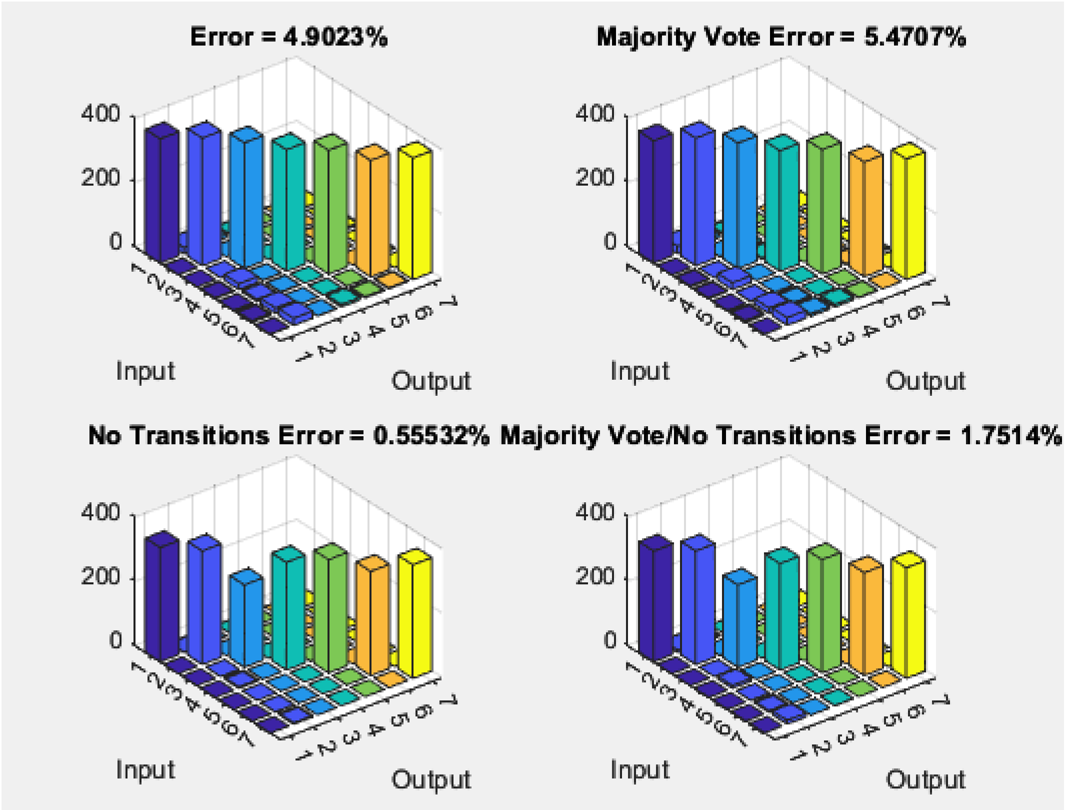


Figure 5. Confusion matrix

b. Labeling and pre-processing

The process of labeling (fig.6) is mandatory in signal processing. The labeling permits the function to explore the more adequate part of the signal with the training set.

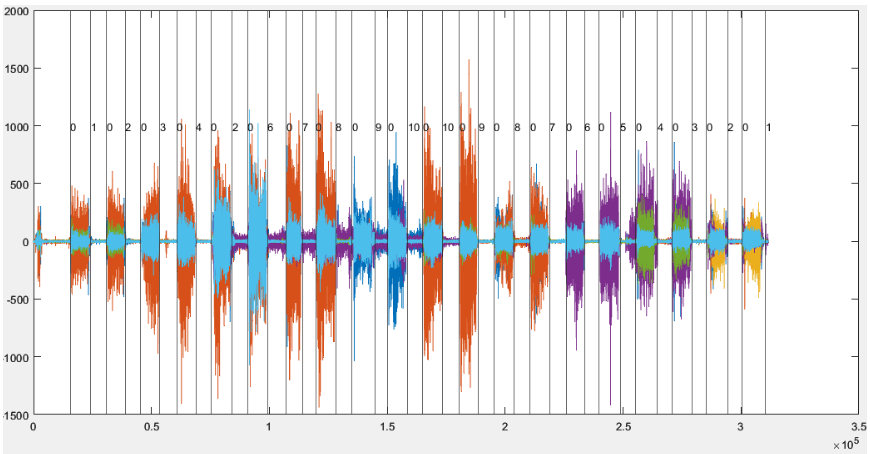


Figure 6. Labeling the motions

Depending on the quality of the signal, it may be more beneficial to either expand or contract our zone of interest (fig.7). In our case, enlarging the zone of interest is preferable as it improves the alignment between our training and testing sequences. This adjustment helps ensure that the model generalizes well across different sequences by considering a broader context during training and testing.

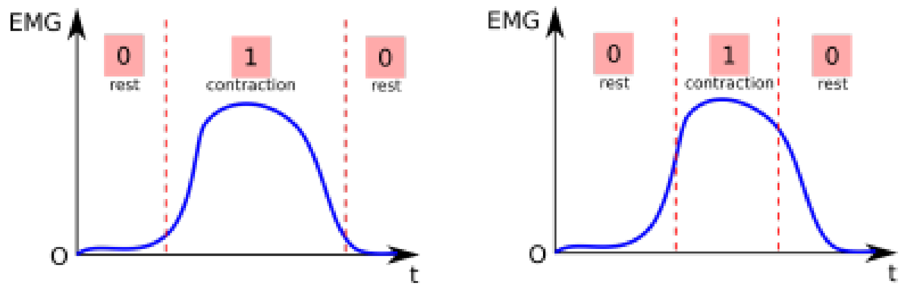


Figure 7. Pre-processing signal graphic

III Results and performance analysis of the classification approaches

For our experiment, we annotated numbers based on movement and assigned a specific number to each muscle according to its electrode. From this information, we obtain

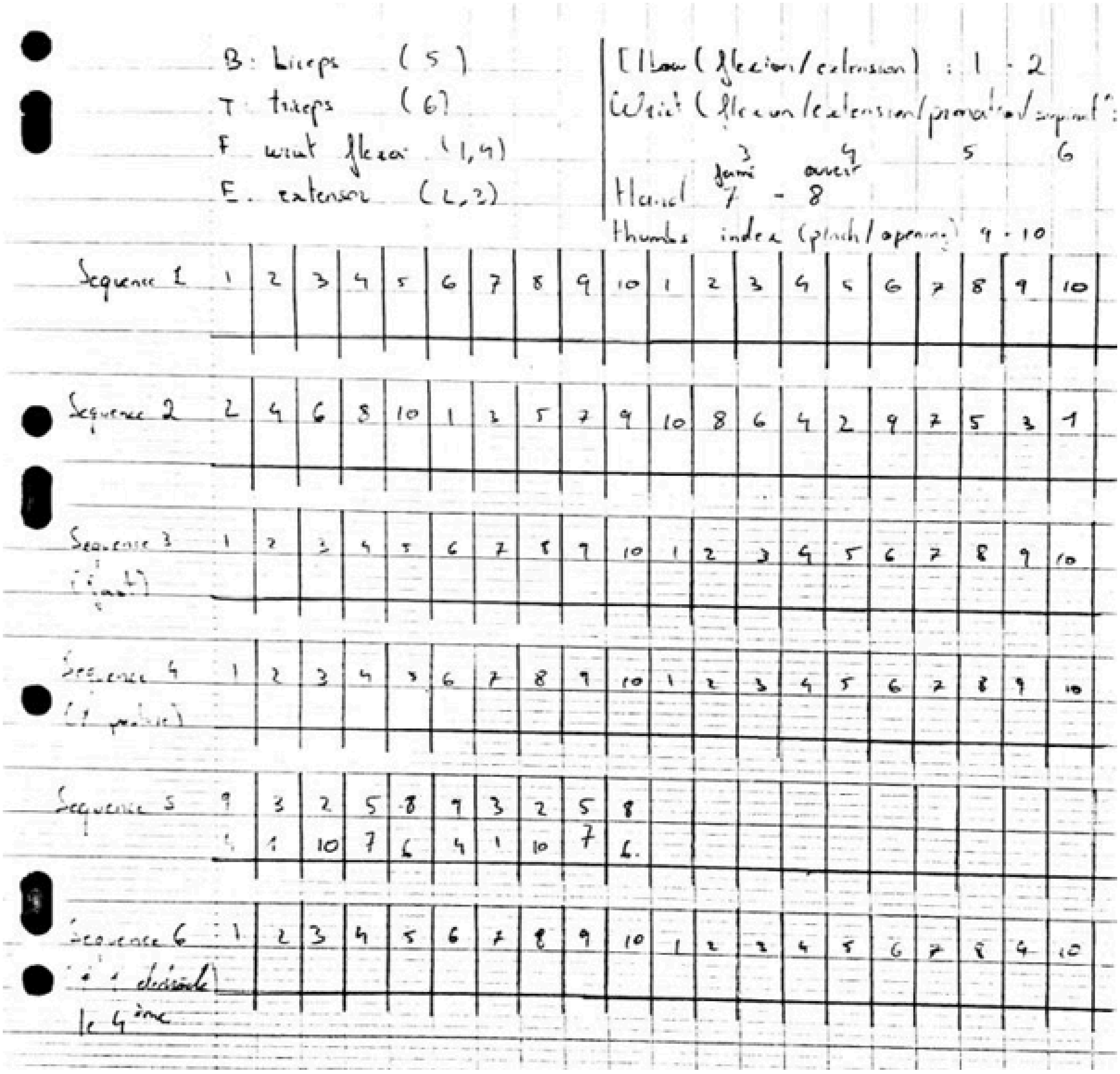
Electrode 1	Wrist flexion
Electrode 2	Wrist extension
Electrode 3	Wrist extension
Electrode 4	Wrist flexion
Electrode 5	Biceps
Electrode 6	Triceps

Figure 8. Placement of electrodes on the muscle

Elbow (flexion/extension)	1 - 2
Wrist (flexion/extension and pronation/supination)	3 - 4 - 5 - 6
Hands (on/off)	7 - 8
Thumbs (close/opening)	9-10

Figure 9. Attribution of numeric value for each motion

Those attributions are being used to be aware of which data corresponds to their motion. We defined a plan (fig.10) in order to realize the asking attribution of movement for each sequence.


B: biceps (5)
T: triceps (6)
F wrist flexor (1,4)
E. extensor (2,3)
Elbow (flexion/extension) : 1 - 2
Wrist (flexion/extension/pronation/supination): 3 4 5 6
Hand fermé 7 - ouvert 8
Thumbs index (pinch/opening) 9 - 10

Sequence 1	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
Sequence 2	2	4	6	8	10	1	3	5	7	9	10	8	6	4	2	9	7	5	3	1
Sequence 3 (fast)	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
Sequence 4	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
Sequence 5	9 / 4	3 / 1	2 / 10	5 / 7	8 / 6	9 / 4	3 / 1	2 / 10	5 / 7	8 / 6										
Sequence 6 le 4ème	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Figure 10. Plan of attack for the experiment

After extracting our signals from the electrodes (6 in totals). First of all, we need to label our data in order to categorize it. We apply the labeling to the 6 sequences to obtain an index for each of the motions. Then we applied a normalization by subtracting the data by its mean and dividing by the difference between the maximum and minimum of the data to have a lower error rate. There was the 'zscore' function that allowed standardization, but the results were not as good.

To evaluate our developed tool on our data, we introduced new files like 'load_data_exp.m,' 'labeling_exp.m,' and 'visualize_exp.m.' Upon executing the LDA classification (visualize_exp.m), the obtained results consistently hover around a 50% error rate. For the confusion matrix, in the aim of looking at the interested motions, we eliminate the correct (between training and testing sequences) quantity of rest. Interestingly, when testing the tool on the last sequence, which is from another group due to the realization that our last series of motions contains errors, the error rate surprisingly drops to around 30-40% which should be higher because without one electrode, noises would increase .

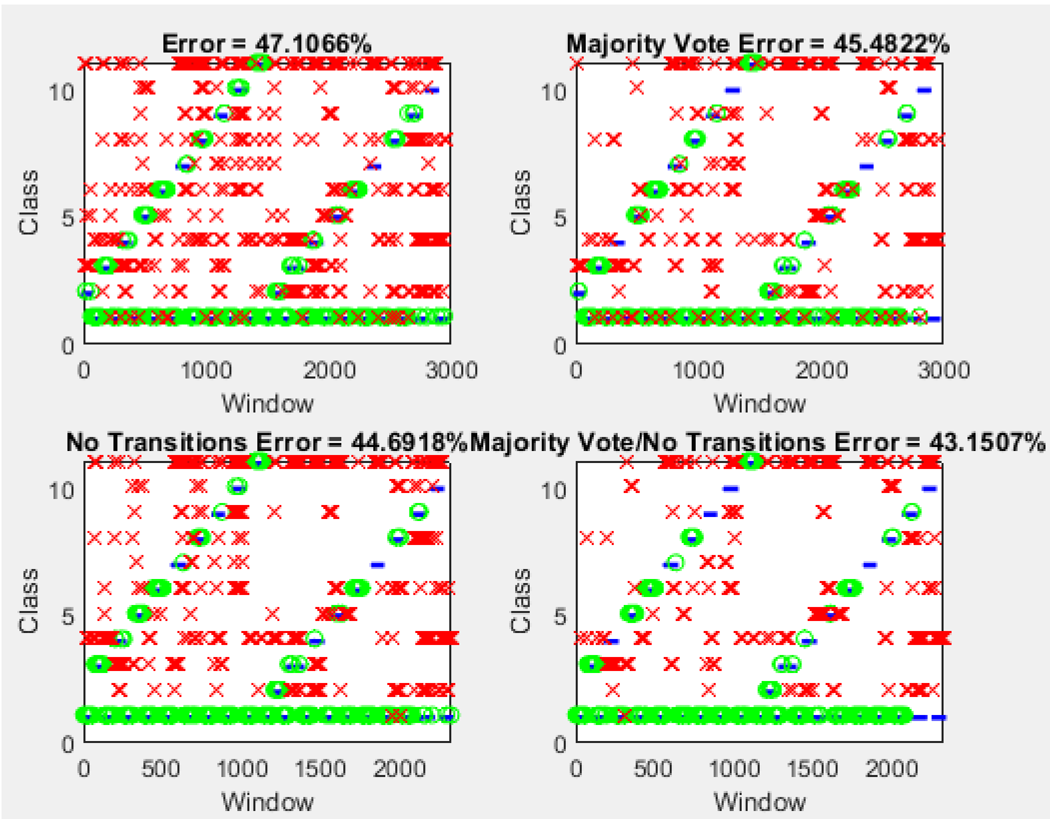


Figure 11. Classification of the experimentation data

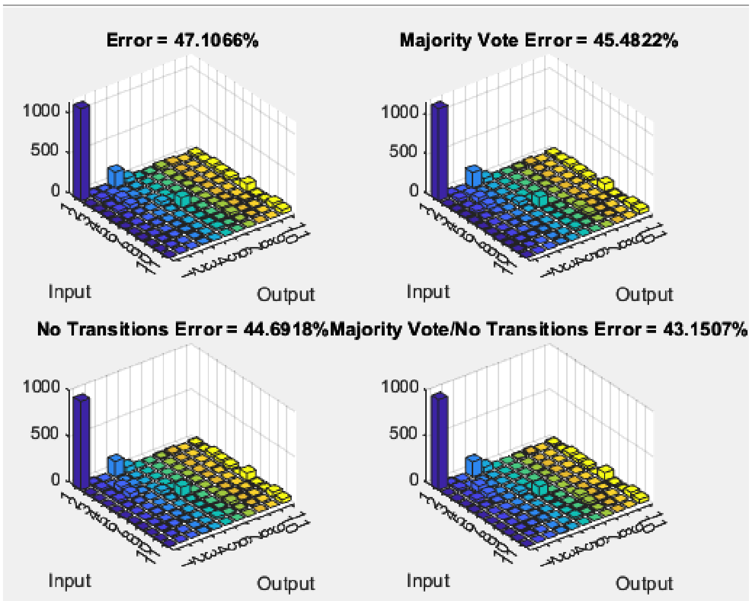


Figure 12. Confusion matrix of the the motions

Figure 13. Confusion matrix of the movement without the correct noise

In conclusion, the ratio of error suggests that the robustness of our classifier is not satisfactory. To enhance its performance, we should consider incorporating additional preprocessing or post-processing treatments. The results also vary depending on the user. Indeed, we are testing the movement of an individual's arm, but this person may use their arm muscles differently compared to other individuals.