ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Ecole Polytechnique Federale de Lausanne

Learning Algorithms and Systems Laboratory

Semester Project

A fast decode of grasping intention from the muscular activity

Antoine Weber
a.weber@epfl.ch

Sciper
229195

Prof. Aude Billard

PhD student Iason Batzianoulis

June 8, 2018

Contents

1 Introduction

Neuroprosthetic devices are used to restore motor abilities lost after pathologies or trauma. In the case of upper-limb prosthesis, an important functionality is grasping. To restore this ability, the prosthesis should be able to achieve rapidly a secure grasp. Thus, the device should adapt promptly to the characteristics of the object, such as its size and its texture. The object's characteristics (mass, friction at the surface) are directly connected to the applied forces to the object and an inaccurate estimation of those can lead to an unsuccessful grasp. To apply the proper forces, the device should make use of the information coming from tactile sensors to determine the real weight and friction of the object.
On the other hand, the configuration of the fingers should also depend on the user's intention. The prosthetic device should identify accurately the grasping intention and perform the desired grasp type. The grasping intention could be decoded from the muscular activity of the arm in the early stages of the reaching motion [1]. Therefore, the prosthesis could predict the grasping type from the muscular activity, command the fingers to close in a desired configuration and make use of the tactile informations to securely grasp the object.

The goal of this project was to predict the grasping intention in the early stages of the reaching motion using 8 differential EMG channels. Reach-to-grasp motions were assessed using two different grasp types : the thumb-2 fingers grasp (the object is lifted using only the thumb, the index and the middle finger) and the power grasp (with all the fingers). On the other hand, the reach-not-to-grasp motions were assessed using a single grasp type being simply the no-grasp (the hand reaches the object but do not grasp it).
The grasping predictions were computed and compared with 4 different classifiers being an Echo State Network (ESN), a Linear Discriminant Analysis (LDA), a C-Support Vector Machine (C-SVM) and finally a Gaussian Mixture Model (GMM) using three different approaches to analyze the data.

2 Materials and Methods

All the different processings were performed on Matlab2018a. SVM was implemented using the LIBSVM toolbox [7]. The data were collected from a single subject through 3 sessions.

2.1 Experimental Setup

Different types of data were recorded for this semester project. The EMG data were captured from 8 different sites and the corresponding targeted muscles are presented on Table 1. It was recorded with the Noraxon DTS desktop system with a sampling frequency set to 1500 Hz. During the experiment, the subject was asked to perform reach-to-grasp and reach-not-to-grasp motions towards three directions (left, right and center).
For the segmentation of the trials, a manual trigger was introduced to set the onset and end of the reaching motions. Moreover, the ground truth being the labels of the different performed reaching motions were set by hand during the experiment. The subject was free to perform the grasp type of it's choice during the session. However, if an unbalancing between the different grasp types was observed at the end of the sessions, the subject was asked to perform some more of the missing grasp types.
The different joint angles of the fingers were measured using CyberGlove System's CyberGlove [1]. These data were measured to assess whether the labels of the thumb-2 fingers grasp and power grasp could be set by computing the aperture of the hand when the object was grasped. Indeed, the hypothesis was, from the point in time when the object is grasped, the aperture between the thumb and the index would be significantly different between the two grasp types. Such an analysis could be used to differentiate the two reach-to-grasp motions, however it could not be used to determine the reach-not-to-grasp labels.

	Name
1	Biceps Brachii long head (BICL)
2	Triceps Brachii long head (TRIC)
3	Flexor Digitorum Superficialis (FLDS)
4	Flexor Carpi Ulnaris (FLCU)
5	Extensor Digitorum Communis (EXDC)
6	Extensor Carpi Radialis longus (ECRL)
7	Extensor Pollicis Brevis (EXPB)
8	Abductor Pollicis Brevis (APB)

Table 1: Set of tracked arm muscles. All these muscles were tracked differentially.

Finally the third and last type of recorded data was the kinematics of the subjects arm. The arm kinematics were used to segment the data into the different grasp trials and also different motion phases as will be explained later in the report. those data were recorded using an OptiTrack motion capture system.
Indeed, the manual triggers allowed to locate the full motion being from the onset of motion to the hand going back to it's original position. However, the returning motion of the hand being from the releasing of the object to the initial position did not contain any relevant information. Thus, the kinematics of the hand were used to keep only the interval of interest being from the onset of the motion to the grasping of the object. I will refer to this interval of interest as *trial*.
Three different markers were positioned on the hand, elbow and shoulder of the subject. These latter markers were tracked using IR cameras positioned in strategical spots in the room to continuously give the 3-axis coordinates of the different markers. Both CyberGlove and kinematics data were

[1]http://www.cyberglovesystems.com/

sampled at around 110Hz.
Figure 2.1 illustrates the complete setup on the subject.

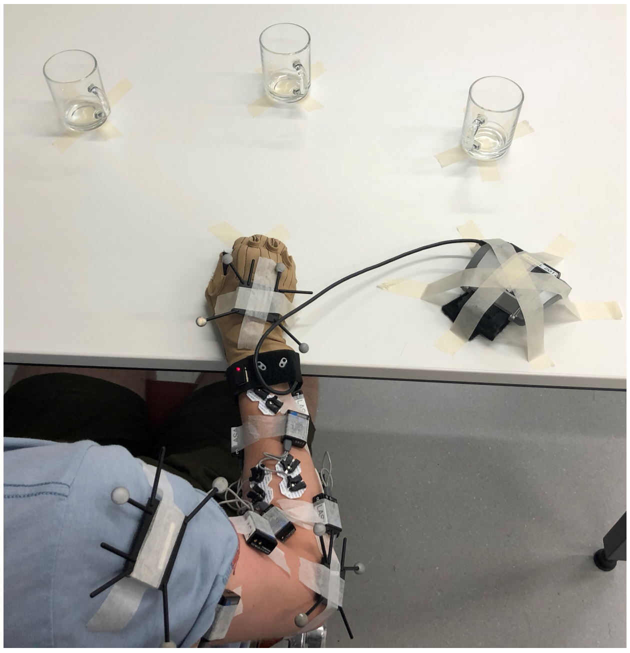


Figure 2.1: Experimental setup with the EMG recording devices, the CyberGlove and the 3 different markers.

When the hand was at rest, its mean initial position was computed at $-2.66\,\text{cm}$ on the y axis with a standard deviation of 5.65 mm using the arm kinematics. Considering these results, as the reaching motion pointed towards the positive direction of the y axis, the onsets of the trials were defined using a distance threshold of 2.66 cm on the y coordinate of the hand.

To define the end of the trials for the different reach-to-grasp motions, a slightly different method was implemented. The average maximum height when picking the cup was computed at 1.60 m along the z axis with a standard deviation of 6.4 cm. To be certain that the end of the trial was defined whenever the cup was grasped, a threshold being at $\mu - \frac{2}{3}\sigma = 1.56[m]$ along the z axis was implemented. For the reach-not-to-grasp motions, the maximum observed value on the y axis was chosen as the end of the trial as it represented whenever the hand reached the object without grasping it.

For both reach-to-grasp and reach-not-to-grasp motions, approximately 90 ms of signal were taken as a margin before and after the computed onset and end of the trials. Illustrations of the trials definition for each grasp type can be observed on Annex A.1 and A.2.

2.2 Data Processing

The workflow displaying the data processing is illustrated on Figure 2.2.

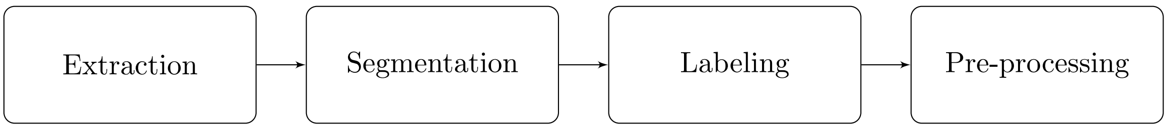


Figure 2.2: Worklow of the data processing

The Extraction consisted of reading and storing all the data from all the files of the different sessions. The manual triggers allowed to locate the entire reaching motion from the onset of the motion to the hand going back to it's original position.

As said in the previous section, the full grasping motion was segmented using the kinematics of the hand. Indeed, the z and y coordinates allowed to detect whenever the hand was going up (object grasped) or stopped progressing in the y axis (the hand reached the object). This allowed to isolate all the different *trials*.
As the different labels were set during the experiment, once all the trials were defined, the same number of trials than the number of set labels should be found. From this step, the different trials and labels could be merged.

2.2.1 EMG Pre-processing

The raw EMG data could not be directly used as it contained high frequency noise as can be seen on Figure 2.3.

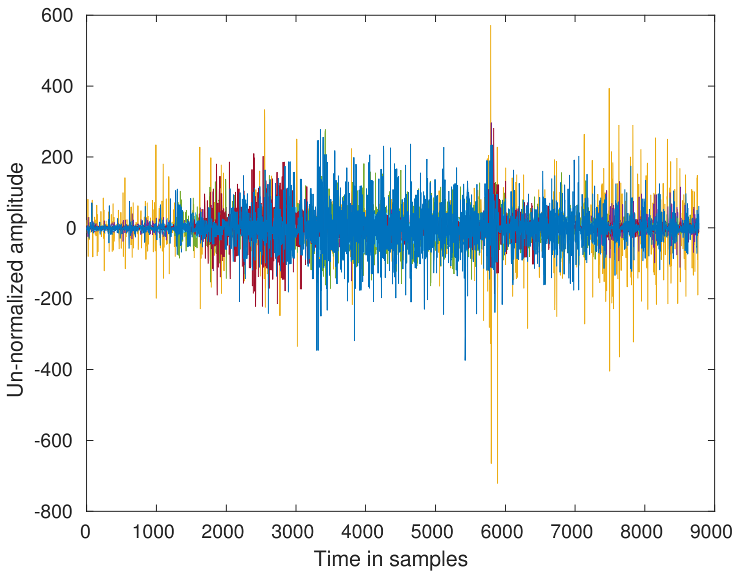
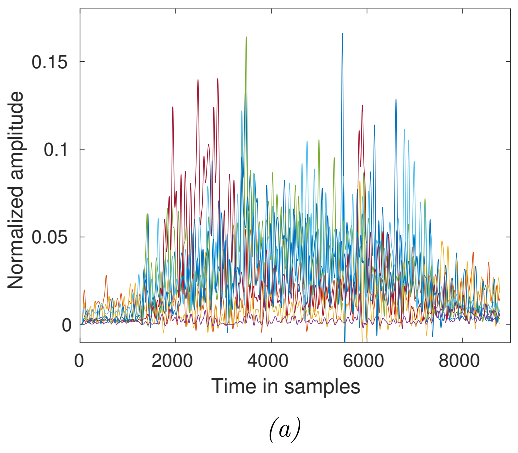


Figure 2.3: Raw EMG data of a reaching motion. The data contains high frequency noise that has to be filtered.

Hence a pre-processing was performed on the raw EMG data. It consisted of 4 main steps [1]

1. Bandpass filtering
2. Rectification
3. Low-pass filtering
4. Normalization

The first bandpass filtering was performed to keep only signals between 40-400 Hz which is the interval where the muscle activity contains relevant informations. Then a rectification was implemented followed by a low-pass filtering to keep only small frequencies (under 20 Hz) to catch the envelope of the signals. Finally the per-channel normalization has been performed by dividing each channel by its MVC (Maximum Volunteered Contraction) which was the maximum observed value per channel through all the sessions.
The pre-processed data is illustrated on Figure 2.4.



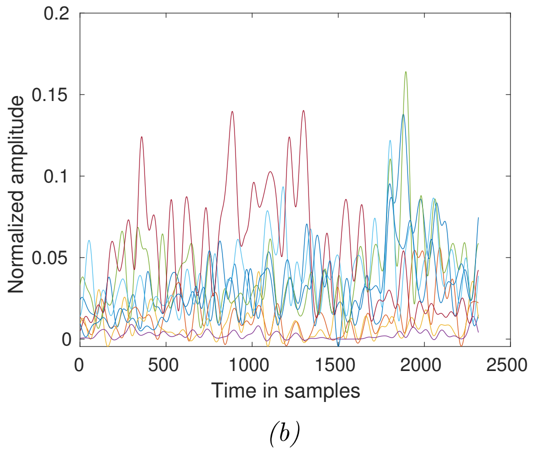


Figure 2.4: (a) represents the pre-processed EMG data for an entire reaching motion, without having used the kinematics to isolate the trial while (b) represents the pre-processed EMG data, keeping only the interval of the trial

2.3 Features Extraction

Depending on the implemented algorithm, different features had to be extracted to best encapsulate the data characteristics.
Neural Networks are generally used to classify raw signals. Different architectures may be implemented to mimic the feature extraction such as CNNs [8]. However, most of Machine Learning algorithms have computational costs growing at least linearly with the dimensionality of the data [9], which could result in very slow training. Training complexity can be of huge importance whenever it is performed *on-line*. Nonetheless, the addressed problematic of this project would involve an *off-line* training, which makes the training complexity of less importance.
The different features were computed channel-wise and then concatenated into one single vector. As the EMG signals were measured on 8 channels, computing 3 features per channel resulted in 24 dimensional vectors. The computed features were [1, 2, 3, 4] :

1. Mean Absolute Value
2. Number of Slope Changes
3. Waveform Length

The Number of Zero Crossings is also a common features for such a task. However, as a signal rectification was performed during the pre-processing, this feature would only result in null components.

2.4 Time Windows definition

The different grasping trials were time series. It is straightforward to consider that different trials may not be of the same length. This became a problem with classifiers receiving inputs of fixed dimensions such as an Echo State Network. Even if the other implemented classifiers also required inputs of fixed dimension, the dimensionality of the input was set by the computed features and not by the length of the signal itself. This could be resolved by normalizing the signals over time, inducing a constant input length. However, such a normalization cannot be implemented for an *on-line* approach as the decision of the classifier would be outputted only at the end of the trial.
To correctly assess such a problem, the different trials were separated into time windows with a small overlap between the windows. This way, the signals given as input to the classifier were of fixed length, even without computing features. Moreover, such a representation allowed to simulate the implementation of the algorithm on a real device with a fixed sampling frequency. Hence, data

would be sent to the algorithm every X ms, inducing each time a new prediction of the grasp trial. The time window length was set to 200ms with a 50s overlap. Such data representation is illustrated on Figure 2.5.

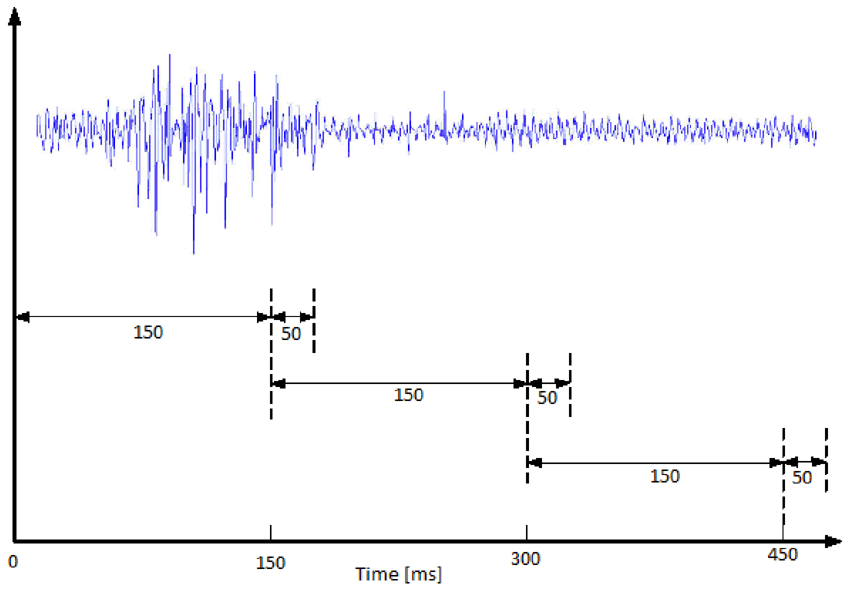


Figure 2.5: Time windows segmenting illustration [2]

2.5 Classification

To try to identify the different grasping types, different classifiers were implemented to compare their results. Three main approaches were implemented: the first one being to simply compute a single classifier using data from all the time windows, and assess its results. Then, one classifier per time window was trained to investigate whether this would be a better approach. Finally, the trials were segmented into 3 phases. The first two being the acceleration and deceleration of the hand and the third one being the grasping of the object. The three phases were detected using the kinematics of the hand. Indeed, the first two phases were respectively represented by an acceleration and decelerations of the hand, resulting in a positive and negative second derivative of the position. Once these two phases were set, the third one was deduced to be the remaining components of the signal. An example of the segmentation of a trial can be visualized on Figure 2.6.

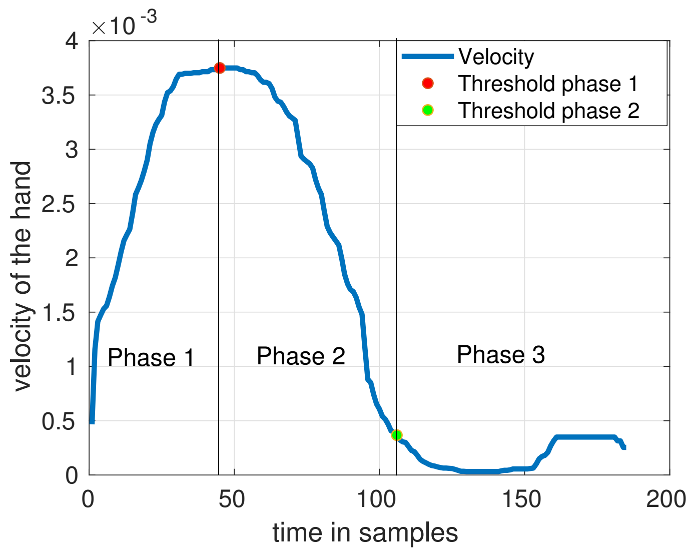


Figure 2.6: 3 phases definition of a trial.

The first phase was defined from the onset of the trial to the maximum reached velocity of the hand which can be observed as a red dot on Figure 2.6. The second phase started directly from the end of the first phase until the velocity reached a threshold set to $0.4 * 10^{-3}$, illustrated as a green dot on the same figure. This threshold was set to define when the hand reached a given velocity being quite low, to represent the arrival close to the object. Finally the third phase was defined from the end of the second phase until the end of the trial. No signal margins were taken when assigning the positions of the switching between the phases.

The different implemented classifiers for this project were :

1. Echo State Network
2. Linear Discriminant Analysis
3. C-Support Vector Machine
4. Gaussian Mixture Model

All different approaches were cross-validated using a 5-fold cross-validation while making sure the class balancing was acceptable on each fold

3 Results

3.1 Labels Certification

As said in section 2.1, a prior hypothesis was made on the data being that the thumb-2 fingers and power grasp labels could be set using the aperture of the hand at the end of each trial.

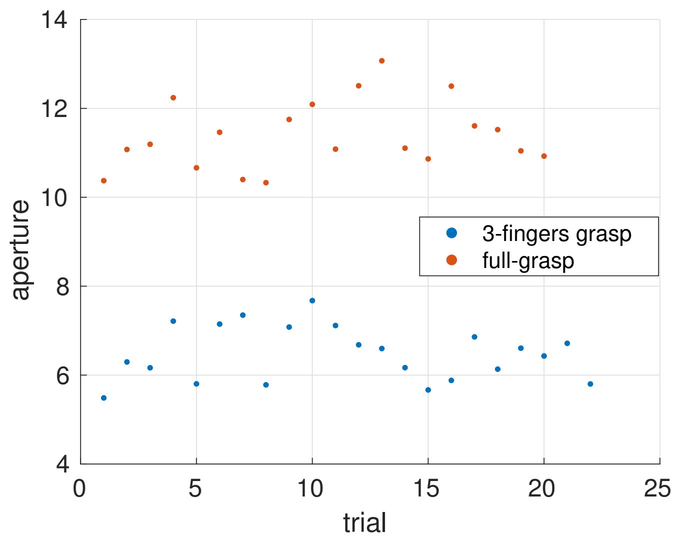


Figure 3.1: Plot of the aperture of the hand at the end of each trial not being a no-grasp motion. A clear threshold is present at an aperture of around 9units.

Referring to Figure 3.1, it is easily observable that the two different grasp types could be separated using a hard threshold at an aperture of $\approx 9units$. To ensure that the two distributions were statistically significant, a student t-test was implemented. The results indicated that the null-hypothesis (meaning that the means of the two distributions are equal) could be rejected with a very small p-value ($p = 1.8 * 10^{-23}$). As this p-value is smaller than a significance level of 5% which is a standard significance level, the conclusion that the difference between the means of the two distributions were statistically significant could be set. This means that the ground truth labels could be set by performing such an analysis without needing a manual setting of the labels during the experiment. However, one would need another trigger to differentiate reach-to-grasp and reach-not-to-grasp trials as the finger placement of the hand do not follow any law when not grasping the object. Hence, such an analysis could not be sufficient to determine the no grasp labels. To be able to identify the reach-not-to-grasp motions, a pressure sensor could be installed under the cup to detect whenever this latter one was being lifted.

3.2 ESN Classification

The first implemented classifier was an Echo State Network. To implement an ESN, no extraction of different features was needed as the network should be trained with the pre-processed EMG data. Such a network has basically two main hyperparameters to be tuned which are : the number of hidden units and the spectral radius of the weight matrix.

The spectral radius always has to be smaller than 1 to ensure the so called *Echo State property* [6]. It is generally chosen close to 1. One should be more careful when choosing the optimal number of hidden units. Indeed, implementing too much hidden units will generally lead to an overfit of the training data, while implementing not enough hidden units will prevent the classifier to extract enough information to generalize the data and perform the classification.

The spectral radius being fixed to 0.9, the optimal number of hidden units was found using a grid-search over a list of different values. Figure 3.2 illustrates the best obtained results for one single ESN trained for all the time windows.

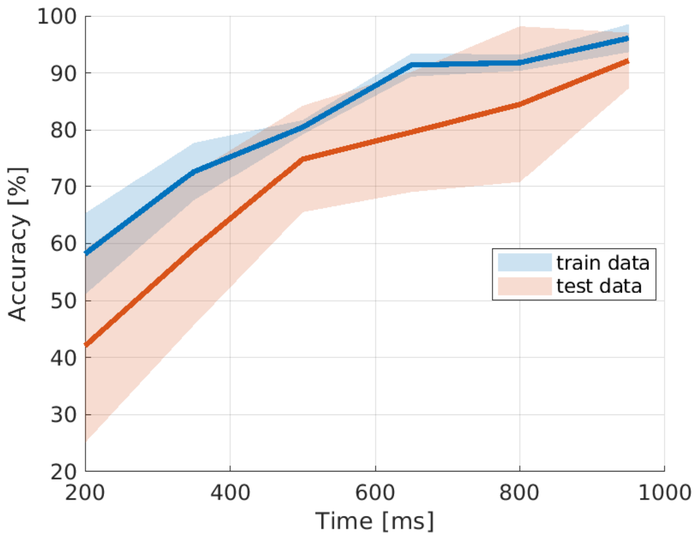


Figure 3.2: Accuracy of a single ESN classifier, tested through the different time windows

For a single trained ESN classifier for all the time windows, the optimal number of hidden units was found to be 250. One can observe from Figure 3.2 that the train and test accuracies increased with time to finally reach 92.2% accuracy on the test set with a standard deviation of 6.5%. Such results are already descent as there is no clear overfit while keeping a test score relatively high.

For implementing one ESN classifier per time window, the optimal number of hidden units was found to be 110. As the amount of data per time window was only a fraction of the data of all time windows, the number of hidden units being proportional to the number of parameters to optimize was found to be smaller with a per time window analysis compared to a single ESN classifier analysis.

The results of implementing one ESN classifier per time window is illustrated on Figure 3.3. The train accuracy remains through all the time windows above 90% while the test accuracy has a decreasing trend until 500 ms after the onset of the trial. This may be caused by outliers in the data as the algorithm should get more confident at the label prediction with time. Indeed, the closer we are to grasp the object, the more significant should the disposition of the fingers be to the wanted grasp type. However, obtaining a train accuracy not reaching smaller values than 90% and a test accuracy barely going under 80% throughout the entire trial is quite promising.

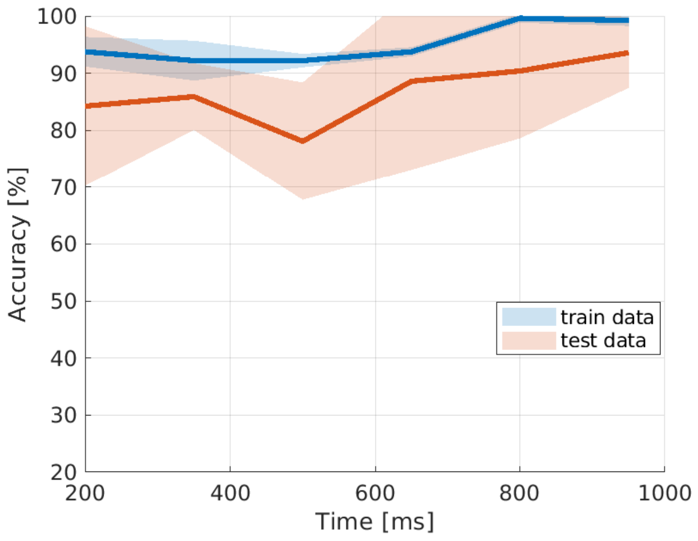
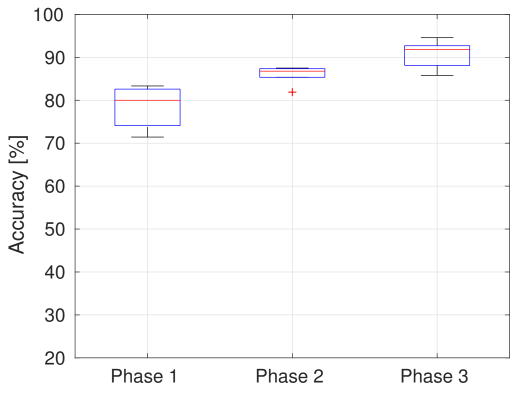
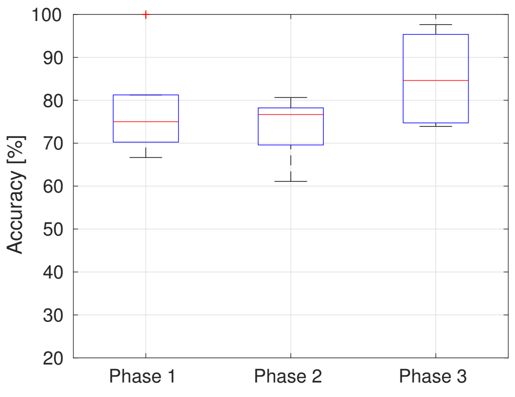


Figure 3.3: Accuracy of one trained ESN classifier per time window.

When implementing the 3-phases analysis, the optimal number of hidden units was found to be 200. Results are illustrated on Figure 3.4 and Table 2. Here the global trend being an increase in accuracy with time throughout the 3 phases. The train accuracy increased to finally reach a mean accuracy of 90.63% on the third phase while the test accuracy fluctuated to finally reach a mean accuracy of 85.15%.



(a)



(b)

Figure 3.4: (a) represents the obtained train accuracy by training one ESN classifier per phase and (b) represents the same analysis only on the test data. The median is represented with a red line and the bottom and top edges of the boxes represent the 25th and 75th percentile respectively.

	Phase 1	Phase 2	Phase 3
Train mean accuracy [%]	78.42	85.99	90.63
Train standard deviation [%]	5.07	2.33	3.37
Test mean accuracy [%]	77.62	73.65	85.15
Test standard deviation [%]	12.97	7.60	10.89

Table 2: Summary of the main results for the 3 phases approach implemented with ESN classifiers

3.3 LDA Classification

An LDA classifier do not require any hyperparameters to be set by the user, hence the results do not depend on the user's choice. Moreover, to implement an LDA classifier, the features presented in section 2.3 were computed. As the task was a 3-class classification problem, training an LDA classifier resulted in a classification into a $n-1$ dimensional space which resulted, for the given task, in a 2D plane. Thus the results could be visually observed.
The LDA embedding using data from all the time windows is illustrated on Figure 3.5

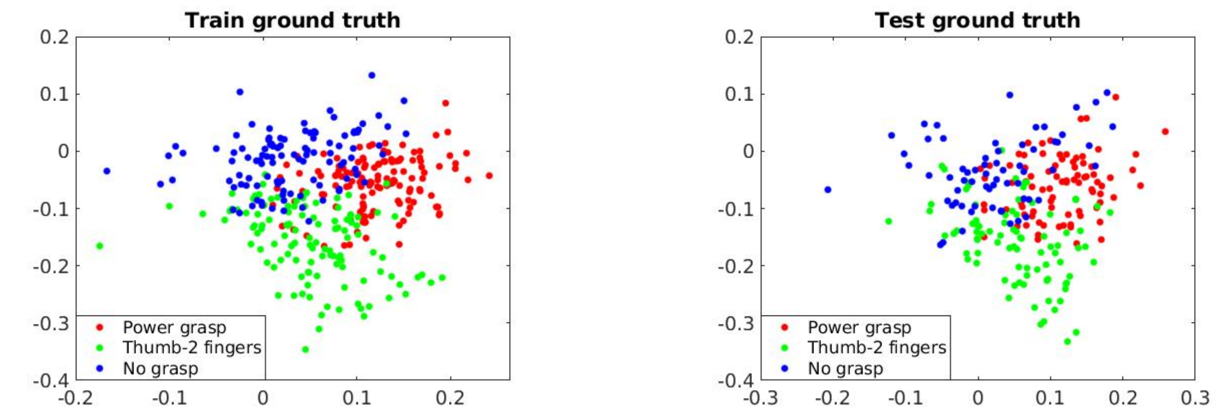


Figure 3.5: Train and Test embedding using LDA

The obtained classification boundaries and confidence matrix with training a single LDA classifier for all the time windows can be observed on Annex A.3 and A.4. A train and test accuracy of 79% and 70% respectively was observed. Considering that an LDA classifier performs linear classification on it's embedded space, these results can be surprisingly good. The results of this single LDA classifier tested per time window is illustrated on Figure 3.6.

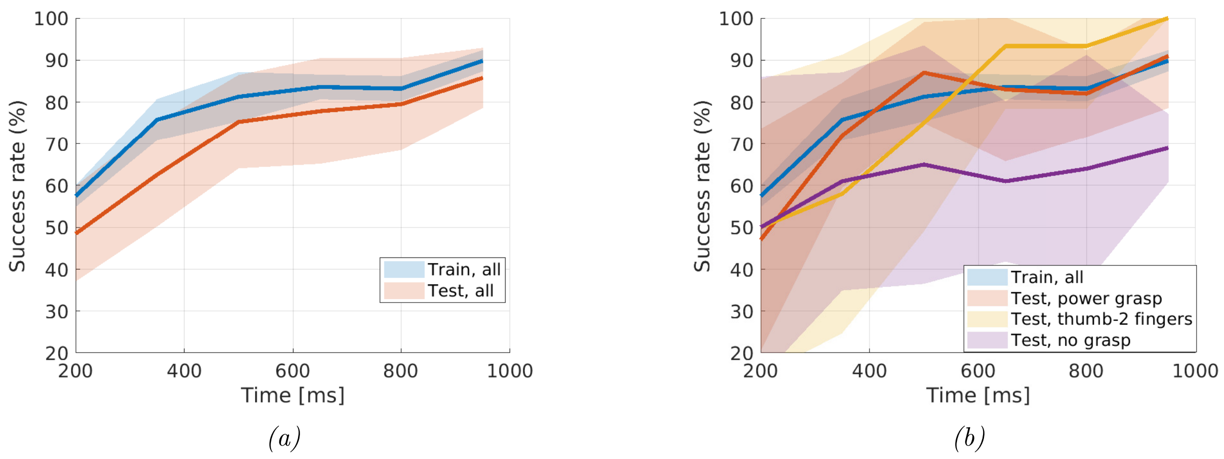


Figure 3.6: (a) represents the train and test accuracy with training one single LDA classifier for all the time windows, tested through all the time windows and (b) represents the same accuracy only per grasp type.

Looking at Figure 3.6 (a), an increasing trend of the accuracy with time was observed to reach a final test accuracy of 86.1% with a variance of 7.2% 950 ms after the onset of the trial. Moreover, it can be observed on point (b) that the classifier had troubles detecting the no-grasp motions.

Figure 3.7 shows the results of implementing one LDA classifier per time window. The same trend can be observed as with the ESN results. The test accuracy having a decreasing behavior until 500ms after the onset of the trial, then increasing until the end of the trial. Looking at Figure 3.7 (b), one can observe that this problem did not come from a single class, but was a general trend through the 3 classes.

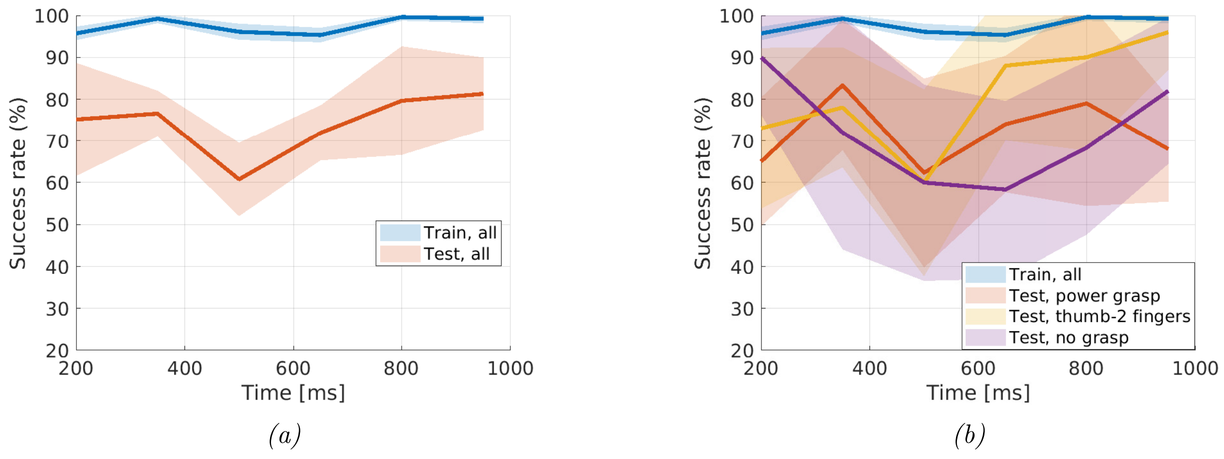


Figure 3.7: (a) represents the train and test accuracy with training one LDA classifier per time windows and (b) represents the same accuracy only per grasp type.

Finally, the 3-phases analysis was performed for one LDA classifier per phase and the results are illustrated on Figure 3.8 and Table 3. The train accuracy remained above 80% through the 3 phases while the test accuracy had an increasing trend with time to finally reach a mean accuracy of 86.05% on the third phase.

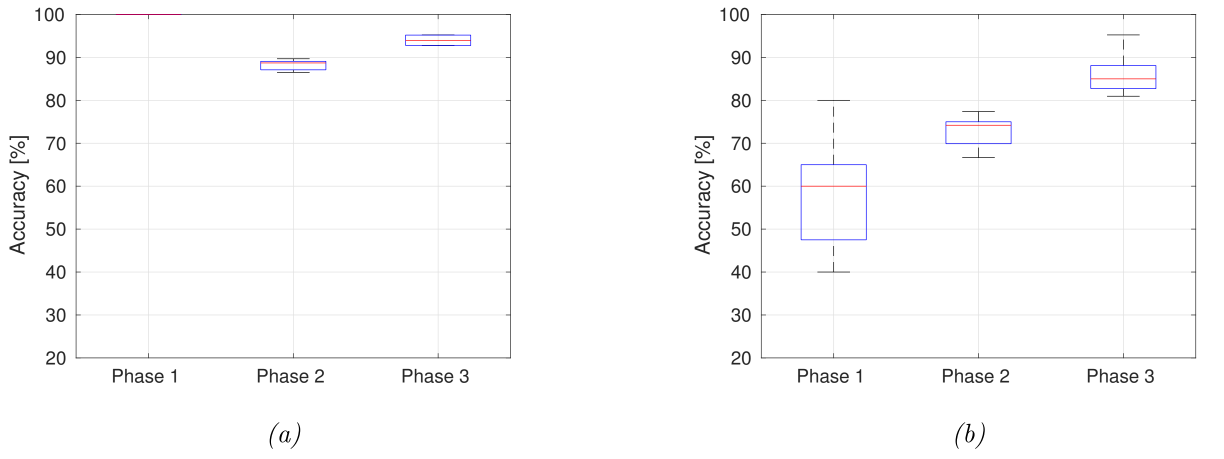


Figure 3.8: (a) represents the obtained train accuracy by training one LDA classifier per phase and (b) represents the same analysis only on the test data. The median is represented with a red line and the bottom and top edges of the boxes represent the 25th and 75th percentile respectively.

	Phase 1	Phase 2	Phase 3
Train mean accuracy [%]	100	88.22	93.99
Train standard deviation [%]	0	1.28	1.22
Test mean accuracy [%]	58.00	72.69	86.05
Test standard deviation [%]	14.83	4.07	5.45

Table 3: Summary of the main results for the 3 phases approach implemented with LDA classifiers

3.4 C-SVM Classification

To assess visually the performances of the C-SVM classifiers, the chosen data to be analyzed was the 2 dimensional data embedding from the LDA classifier. Different embeddings were tried with different dimensionality reduction algorithms such as PCA, KernelPCA, Locally Linear Embedding, Laplacian eigenmaps, etc... However the subspace allowing the best performances was found to be the induced LDA embedding using data from all the time windows.

As points were picked randomly, the training and testing points may vary from the ones depicted on Figure 3.5. The different hyperparameters of C-SVM which are the C factor and the kernel width σ were found through a grid search. Indeed, as the data was embedded in a 2D space, the trainig complexity was not acting as a bottleneck, allowing to perform such a technique to find the optimal hyperparameters. Figure 3.9 illustrates the train results of a single C-SVM classifier trained with data from all the time windows.

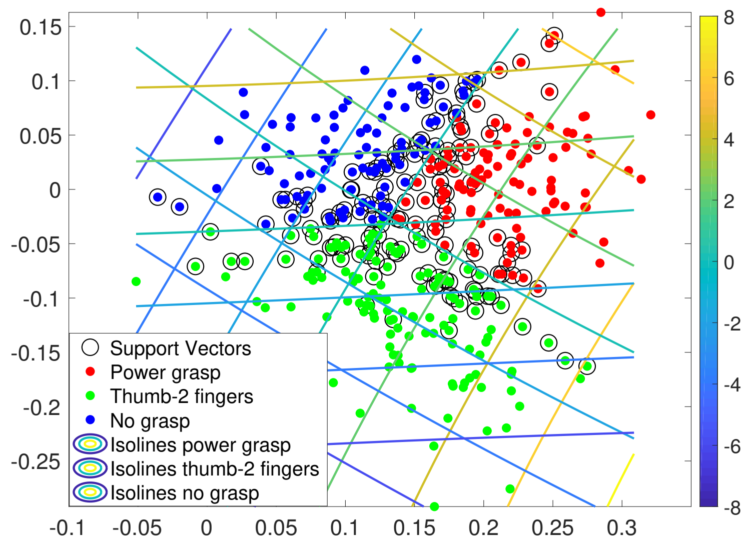
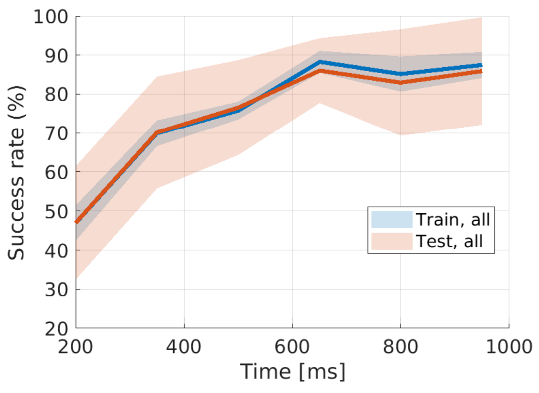
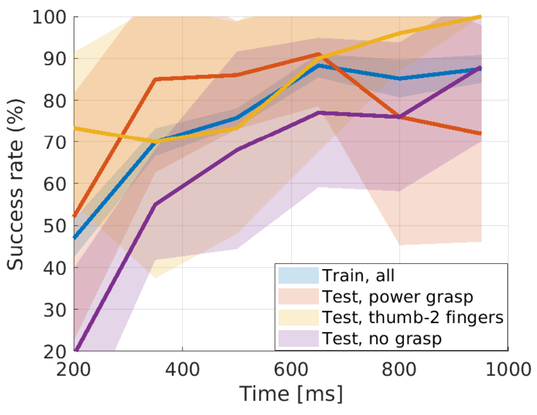


Figure 3.9: Decision boundaries of the C-SVM classifier on the training data

The resulting classification boundaries and confidence matrices can be observed on Annex A.5 and A.6. A global train and test accuracy of 78.4% and 77.3% was observed. Moreover, the results of the single C-SVM classifier for all time windows tested through all the time windows is illustrated on Figure 3.10.



(a)

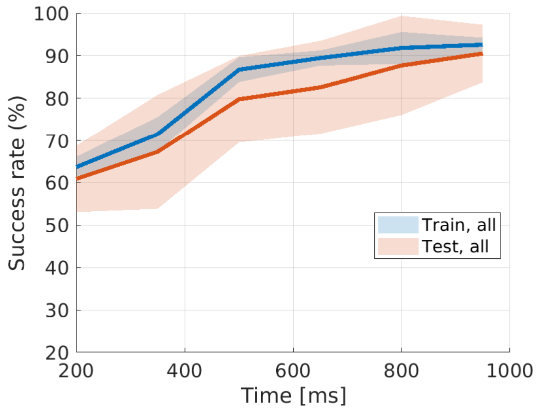


(b)

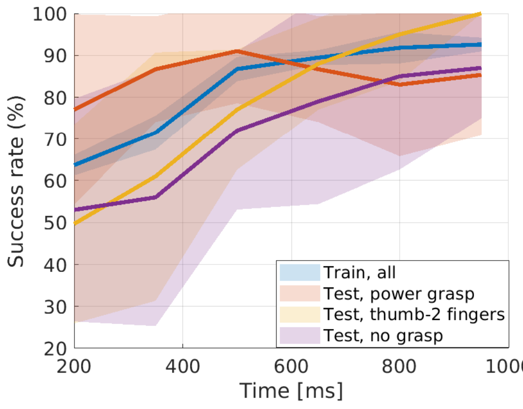
Figure 3.10: (a) represents the train and test accuracy with training one C-SVM classifier for all the time windows, tested through all the time windows and (b) represents the same accuracy only per grasp type.

The train and test curves are nearly identical until 600 ms after the onset of the trial. Then the test curve slightly goes under the train one to finally reach a score of 86.3%. It can be observed on (b) that the C-SVM classifier also had troubles identifying the no-grasp trials.

Figure 3.11 illustrates the results of implementing one C-SVM classifier per time window. A global increasing trend is observed compared with the ESN and LDA results which may be more intuitive. Moreover, (b) shows that the no-grasp motions remained a problem.



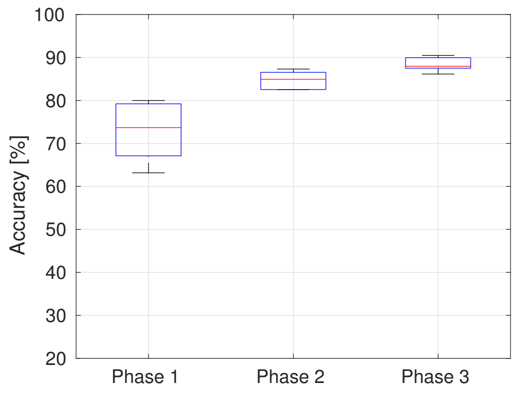
(a)



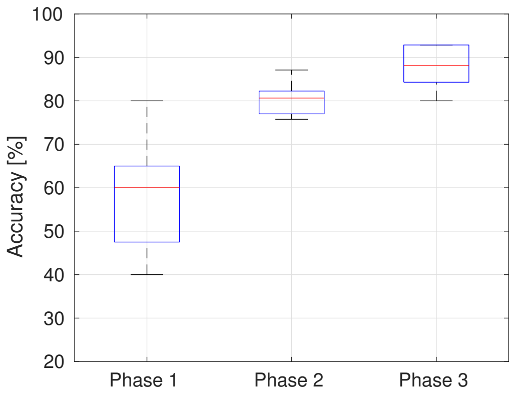
(b)

Figure 3.11: (a) represents the train and test accuracy with training one C-SVM classifier per time windows and (b) represents the same accuracy only per grasp type.

Finally, the 3-phases analysis was performed for one C-SVM classifier per phase and the results are illustrated on Figure 3.12 and Table 4. As in the single SVM approach, the train and test accuracy are close to finally reach a mean score of 88.46% and 87.90% in the third phase for the train and test set respectively.



(a)



(b)

Figure 3.12: (a) represents the obtained train accuracy by training one C-SVM classifier per phase and (b) represents the same analysis only on the test data. The median is represented with a red line and the bottom and top edges of the boxes represent the 25th and 75th percentile respectively.

	Phase 1	Phase 2	Phase 3
Train mean accuracy [%]	72.84	84.72	88.46
Train standard deviation [%]	7.12	2.16	1.70
Test mean accuracy [%]	58.00	80.31	87.90
Test standard deviation [%]	14.83	4.34	5.39

Table 4: Summary of the main results for the 3 phases approach implemented with C-SVM classifiers

3.5 GMM Classification

GMM was also implemented using the LDA embedded space for the same reasons mentioned previously. The hyperparameters to be set by the user was only the number of gaussian components per class. Such a value could be quickly assessed visually. Indeed, looking at Figure 3.5, one can observe that to model each classes, 1 or 2 gaussian components should be sufficient. Moreover, selecting a low number of gaussian components would prevent overfitting. The type of covariance matrix was chosen to be full as the different trend of the three classes were not aligned to the axis. Figure 3.13 illustrates the results of fitting 2 gaussians components per class using data from all the time windows. Again, as training and testing points were taken randomly, the points may vary from the ones depicted on Figure 3.5.

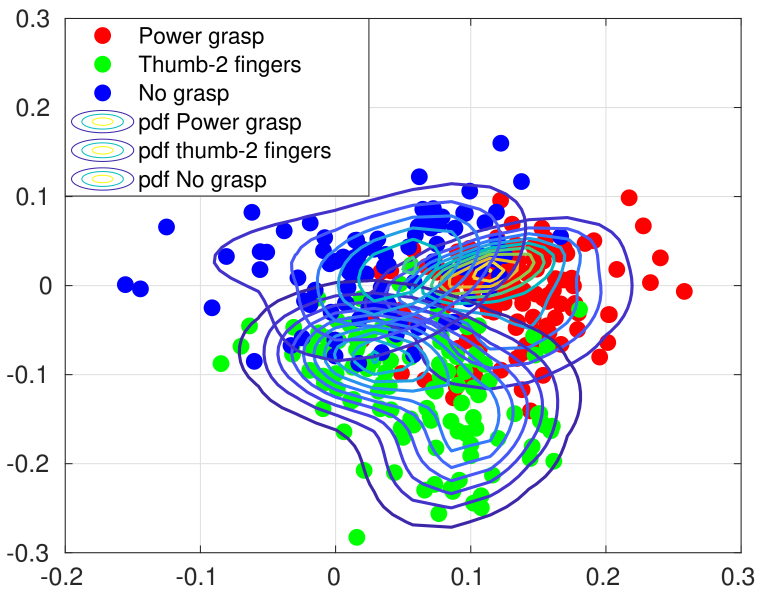
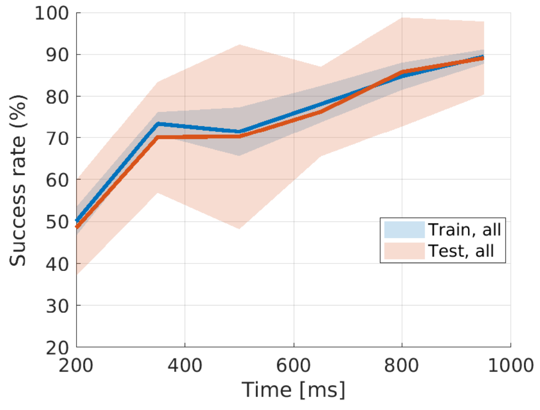
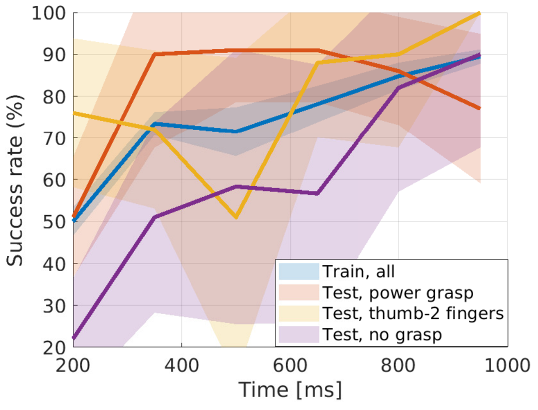


Figure 3.13: Results of fitting 2 gaussian components per class on the training data

The resulting classification boundaries and confidence matrices can be observed on Annex A.7 and A.8. A global train and test accuracy of 79% and 73% respectively was performed. The results of the same classifier only tested through all the time windows is illustrated on Figure 3.14. Like in the C-SVM case, the train and test curves are nearly identical, ending with a test accuracy of 89.4% with a variance of 7.6%.



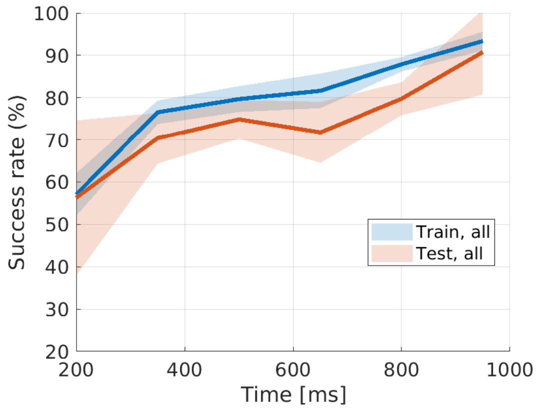
(a)



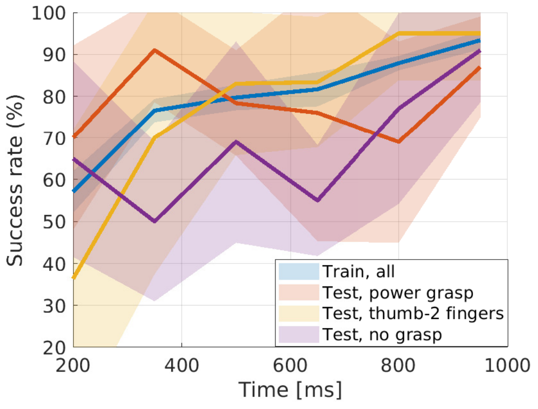
(b)

Figure 3.14: (a) represents the train and test accuracy with training a single 2-components GMM classifier for all the time windows, tested through all the time windows and (b) represents the same accuracy only per grasp type.

The results of implementing a 1-component GMM classifier per time window is illustrated on Figure 3.15. The global trend was again an increasing behavior as with C-SVM. For this approach, only one component per class was implemented. Implementing more components per class resulted on ill-conditioned covariance matrix which prevented their inverse computation. Considering that the number of datapoints per time window was only a fraction of the total number of datapoints, one could think that implementing only one component per class is sufficient as fitting a very small number of datapoints with 2 gaussian components may cause overfitting. Moreover, implementing several gaussian components would result in optimizing more parameters, which may become impossible if too few datapoints.



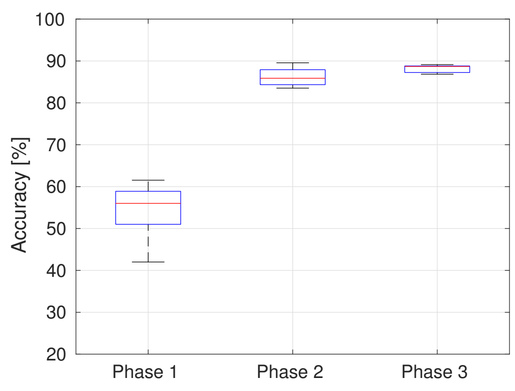
(a)



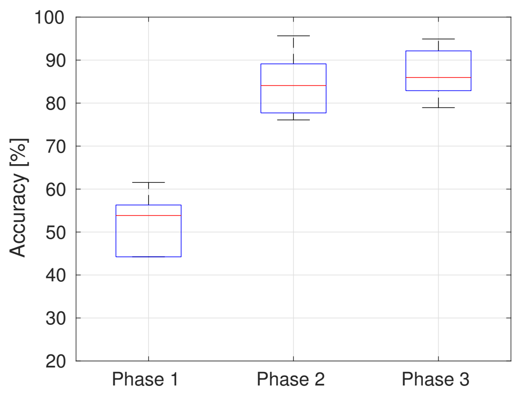
(b)

Figure 3.15: (a) represents the train and test accuracy with training a single 1-component GMM classifier per time windows and (b) represents the same accuracy only per grasp type.

The first two approaches on GMM again confirmed that the toughest trials to classify were the no-grasp motions. Finally, the 3-phases analysis was performed for one GMM classifier per phase and the results are illustrated on Figure 3.16 and Table 5. The general trend is similar to the one observed with the same approach but using a C-SVM classifier. The train and test accuracy were close, resulting in a final train and test accuracy of 88.15% and 87.05% respectively on the third phase.



(a)



(b)

Figure 3.16: (a) represents the obtained train accuracy by training a 1-component GMM classifier per phase and (b) represents the same analysis only on the test data. The median is represented with a red line and the bottom and top edges of the boxes represent the 25th and 75th percentile respectively.

	Phase 1	Phase 2	Phase 3
Train mean accuracy [%]	54.31	86.18	88.15
Train standard deviation [%]	7.42	2.37	0.98
Test mean accuracy [%]	47.83	84.21	87.05
Test standard deviation [%]	18.43	7.74	6.21

Table 5: Summary of the main results for the 3 phases approach implemented with 1-component GMM classifiers

4 Discussion

To be able to compute smooth trajectories, the grasping decision should be sent to the controller as close as possible to the onset of the motion. Thus, the most important part of the accuracy curves were the first time windows. The goal being to perform robust predictions already at the beginning of the motion. Observing the results of the three approaches, it was easily noticeable that the best performing approach was the second one being to train one classifier per time window. Looking at Figure 3.3 for example, impressive results were obtained with the ESN classifier as the test accuracy barely went below 80%. Indeed, the first approach performed decent accuracies (looking at Figures 3.2, 3.6, 3.10, 3.14) but the time needed to reach a test accuracy of 80% was generally from 600 to 800 ms after the onset of the trial which is below the average performances of the second approach. Considering the three phases analysis, there was a general increasing trend through the three phases but the early predictions never surpassed the results of the second approach. Indeed, looking at Figure 3.16 or 3.12, the first phase performed fluctuating test accuracies, with a mean close to 60%. Hence, from this point, the discussion will be centered on the second approach.

Generally, all the different test accuracies showed a non-negligible variance through the 5-folds cross-validation. Indeed, looking at Figure 3.5, one can observe that the data is easily separable on the edges of the three classes, but the task is way harder in the center of the blob. Even if the classes were balanced through the different folds, as the data is still taken randomly, the classification difficulty may fluctuate a lot between the different folds. Taking datapoints on the extremity of each classes resulted in an easier classification task compared to taking points inside the center blob. However, observing an overall trend being that the accuracy increased with time, points on the edges of each class may come from late time windows, compared to points in the center of the figure that may come from early time windows. Moreover, looking at figures 3.6, 3.7, 3.10, 3.11, 3.14 and 3.15, each class separately also had a high variance through the 5-folds. As the total amount of trials was 64, this means that there was approximately 13 trials per fold. Considering the balancing of the classes within the folds, there was 4 to 5 trials per grasp type per fold. Hence, misclassifying 1 trial out of 4, for example, already results in 75% accuracy for the given grasp type. Thus, having such a small amount of data per fold could explain the high fluctuation of the test accuracy within the different grasp types.

Furthermore, all classifiers seemed to have trouble identifying the reach-not-to-grasp motions. Both reach-to-grasp motions were characterized by finger placement patterns that should stay the same between the different attempts to grasp the object. However, the no-grasp motions did not embed such patterns, hence the classification of such grasps may be more difficult. Thus, when the subject performed a reach-not-to-grasp trial and its hand configuration were sufficiently close the the configuration of a reach-to-grasp motion, the classifiers may have taken it as a reach-to-grasp trial. Such a hypothesis relies on the classifiers mainly identifying the finger placement to deduce the grasping intention.

Considering the 4 different implemented classifiers, two types of results were observed.
The LDA and ESN classifiers both induced the same type of results being relatively high at the first time window, but with a decreasing behavior until 500 ms. From this point, the prediction accuracy only increased with time. This type of behavior may be explained by the finger distributions not being significantly different between the grasp types before 500ms. To assess this hypothesis, the aperture of the hand was computed continuously through all the trials and its maximum was located in time. Table 6 illustrates the obtained results. Indeed, the mean time to reach the highest aperture of the hand is generally around 500 ms for the two reach-to-grasp type of motions. Hence it could be a valid justification of the increasing trend in the accuracy starting around 500 ms after the onset of the trial. This may also confirm the expressed hypothesis on the classifiers having troubles identifying the reach-not-to-grasp motions due to an ambiguity in the finger configurations with the reach-to-grasp motions, which relies on the classifier mainly analyzing the finger placement to deduce the grasping intention.
The same table also explains why the test accuracy was being generally low during the first phase as it ended on average after 283.38 s which is quite early in the reaching motion.

	Power Grasp	Thumb-2 fingers	No grasp
Mean threshold phase 1 to 2 [ms]	273.9	249.6	326.64
Mean time to get max aperture [ms]	506.8	443.9	733.3

Table 6: Time taken to reach the highest aperture of the hand against the different grasp types. As a comparison, the mean time to transfer from phase 1 to phase 2 of the third approach is also illustrated

The second type of classifiers was the case of both SVM and GMM classifiers. The global trend was an increasing accuracy with time from the beginning until the end of the trial. Hence, these two algorithms seemed to overpass the problem mentioned for the classifier type explained previously. This would be characterized as a strength. However, even if the global performances of C-SVM and GMM were decent, they seemed to be surpassed by the ESN classifier.

Using an ESN classifier, if one defines a robust decision being a prediction with more than 80% accuracy, the time needed to predict accurately a grasp type would be from 500 ms after the onset of the trial as illustrated on Figure 3.3. Moreover, from that time, the prediction accuracy only grew, and did not decrease compared to the trend before 500 ms. For the LDA classifier, the results were not very satisfying. Using the same definition of a robust decision, one should wait until around 800 ms after the onset of the trial as illustrated on Figure 3.7. For the C-SVM classifier, the results were presented on Figure 3.11. A test accuracy of 80% was reached after 600 ms. At the end of the trial, meaning after 950 ms, an accuracy close to 90% was observed. Finally, for the GMM classifier, the global trend was similar to the C-SVM classifier, only slower. An accuracy close to 90% was observed at the last time windows while an accuracy of 80% was ensured after 800 ms as illustrated on Figure 3.15. These different values are summarized on Table 7.

	ESN	LDA	SVM	GMM
Time needed to ensure 80% test accuracy [ms]	500	800	600	700

Table 7: Summary of the different classifier performances

Hence, the best performing classifier for this application was the Echo State Network. Indeed, it could nearly assure a prediction accuracy above 80% during the entire motion. However, this algorithm was very slow in terms of training time.

Talking about training complexity, the slower algorithm was by far the Echo State Network. Indeed, training neural networks can take a very long time. As the data was pushed as input without extracting features, it resulted in very long training time. Considering the three other classifiers, the training time could always be set as negligible. Nonetheless, for the studied application, the training of the algorithms would be most likely performed *off-line*. This means that the training complexity do not plays a big role in the critical complexity to implement such a method. The test complexity is, however, an important question as the prediction must be made in real time. Hence long computation cannot be accepted. To simulate the *on-line* applications, the time windows were set to 200ms with 50ms overlap. This means that the classifier should output a prediction before the upcoming of the next time window. Considering the overlap, the classifier should make its prediction within 150ms after the reception of the data.

The final prediction of the algorithm would be made using a *majority vote* [3]. This means that, after a given number of predictions outputted by the algorithm, the final prediction would be the grasp type with the greatest number of occurrence within the stream of outputs. Fixing the number of outputs to 3 and assuming that the constraint on the test computational complexity is respected, the final prediction of the grasp type would be outputted by the algorithm after 500 ms. Implementing this policy with a trained ESN, the three outputs being accurate with a probability above 80%, would be the optimal solution considering the resulting accuracy of the different classifiers compared in this report.

An important question is, when can one assure that the prediction is confident enough ? What accuracy is necessary to perform a robust prediction ? The answer to this question would depend on the application. For a daily task, meaning grasping simple object like a chair, a glass, etc.. the most important aspect is to differentiate between reach-to-grasp and reach-not-to-grasp motions. However, for more critical applications like in medicine, such a differentiation may become crucial.

Overall the most powerful algorithm appeared to be the Echo State Network. Indeed, despite its high training complexity, it beats the other three classifiers in terms of overall accuracy through time. Figure 4.1 illustrates the dominance of this classifier. The mean length of the trials was around 1.2 to 1.4 s, meaning that the prediction of the grasping intention would be outputted approximately after 1/3 of the reaching movement.

One should still consider that implementing this approach could induce problems for an *on-line* utilization. Indeed, it relies on a robust detection of the onset of the motions. Having delays with this definition would shift all the different time windows, potentially resulting in poor classifiers accuracy. To reduce the impact of a noisy definition of the onset of the motion, the best approach would be the third one. Using this approach, the best classifier would also be the ESN considering the Figures 3.4, 3.8, 3.12 and 3.16.

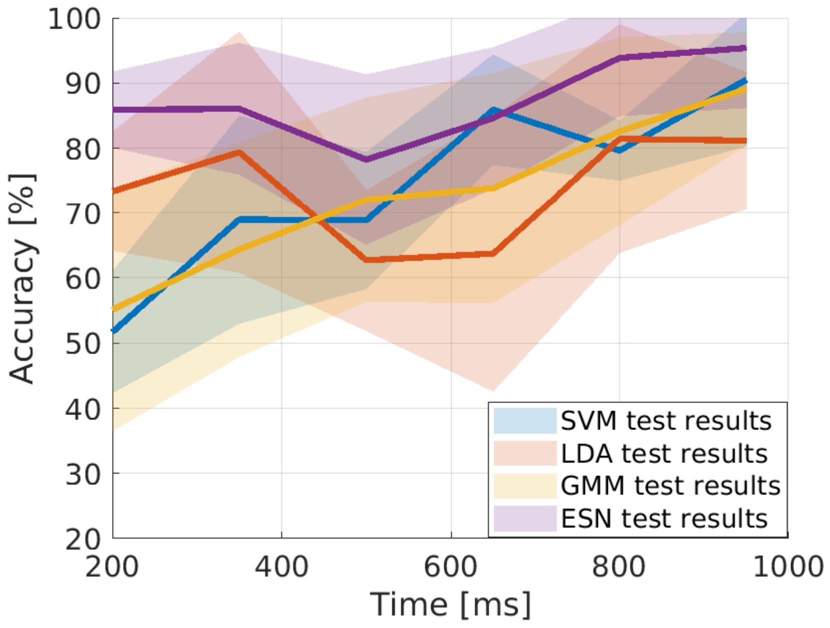
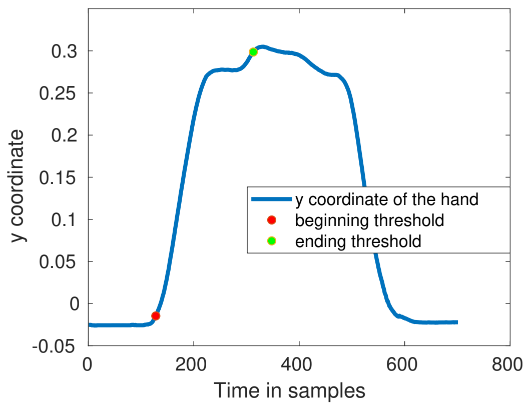


Figure 4.1: Figure containing all the test accuracies of the 4 different classifiers. ESN surpasses overall the three other classifiers.

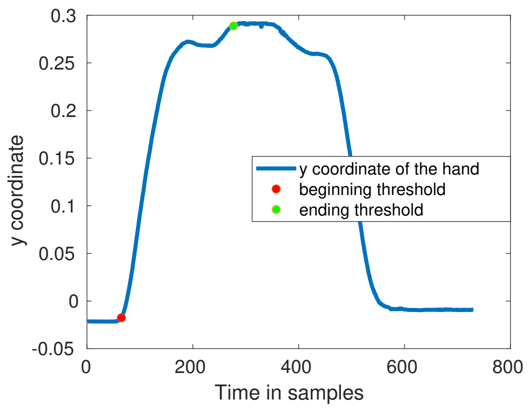
5 Conclusion

For this project, a concrete analysis of a given problem was performed from the data measurement to the results and comparison of different classifiers. Three different approaches were implemented to analyze the data. The first one being to segment the signal into time windows of constant length and train either one unique classifier for all the time windows or one classifier per time window whereas the last method was considering the trial as mainly 3 stages being the acceleration of the hand, the deceleration of the hand, then the grasping.
It was observed that such a task is not trivial but showed some promising results for the future. Indeed, being able to detect accurately the grasp type from the early stages of the reaching motion could be one more step until the perfect embedding of neuroprosthetic arms. It was shown that a robust prediction could be made with the best classifier being an Echo State Network after 500 ms, which may be enough to allow the controller to compute smooth control to securely perform the grasp movements.

A Figures



(a)



(b)

Figure A.1: Illustration of the trial definition on a power-grasp (a) and a thumb-2 fingers (b) motion. The used axis was chosen to be y as the motion pointed towards the positive direction of this axis.

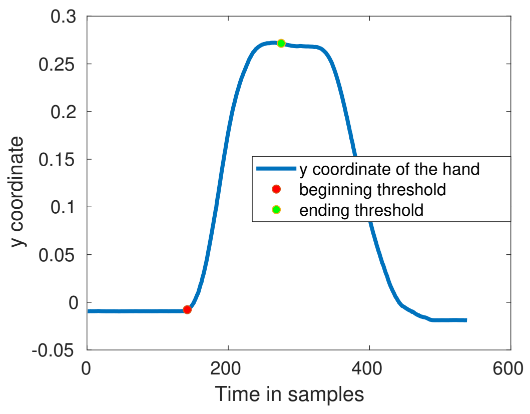
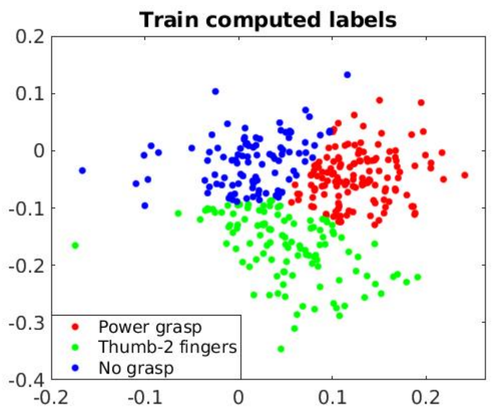


Figure A.2: Illustration of the trial definition on a no-grasp motion. The used axis was chosen to be y as the motion pointed towards the positive direction of this axis.



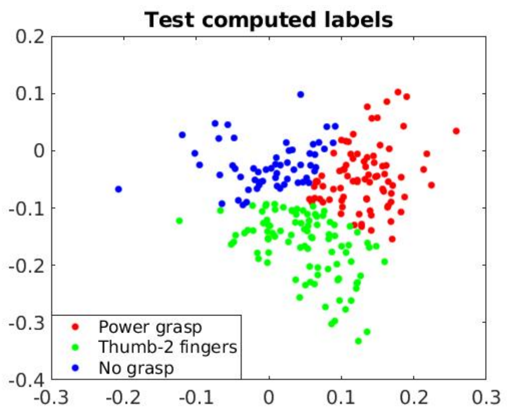
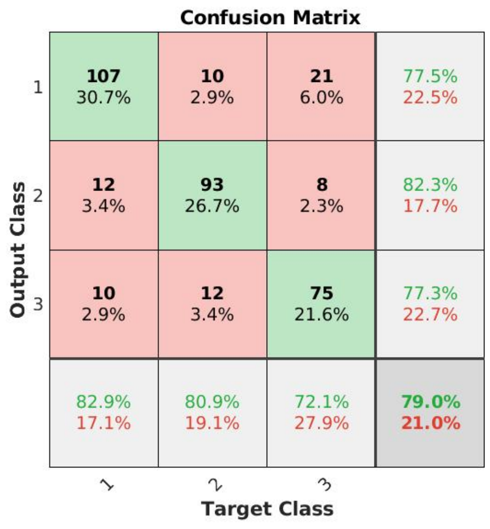
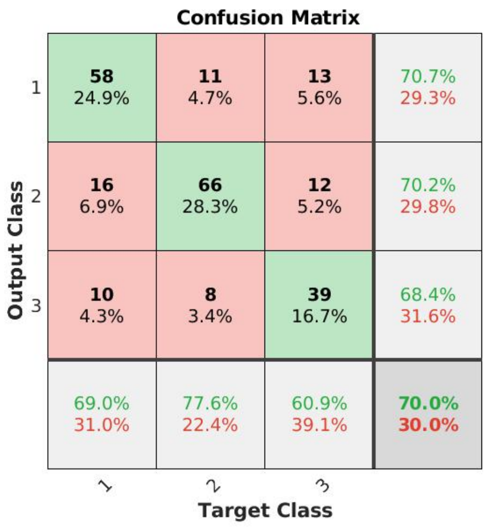


Figure A.3: Train and Test computed classification using a single LDA classifier

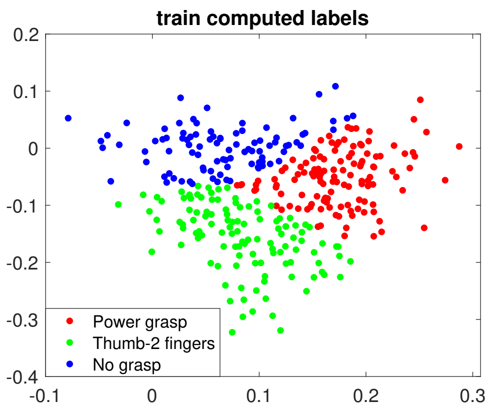


(a)



(b)

Figure A.4: train (a) and test (b) confusion matrix for one single LDA classifier trained on all the time windows. The interesting part is the last line, there is no clear class acting as an outlier as the accuracy per class remains quite stable.



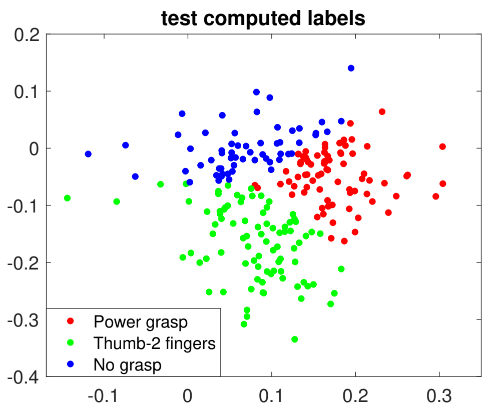
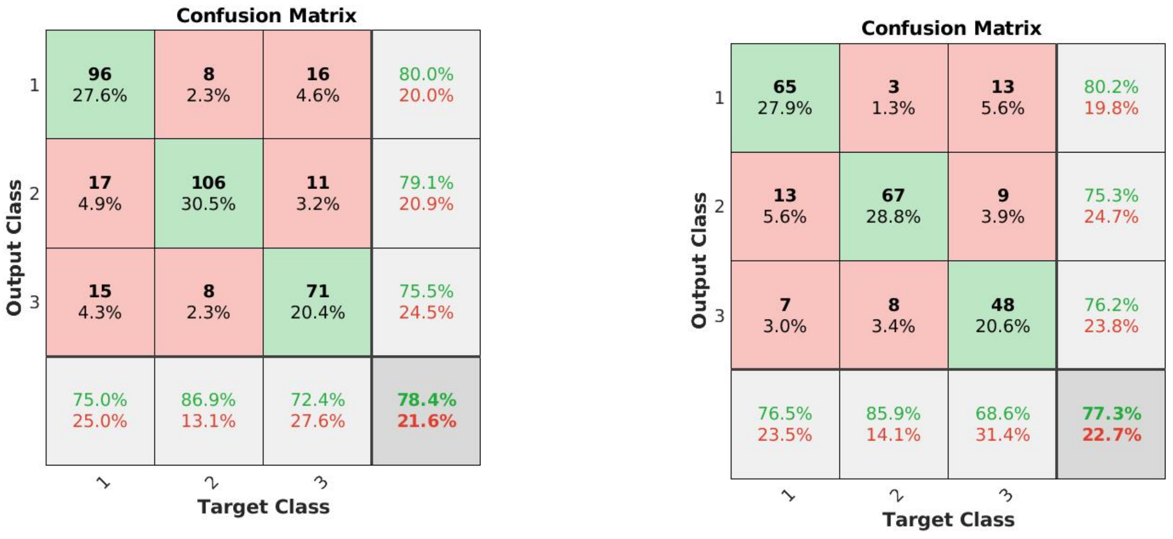


Figure A.5: Train and Test computed classification using a single C-SVM classifier



(a) *(b)*

Figure A.6: Train (a) and test (b) confusion matrix for a single C-SVM classifier. The same results are observed as with the LDA confusion matrix : there is no clear outlying class.

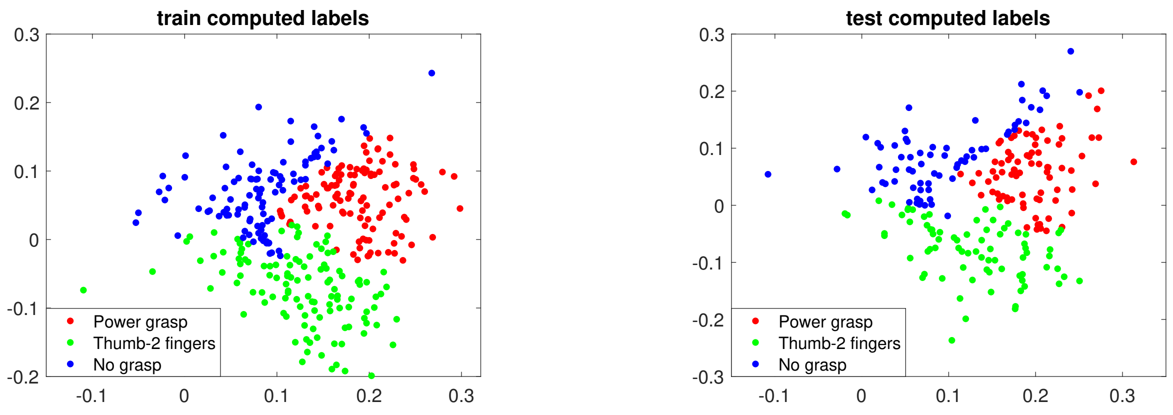
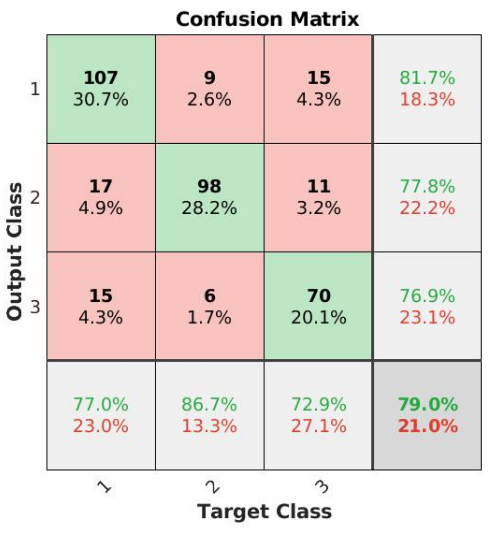
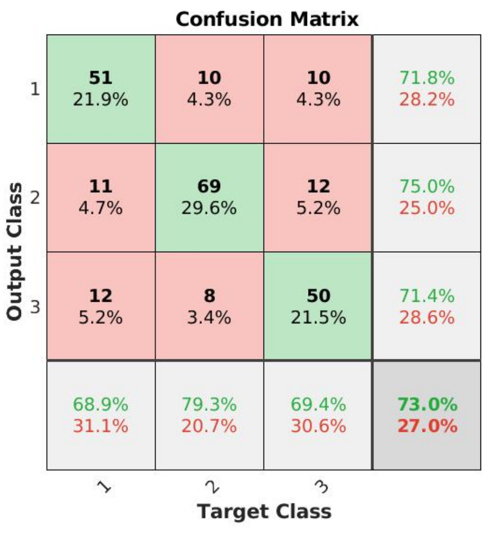


Figure A.7: Train and Test computed classification using a single 2-components GMM classifier



(a)



(b)

Figure A.8: Train (a) and test (b) confusion matrix for a 2-components GMM classifier. The same results are observed as with the LDA confusion matrix : there is no clear outlying class.

References

[1] I.Batzianoulis, S.El-Khoury, E.Pirondini, M.Coscia, S.Micera, A.Billard *EMG-based decoding of grasp gestures in reaching-to-grasping motions.* 2017 Elsevier B.V

[2] N.Nazmi, M.Azizi, A.Rahman, S.Yamamoto, S.Ahmad, B.Malarvili, S.Mazlan, H.Zamzuri *Assessment on Stationarity of EMG Signals with Different Windows Size During Isotonic Contractions.* 2017

[3] K. Englehart, B. Hudgins *A robust, real-time control scheme for multifunction myoelectric control.* 2003

[4] E. Scheme, K. Biron, K. Englehart *Improving myoelectric pattern recognition positional robustness using advanced training protocols.* 2011

[5] L.Smith, L.Hargrove, B.Loc, T.Kuiken *Determining the Optimal Window Length for Pattern Recognition-Based Myoelectric Control: Balancing the Competing Effects of Classification Error and Controller Delay.* 2010

[6] M. Lukoševičius *A Practical Guide to Applying Echo State Networks.* Springer, 2012.

[7] Chang, Chih-Chung and Lin, Chih-Jen *LIBSVM: A library for support vector machines* ACM Transactions on Intelligent Systems and Technology, 2011

[8] Masci J., Meier U., Cireşan D., Schmidhuber J. *Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction.* Springer, Berlin, Heidelberg, 2011

[9] Pedro Domingos *A Few Useful Things to Know about Machine Learning* University of Washington

Link to the GitHub repository containing the code used for this project:
https://github.com/AntoineWeber/Project_LASA.git