EMG-Based Control of a Simulated Robot Arm

Scientific work within the practical course
from the Department of Electrical and Computer Engineering at the
Technical University of Munich.

Supervised by Univ.-Prof. Dr.-Ing. Sandra Hirche
Dr. Yi Ren
Chair of Information-Oriented Control

Submitted by B.Sc. Oussema Dhaouadi
Leitershofer Str. 75a
86157 Augsburg
oussema.dhaouadi@tum.de

Submitted on Munich, 13.10.2019

Abstract

In recent years, Deep Learning techniques have played an important role in the computer vision and the speech recognition fields due to its high performance. Given the recent progress in machine learning, this work illustrates the various traditional and deep-learning enhanced methodologies for classifying the electromyography signals encoding the biceps and the triceps muscle activities. A comparison study is also given to show the performance of using the so-called spectrogram analysis that describes the time and frequency domains in a correlated fashion. Further, different segmentation methods of the signals and various neural network architectures are presented. While using a traditional machine learning method achieved only 38.28% on the validation set, the spectrogram-based convolutional neural network outperforms all other techniques with an accuracy of 89.84%. The best trained models are used to predict gestures to control a simulated robot arm.

Contents

Chapter 1

Introduction

1.1 Motivation

Muscles are fundamental elements to perform movements in the human body. Movement and locomotion are initiated by the electrophysiological excitation of a group of motor units. The electrophysiological signal generated during excitation/contraction can be detected and used to study muscle function and is known as electromyography (EMG). EMG signals are complex and noisy biomedical signals that can be recorded at the muscle level. It not only provides the information on the condition of the muscles (contraction or relaxation) but also encodes the force generated by a muscle, which is controlled by the central nervous system. Its quality depends therefore on the physiological and the anatomical states of the human body.
A wide range of applications that are based on EMG signals have been developed in clinical health processes, such as analyzing and determining disorders, and in the robotic field as human-machine interface devices that are able to control an intelligent system.
This work presents a summary of some methods used to classify these signals for the sake of controlling a simulated robot arm which can be implemented on a real robotic system.

1.2 Problem Statement

The main objective of this work is to explore different advanced methods for detection, decomposition, processing, pattern extraction and classification in order to build an EMG-based robotic control. Analogously to an audio recognition systems, the developed system senses the activity of the muscle in form of EMG signals and outputs a statement on the type of the movement that has been performed. Starting by developing preprocessing units is indispensable to reduce the noise and the complexity of the signals and to help the classifier finding the rule of decision. After trying different preprocessing steps and after designing different classifiers, the best model will be concluded and used for the robot arm control system.

1.3 Contributions

In the work, the following steps will be presented:

- EMG signals segmentation techniques.
- Time Domain (TD) and TD/FD features extraction techniques.
- Conventional pattern recognition systems: Support Vector Machines (SVMs).
- Deep Learning enhanced classification systems: Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) and Long Short Time Memory Neural Networks (LSTMs).
- A 3 DoFs robot arm model and its forward and inverse kinematics.

This work consists of four parts. The first part is dedicated to the methodology adopted in this work including the preprocessing engine followed by the pattern extraction techniques and the recognition module. Results of the various techniques are shown in the second part. The third part describes the robot arm simulator and finally, the last part infers conclusions.

Chapter 2

Approach and Methodology

This chapter briefly describes the overall pipeline of the system. Afterwards, it introduces the different processing components and discusses the various methods used for the classification task. Finally, it presents the robot arm simulation framework.

2.1 System Overview

In order to assess and control a robotic environment, a data-set must be formed. These digital signals have to be processed to extract the desired information that will be used to recognize the muscle activities and the corresponding movement. The predicted movement class will be sent as simple command to the simulation module. The following sections introduce the overall system architecture and the technical aspects behind the processing modules of the designed system. 2.1 shows the pipeline of the system and the different techniques used to achieve the goals of this work.

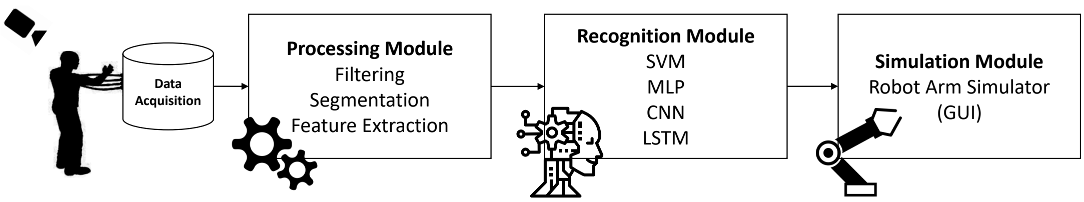

Figure 2.1: Overall System

2.2 Data Acquisition

This work focuses on measuring the biceps and the triceps muscles activities in order to learn their behaviors when performing simple gestures.
First of all, the movements are chosen by moving the human arm and trying to sense

the muscles that are active. After selecting a set of six movement classes shown in Figure 2.2, the biceps and the triceps were the best candidates that may accurately encode the movement. The data recording is divided into three sessions, where

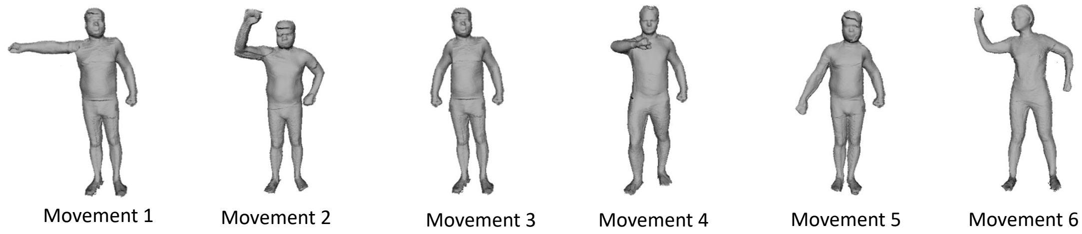

Figure 2.2: Ground truth movements

in every session two movements are repeatedly and alternately performed. Every movement is recorded within a segment of 3 seconds using the Shimmer hardware platform. The camera system is set up to tack the movement of the arm and will be used to label the movements. It consists of 3 calibrated cameras and provides 3D trajectories of a marker sticked to the tip of the hand. A trigger button is used to start and stop the camera recording system and to synchronize it with the EMG signals.

2.3 Processing Module

Three steps are performed in the processing module. A filter is applied to increase the singal-to-noise ratio, a segmentation of the signals is performed and finally a feature extraction module generates the data-sets to train, validate and test the classifiers.

2.3.1 Filtering

The raw data is imported from the collected data set, filtered using a 5th order low pass with a cutoff frequency of 55Hz and a notch filter that rejects unwanted line frequency of 50Hz.
The recorded signals of length N are described by their discrete form as $s[n] = s(nT)$, where T is the sampling period. A statistical normalization defined as $\hat{s}[n] = \frac{s[n]-E[s[n]]}{Var[s[n]}$ is performed to remove the bias and normalize the statistical variance, where the estimation of the expected value $E[s[n]]$ of the discrete random sequence $s[n]$ is given by the empirical mean $\tilde{E}[s[n]] = \frac{1}{N}\sum_{n=0}^{N} s[n]$ and the estimation of the variance $Var[s[n]]$ is given by $\tilde{Var}[s[n]] = \frac{1}{N}\sum_{n=0}^{N}(s[n] - \tilde{E}[s[n]])^2$. Finally a normalization that maps the signal to the range $[-1, 1]$ is applied.

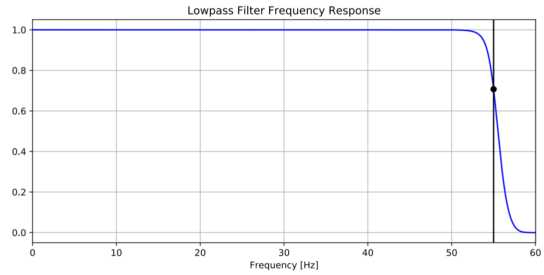

Figure 2.3: Low pass with a cutoff at 55Hz

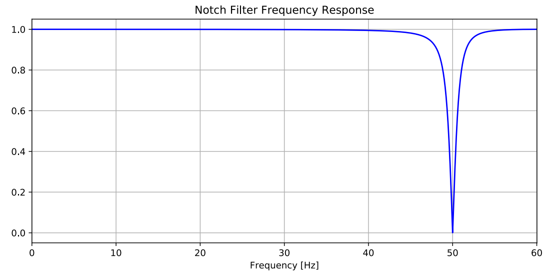

Figure 2.4: Notch filter at 50 Hz

2.3.2 Segmentation

After getting the signals ready, the segmentation sub-module divides at a first level the signals into segments of 3 seconds, which corresponds to the duration of a sample. Then, every signal is divided into 600ms[OH08, CRMA+13] independent sub-segments (without overlapping). At the same time the camera data are segmented by setting a threshold indicating the beginning and the end of a movement. The output of this part are 2 labeled data-sets, which will be called "full segments data-set" and "partial segments data-set", indicating samples of full 3 seconds and a set of 600ms long independent signals respectively. According to [EFM17], 600ms segments are enough for a real time control.

2.3.3 Features Extraction

As a first step, a visualization of the data should provide a first idea about the complexity of the features. As shown in Figure 2.5, extracting a recognizable feature for the very similar movements 3 and 5 is hard. This motivates the development of a sophisticated machine learning algorithm that solves the classification problem. Since the data is high dimensional, it is indispensable to project it onto a low dimensional feature space, allowing the machine learning algorithm to accurately separating each movement class. For that, the number of dimensions is reduced from 360 to 42 consisting of the following time domain (TD) features: The Mean Absolute Value (MAV), the Root Mean Square (RMS), the Variance (Var), the Slope Sign Change (SSC), the Maximum Amplitude (MaxAmp) and the Minimum Amplitude (MinAmp) [TAZ+17]. The representation of each sample in the TD features space is collected in a new data-set called "TD features data-set".
Moreover, ignoring the frequency domain information may increase the classification error. Therefore, the so-called spectrograms are extracted from each full segment resulting a 2D data-set, which will be called "sepectrograms data-set" [ZJCT17]. The idea behind that is to extract the Fourier Spectrum of each sub-segments of the data sample. These samples are overlapped in order to provide the time correlation information. The generated spectrogram has the shape of 15x15 which is lower

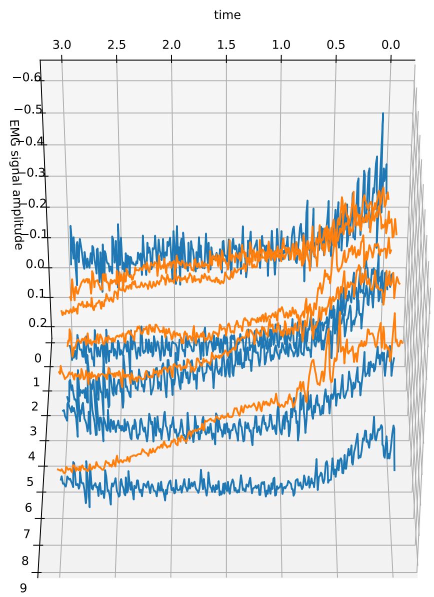

Figure 2.5: 3D representation of the extracted full segments for movement classes 3 and 5.

than the original full segment dimension, but combines explicitly the TD and the Frequency Domain (FD) spaces.

2.4 Recognition Module

Four data-sets, namely the "full segments data-set", the "partial segments data-set", the "TD features data-set" and the "spectrograms data-set" are available for the classifiers. As the type of the features is important for the accuracy of the classifier, all models are trained on these 4 data-sets. This allows a comparison of the amount of information needed to achieve a reasonable accuracy. All models were trained offline, i.e. the data-set is available on advance and no additional external signals are fed to the network while training. The data-sets are divided into 2 sets, namely the training set and the test set. As the name suggests, the training set is used to train the models, while the test set is an unseen set and used to evaluate the model. After shuffling the data, 20% of it is allocated for the test-set and should cover different movement classes. This module includes different python packages, well known in the machine-learning community, which are *keras*, *tensorflow*, *sklearn*, *numpy* and others.
For the sake of pattern identification, 4 models are compared. In the non-parametric model class, the Support Vector Machine is considered. In the parametric model class, 3 Neural Network models with increasing level complexity are discussed.

Support Vector Machine (SVM) The Support Vector Machine (SVM) separates the classes by finding a separating hyperplanes or surfaces. This is obtained by maximizing the margin between the subspaces of the different classes and by minimizing the number of outliers. Once the model is trained, a new sample will be classified by the means of its position in the feature space. Defining the type of the separative surface is possible by a high-dimensional function called kernel function. In this work, the linear, the Radial Basis Function (RBF), the polynomial and the sigmoid functions are considered. Hereby the training is done on the four data-sets, independently. As a first step, the Principle Component Analysis (PCA) model reduces the dimensions of the input-space [AK10]. Only two principle components are preserved, which conserves the most of the energy of the data and offers a better visualization of distribution of the projected data.

Multilayer Perceptron (MLP) Neural Network (NN) models are very sophisticated function approximators. In particular, a Multilayer Perceptron as classifier, also called Fully Connected Network, tries to approximate the posterior probabilities $p(w_k|\mathbf{x})$ for every class w_k, where $\mathbf{x}$ is the input signal by minimizing the cross-entropy loss. In addition to the different data types, different network architectures are trained and the best model achieving the highest accuracy is used afterwards for the comparison between the different NN models, whereby the tuned parameters are the number of neurons, the depth of the network, the dropout rate and the batch size.

Convolutional Neural Network The Convolutional Neural Network (CNN) guides the flow of information in the network by a specific neuron connections in a way that they perform the convolution operation. After training the network, features are extracted by learning the filters. Since selecting predefined features to be fed to the classifier may not be optimal, the CNN is designed to extract and learn common features of the data. It is followed by a simple fully connected network allowing the mapping of features to classes.
Similar to the previous models, this model is fine-tuned by varying the number of filters of the convolutional layers, the depth of the network, the kernel size, the pooling size, the dropout rate used in the fully connected layer and the batch size. Note that the training on the "TD features data-set" is not relevant for the CNN.

Long Short-Term Memory (LSTM) The Long Short-Term Memory(LSTM) learns to store information over extended time of intervals and tries to find the correlation in time. This model is well accurate for non-stationary signals, where different time slots should be considered. The storage of information is assured by its closed-loops making the network dynamic. Since time is relevant for the EMG signals, the LSTM is introduced. The same experiment is performed for the LSTM as the other network. The hyperparameters have to be tuned and the best model has to be extracted for each of the data-set types.

2.5 Robot Arm Simulator

To test the performance of the designed classifiers on a robotic system, a robot arm simulator is considered. This should include following features:

- A simple robot arm model with at least 3 DoFs.
- A real-time controller by solving the forward and the inverse kinematics.
- Immediate response to the predicted human movement given as a command with predefined 3D target positions.

For simplicity, a Matlab Simulator was used. It is based on a 2D Matlab Kinematic model of a Puma robot located in the robotics lab of Walla University. Robot geometry uses the CAD2MATDEMO code in the Mathworks file exchange. Figure 3.14 shows the Graphical User Interface of the simulator. The User Interface Control

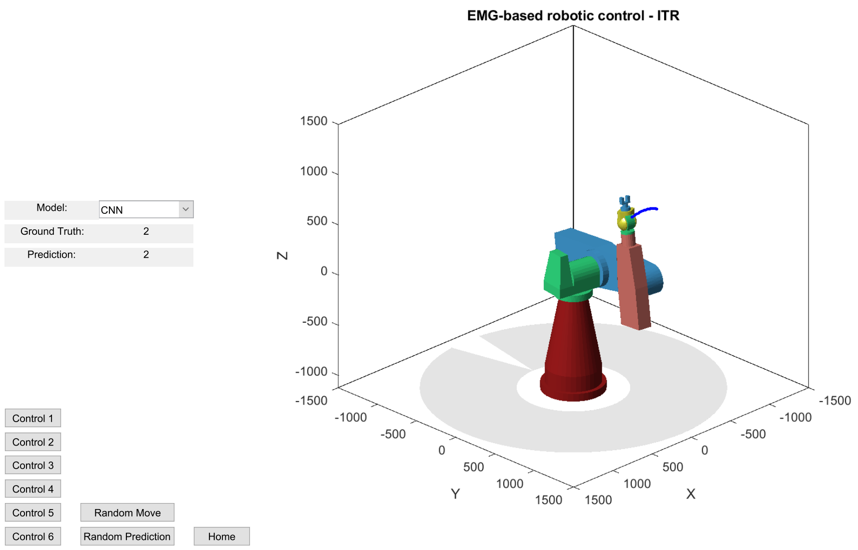

Figure 2.6: Graphical User Interface of the robot simulator (modified).

(uicontrol) block was added to enable the interaction of the user with the simulation. Moreover, by slightly changing some functions and adding more features, we adapted the simulator to our purposes. This framework presents 6-DoFs Robot Arm system, but only 3-DoFs are considered and are enough to reach every single point in the 3D space (points should belong to the spherical space that the arm can reach). After receiving the predicted movement of a sample from the test-set, the robot is able to solve the inverse kinematics by determining the joint positions for a given 3D-coordinates target position. The response of the robot should be similar to the human action.

Chapter 3

Results and Discussion

After describing the technical approaches and the functioning of every module in the system pipeline, the outputs of the different components are evaluated and discussed.

3.1 Generated Data

The data is generated after recording EMG singals and applying processing steps. This section provides an idea of the structure of the data by showing the output of each processing procedure.

3.1.1 Data Acquisition

Figures 3.1 and 3.2 show the recordings of the biceps and the triceps muscles activities with a sampling rate of 120Hz which means only 60Hz bandwidth is available. Around 110 samples per channel are recorded in .csv files and are then fed to the processing module.

3.1.2 Processing Module

The processing module starts by applying a filter, which reduces the noise from the raw signal and therefore improves the quality of the generated data-sets. The normalization step removes the unit dependency of the system and avoid the classifier to be trained on biased data. The result of both steps is shown in Figure 3.3. The segmentation step separates the movements to generate the "full segments data-set". These segments are than divided into small sub-segments to create the "partial segments data-set". In total the "full segments data-set" contains 636 samples, which is the result of row-wise stacking of all samples from different movements. Note that the first, the second and the third recording sessions respectively provide 102, 106 and 110 pair of samples, while the "partial segments data-set" has 3180 samples of length 72, i.e. a full sequence consists of $120Hz \cdot 3s = 360$ sampling points and $120Hz \cdot 600ms/1000ms = 72$ sampling points is the length a partial segment.

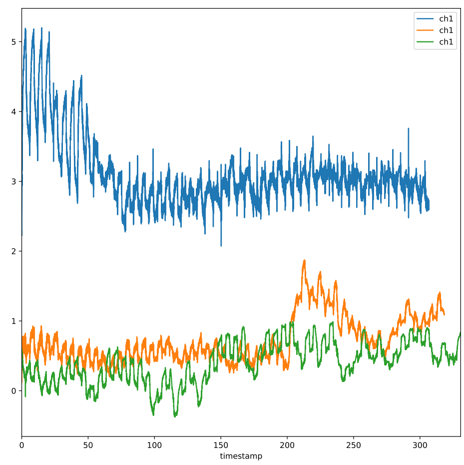

Figure 3.1: Recordings during the 3 sessions of biceps muscle activities

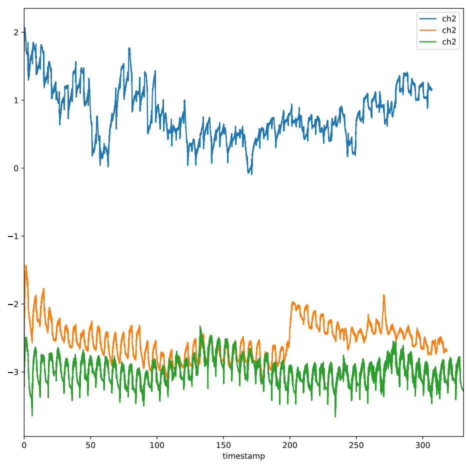

Figure 3.2: Recordings during the 3 sessions of triceps muscle activities

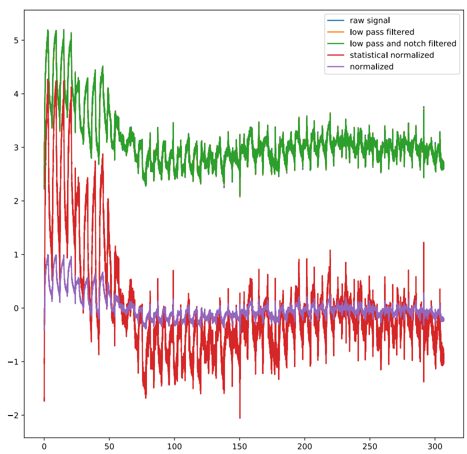

Figure 3.3: Biceps EMG signals after filtering and normalization.

The resulted samples from the segmentation procedure are presented in Figures 3.4 and 3.5. The last step consists of the features extraction. Some TD features are computed and each signal is represented by a single vector. All these vectors are stacked to create the "TD features" data-set. Furthermore, as montioned in the previous chapter, the "spectrograms" data-set is a sophisticated representation of the EMG signals. As depicted in Figures 3.6 and 3.7, although the two movements 3 and 5 are very similar, their corresponding spectrograms are slightly distinguishable.

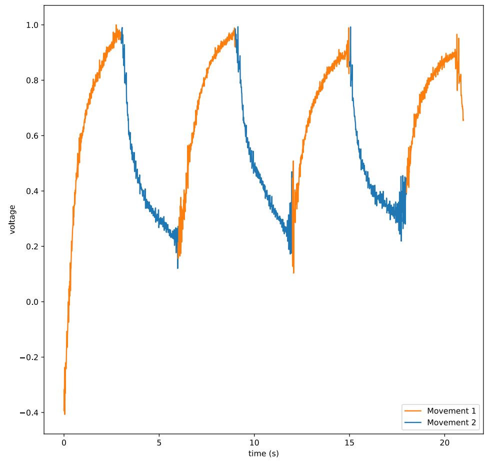

Figure 3.4: Labeled biceps EMG full samples recorded in the first session: movements 1 and 2

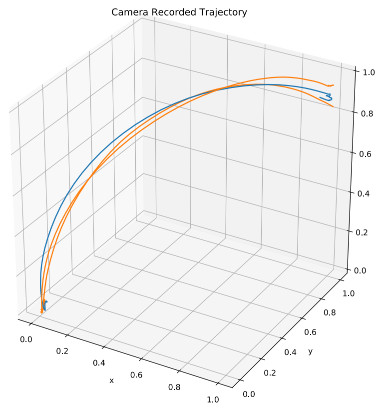

Figure 3.5: Labeled camera trajectories recorded in the first session: movements 1 and 2

Therefore, the classifier should be able to find common patterns of the samples of the same movements and dissimilarity pattern of the samples from different classes.

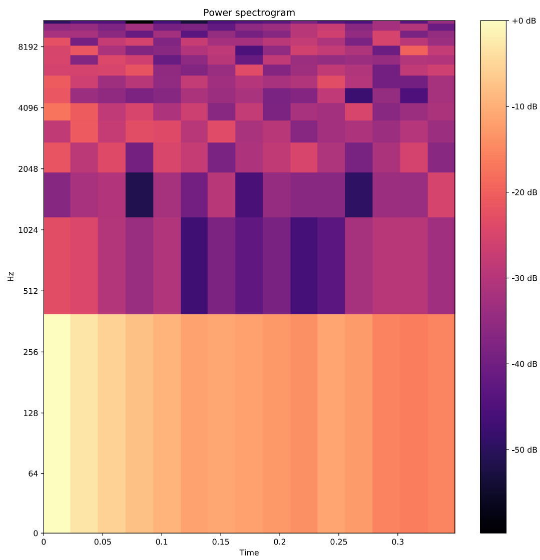

Figure 3.6: Example spectrogram of a sample from the class 3.

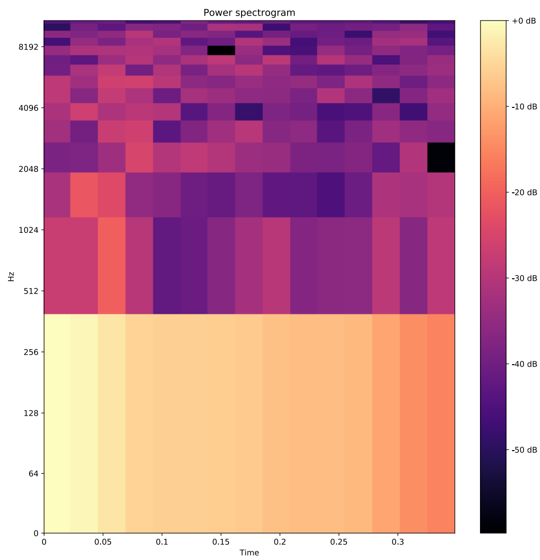

Figure 3.7: Example spectrogram of a sample from the class 6.

3.2 Recognition Module

This section shows the results of the classifiers and discuses the obtained results. The evaluation of the classifiers is performed on the validation set, which includes 20% of the total samples.

3.2.1 Evaluation of the Support Vector Machine Model

As mentioned in the previous chapter, the idea is to reduce the dimensionality of the TD feature space. Figure 3.8 shows the sufficiency of reducing the problem to a 2-dimensional space, as the highest variances of information belong to the 2 first-principle components and hence, considering the projected data does not lose much information. The accuracy obtained for this model is low on all data-sets. Since two

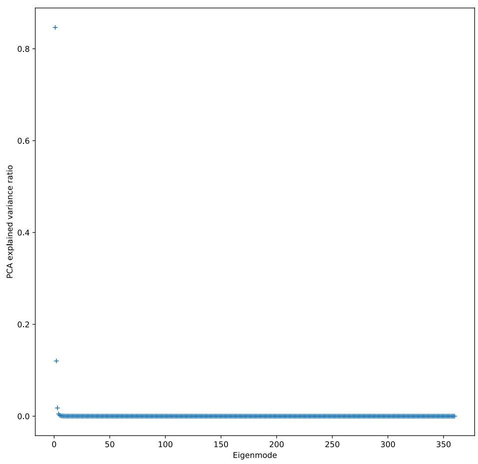

Figure 3.8: Full explained variance ratio (Energy as a function of the principle components) of the "full segments data-set".

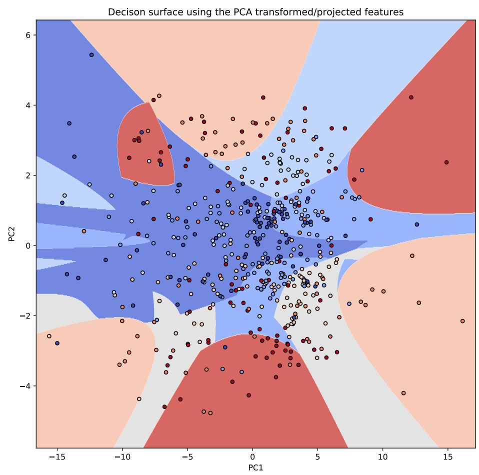

Figure 3.9: Decision boundries of the SVM with RBF kernel and using "full segments data-set".

hyperparameters are tuned, namely the data-set type and the kernel function, only the best accuracies for each data-set type are illustrated. For more results, please refer to the "results" folder attached to the source code.

Data-set	Kernel	Accuracy
Full segments	RBF	**38.28%**
Partial segments	Linear	21.85%
TD features	Linear	16.41%
Spectrograms	Sigmoid	17.96%

Table 3.1: SVM achieved accuracies.

As shown in Figure 3.9, the SVM is not able to find the hyperplanes sperating the subspaces. The difference in accuracy is that the ”full segments data-set” is not reduced and comprises more information than the others.

3.2.2 Evaluation of the Multilayer Perception Model

A Multilayer Perceptron network provides a high capacity of learning, since it has a high degree of freedom. The following table presents the network architectures with the best accuracies achieved on the different data-sets.

Data-set	Number of Neurons	Dropout	Batch Size	Accuracy
Full segments	256	0.5	32	66.40%
Partial segments	64, 32, 16	0.2	64	36.32%
TD features	64, 32	0.2	32	61.71%
Spectrograms	64, 32	0.2	32	**81.25%**

Table 3.2: MLP achieved accuracies.

The best model accuracy and loss behavior during the training and the testing on every signal epoch are depicted in Figures 3.10 and 3.11. The MLP shows better re-

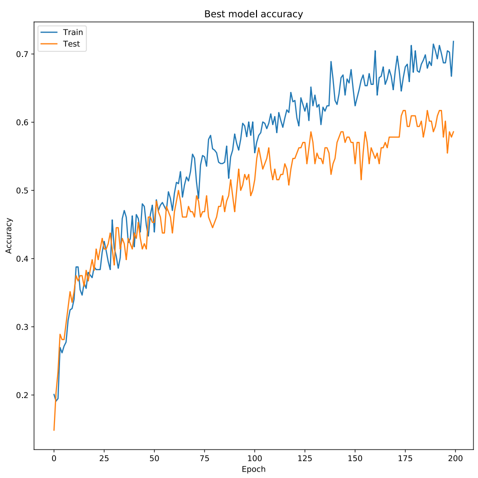

Figure 3.10: Training and test accuracies of the best MLP model.

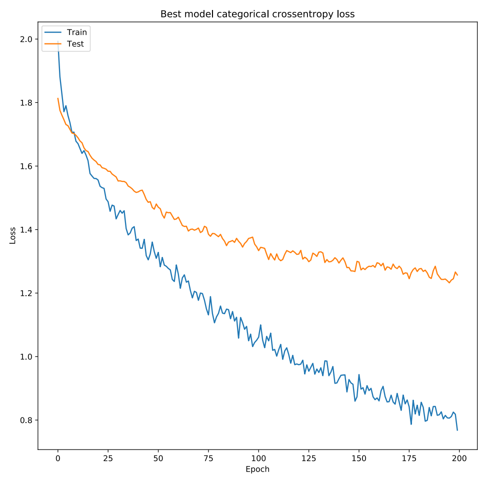

Figure 3.11: Training and test losses of the best MLP model.

sults than those given by the SVM. The low performance on the ”partial segments” data-set is due to some similar segments (e.g. segments describing a resting state of the muscle), which are present in all classes. The results on the "TD features" data-set is better than those provides by SVM, since a Neural Network enough capacity to distinguish between similar samples and to find complex decision boundaries. The reason why the MLP performs better on the "Full segments" data-set can be explained by the fact that a Neural Network is able to represent and learn complicated data and therefore obscure the noise in it. Finally the training using the "spectrograms" data-set achieves better accuracy, since the structure of the data is easier to learn than the "full segments" data-set. In fact, using the TD and the FD information in a single sample helps the network focusing on finding the best matching pattern of different samples from the same class.

3.2.3 Evaluation of the Convolutional Neural Network Model

The best achieved accuracies depending on the data-set type are presented in the following table.

Data-set	Number Filters	Kernel Size	Pooling Size	Number Neurons	Dropout	Batch Size	Accuracy
Full segments	64, 32, 16	3	2	512	0.5	64	84.43%
Partial segments	64, 32, 16	3	2	512	0.5	128	51.88%
Spectrograms	64	3	2	512	0.5	64	**89.84%**

Table 3.3: CNN achieved accuracies.

Figures 3.12 and 3.13 show the accuracies and the losses of the best CNN model. The "spectrogram" data-set is better than all data-sets due to its lower dimensionality and high information content. The CNN outperforms the MLP model due to the fact the CNN starts by extracting features and then finds a mapping from the learned feature space to the output space.

3.2.4 Evaluation of the Long Short-Term Memory Model

The Table 3.4 shows that, similar to CNN, the LSTM achieves its best accuracy on the "spectrogram data-set", but its complexity level is higher.

Data-set	Number of Units	Dropout	Number of Neurons	Batch Size	Accuracy
Full segments	64	0.5	64	32	28.90%
Partial segments	32	0.2	64	128	25.00%
Spectrograms	64	0.2	64	64	**82.03%**

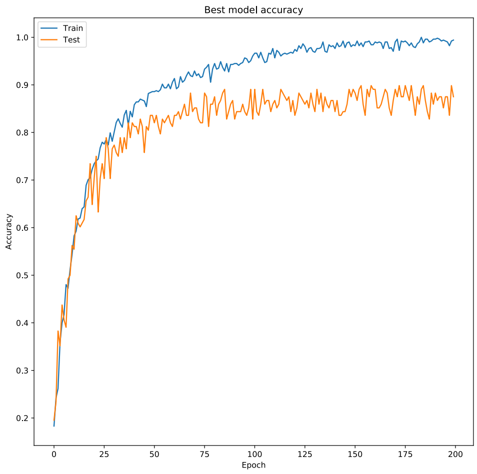

Figure 3.12: Training and test accuracies of the best CNN model.

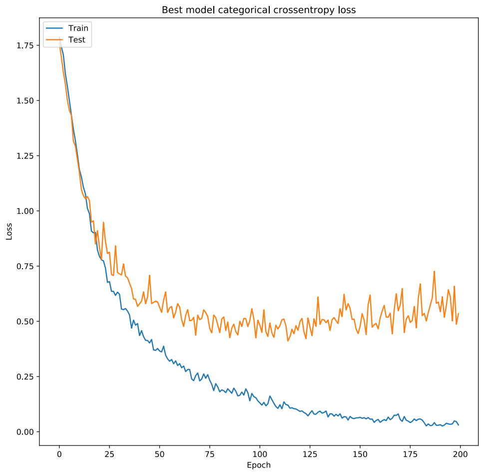

Figure 3.13: Training and test losses of the best CNN model.

Table 3.4: LSTM achieved accuracies.

A possible reason why its achieved accuracy is worse than CNN is that LSTM introduces a more complex neuron structure and needs more tuning to perform better.

3.3 Experiments on a Simulated Robot Arm

The last section shows experiments conducted on the simulator. Figure 3.14 presents an example of a prediction of a random movement. As expected, the SVM model with an accuracy of 38.28% is prone to fail, however, by trying several random movements, NN models estimate the movement class with high accuracy. The failure rate when selecting the CNN model is very low.

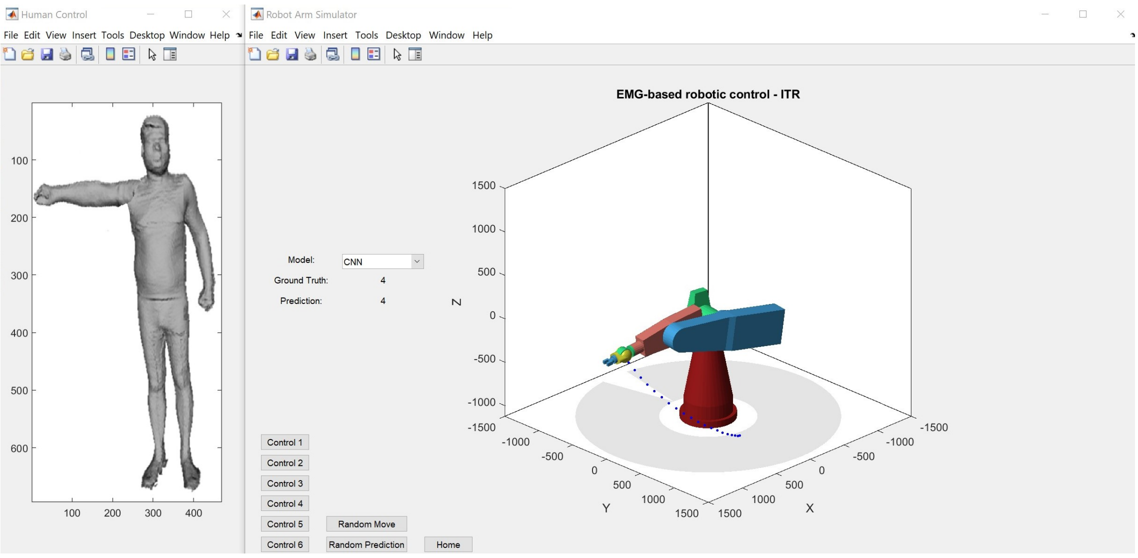

Figure 3.14: 3D representation of the extracted full segments for movement classes 3 and 5.

Chapter 4

Conclusion

This work presents solutions to the EMG-based robot control problem and shows a demo on a simulated robot arm. A tool-chain was build starting by collecting the data, going through different processing steps and up to the determination of the joint positions of the robot. Six movement classes were considered, which makes the classification task much more challenging. A comparison shows that neural network based systems combined with the spectrograms are able to achieve high accuracy. A training on different data-set types shows that classification systems based on the full-semgents or on the sepectrograms outperform those that are based on the partial-segments. This could be explained by the fact, that the full-segments and the spectrograms cover the entire recording time of the movement. However, they may be not suited for the real-time robotic applications. More specifically, CNN and LSTM-based approaches achieve the highest accuracies using spectrograms, which implies that spectrograms are able to cover enough data for an accurate classification. A robot arm simulator illustrates one of the useful application of EMG signals.

List of Figures

Bibliography

[AK10] P. K. Artemiadis and K. J. Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *IEEE Transactions on Robotics*, 26(2):393–398, April 2010. doi:10.1109/TRO.2009.2039378.

[CRMA+13] Rubana Chowdhury, Mamun Bin Ibne Reaz, Mohd Mohd Ali, Ahmad Ashrif A Bakar, Kalaivani Chellappan, and Tae-Gyu Chang. Surface electromyography signal processing and classification techniques. *Sensors (Basel, Switzerland)*, 13:12431–66, 09 2013. doi:10.3390/s130912431.

[EFM17] Gabriel D. Eisenberg, Kyle G.H.M. Fyvie, and Abdul-Khaaliq Mohamed. Real-time segmentation and feature extraction of electromyography: Towards control of a prosthetic hand. *IFAC-PapersOnLine*, 50(2):151 – 156, 2017. Control Conference Africa CCA 2017. URL: http://www.sciencedirect.com/science/article/pii/S2405896317335644, doi:https://doi.org/10.1016/j.ifacol.2017.12.028.

[OH08] M. A. Oskoei* and H. Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Transactions on Biomedical Engineering*, 55(8):1956–1965, Aug 2008. doi:10.1109/TBME.2008.919734.

[TAZ+17] Jingwei Too, Abdul Rahim Abdullah, Tengku Zawawi, Norhashimah Mohd Saad, and Haslinda Musa. Classification of emg signal based on time domain and frequency domain features. 1:25–30, 10 2017.

[ZJCT17] Xiaolong Zhai, Beth Jelfs, Rosa Chan, and Chung Tin. Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network. *Frontiers in Neuroscience*, 11, 07 2017. doi:10.3389/fnins.2017.00379.