Learning by Demonstration applied to NAO Robot – 2015-2016

Movement Generalization and Classification

Kinesthetic Learning

José Pedro Medeiros
2010129934

Abstract

This report describes four approaches for learning and generalize NAO Robot movements. All of the approaches are based on [1], where kinesthetic movements are learned and generalized by a robot. We have acquired a dataset consisting of data from 12 joints from NAO arms (6 joints corresponding to the left arm and 6 joints corresponding to the right arm) in order to test it in the different approaches. This dataset was collected by reproducing the same movement on a total of 6 repetitions done by two different persons, 3 repetitions each.

The approaches taken in this work have the goal to learn and represent a dataset of movements performed by the NAO robot trough kinesthetic movements. In the first approach applied Principal Component Analysis **(PCA)** was used to reduce the dataset dimensions, later on the data was aligned using the Dynamic Time Warping **(DTW)**. These are two pre-processing steps used to improve the Gaussian Mixture Model **(GMM)** algorithm that was applied to the resulting data, next and finally after this classification we obtain a "signature" using the Gaussian Mixture Regression **(GMR)**.

In a second approach **GMM** and **GMR** were applied directly to the original data without any pre-processing steps.

In the third approach the first two steps are equal to the first approach but instead of concatenating the data, "signatures" were created for each repetition of the movement and in the end, an average from all signatures was generated.

The last approach applies **DTW** as a pre-processing step and **GMM** and **GMR** are applied in the concatenated data.

Figure 1: Imagem original

Contents

1 Introduction

The objective of this work is to learn, generalize and classify NAO robot movements. The kinesthetic movements were learned by NAO robot, the dataset was generated by repeating the same movement 6 times by two different persons, each person repeated the same movement 3 times. The robot learned 3 different movements, the "goodbye" movement, the "clapping" movement and the "bolt" movement. Since the 3 different movements only involved moving the arms, we focused on collecting the data related to the joints from the arms. NAO robot as 6 joints in each arm, so in total we collected data from 12 different joints.

For the data collection a python script was created. The script was programmed to read joint's values each 100 milliseconds, in the end it's created a .txt file containing data formatted in 12 vectors, each vector corresponding to one joint. Taking into account the time step time vectors were created for each vector.

After this step, 4 different approaches were followed in order to generalize and classify the movements. All of these approaches were based on [1], section 3 of this report will explain in detail each one.

Is important to refer that this report only present results for the "bolt" movement.

2 Algorithms

2.1 Principal Component Analysis:

Principal Component Analysis algorithm was applied to reduce the dataset dimension. Although we are not explaining here the algorithm in detail a brief explanation is given since it is important to understand how to approximate the data back to the original after **PCA** was applied.
If we start with a dataset X_n^m

Matlab Code

(Melhorar)

[COEFF, PC, LATENT] = pca(X); The combination pc * w' will recreate the original data, minus its mean. The mean is always subtracted prior to performing PCA. Therefore to get the original data we do mu = mean(X); xhat = bsxfun(@minus,X,mu); norm(PC * COEFF' - xhat);

Because COEFF is orthogonal, you also have Xhat * COEFF = pc, or schematically (i.e. this code won't execute) (X - mu) * COEFF = PC <=> X = mu + PC * COEFF'

To get an approximation to your original data, you can start dropping columns from the computed principal components. To get an idea of which columns to drop, we examine the LATENT(eigenvalues) Relevance=ev/sum(ev)*100

2.2 Dynamic Time Warping:

2.3 Gaussian Mixture Model:

2.4 Gaussian Mixture Regression:

3 Approaches

3.1 First Approach:

Given the movement observations represented in 2 dimensions vectors with raw data, time and joint values([-2;2]) , we have resized all vectors to the smallest sized one (if we resized to any other

Table 1: PCA Analysis

Bolt Movement :	Performance	Number of Princ. Comp.
Dataset: medeiros-bolt2015-12-7-17-4-31	performance1 = 99.4579	num_pc1 = 3
Dataset: medeiros-bolt2015-12-7-17-5-12	performance2 = 99.3659	num_pc2 = 3
Dataset: medeiros-bolt2015-12-7-17-5-12	performance3 = 99.5385	num_pc3 = 4
Dataset:rui-bolt2015-12-7-17-5-49	performance4 = 99.4232	num_pc4 = 3
Dataset:rui-bolt2015-12-7-17-6-3	performance5 =99.0995	num_pc5 = 2
Dataset:rui-bolt2015-12-7-17-6-14	performance6 =99.6601	num_pc6 = 3

we could break or slow down the robot movements). After this step Principal Component Analysis **(PCA)** was applied to the vectors in order to reduce dimensionality of the data, as shown in 1.

After applying **PCA** to the data we verified how many principal components we need to obtain a 99 % "comeback" to the original data. Looking at table 1 we can easily conclude that we need to use 4 principal components to obtain this performance, since the third dataset needed 4 principal components to achieve the desired performance. The next step was to temporally align the different datasets, to accomplish that the **DTW** algorithm developed in [2] was used.

After these two pre-processing steps the data was concatenated. We have then, applied the Gaussian Mixture Model and Gaussian Mixture Regression to the concatenated data using the code provided with the article [1], since we have a time constraint concatenate data will improve **GMM** performance since we group the data and this algorithm uses clusters to classify the data. Results for this approach applied to the bolt movement are shown in the experimental results section.

We applied the resulting **(GMR)** "signature" to the mean of the resized vectors.

3.2 Second Approach:

In this approach there were no pre-processing steps the Gaussian Mixture Model and the Gaussian Mixture Regression were applied to the original Data. The algorithm present a poor performance probably due to the fact we have not used **DTW** to align the data, it is known that **GMM** can perform poorly in this context.

The results can be observed in the experimental results section.

3.3 Third Approach:

This approach was similar to the first one, the same pre-processing steps were applied to the data. The same **PCA** analysis from 1 was performed. The difference is that the data was not concatenated instead it was created a "signature" for each movement repetition. In the end it was created a final "signature" that was a result from all the previous. This "signature" was then applied to the mean of the original vectors.

The results can be observed in the experimental results section.

3.4 Fourth Approach:

After the poor performance in the second approach we decided to align data, using **DTW** before applying **GMM** and **GMR** to the data.

The results can be observed in the experimental results section.

4 Experimental Results

Results were only shown for the first joint of the "bolt" movement since it was one of the most important joints in this movement and a lot of variation.

4.1 Refults for the First Approach:

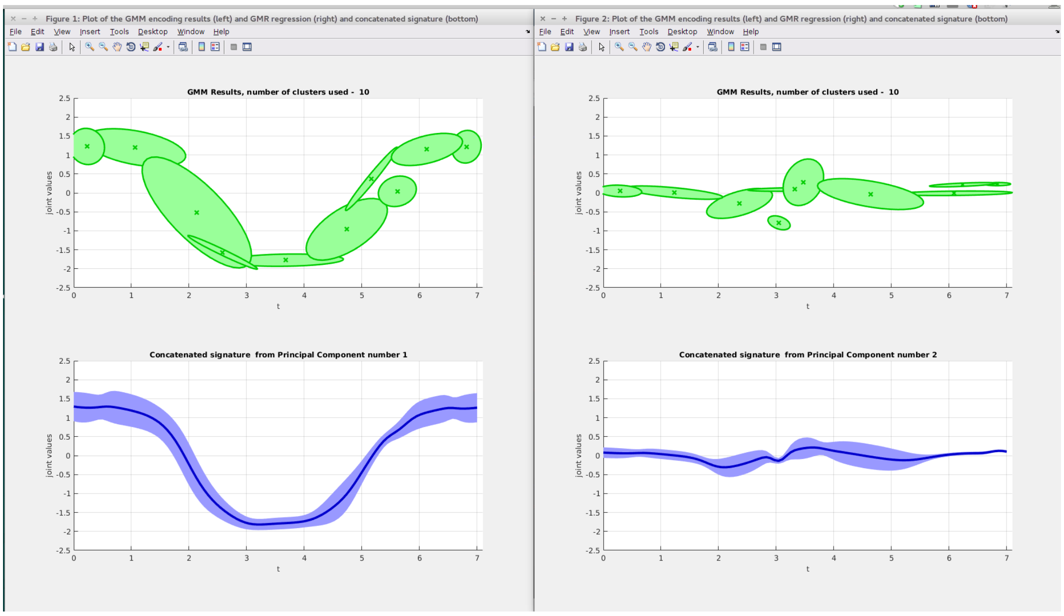

Figure 2: Joint Number 1. Number of cluster used - 10. First Principal component(left) Second Principal Component(right)

We can observe that when we use 5 clusters we have a smoother "signature" than when we use 10 clusters. Note that we are only plotting here 2 of the 4 principal components since the first 2 are the more relevant.

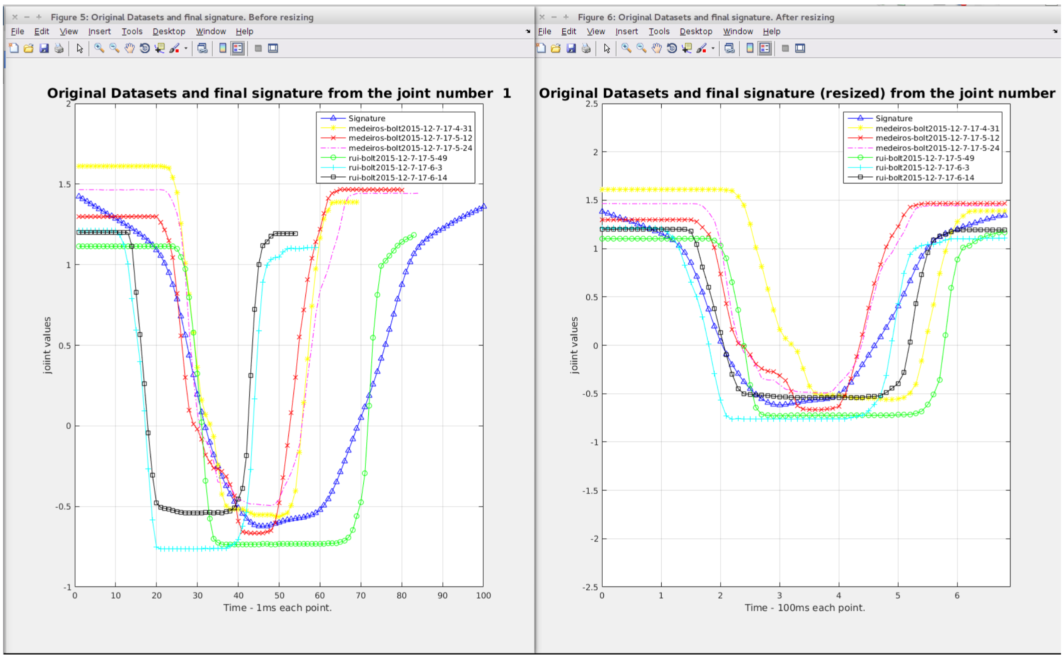

Figure 3: Joint Number 1. Original data and final "signature"(left) Original data and final "signature" resized(right)

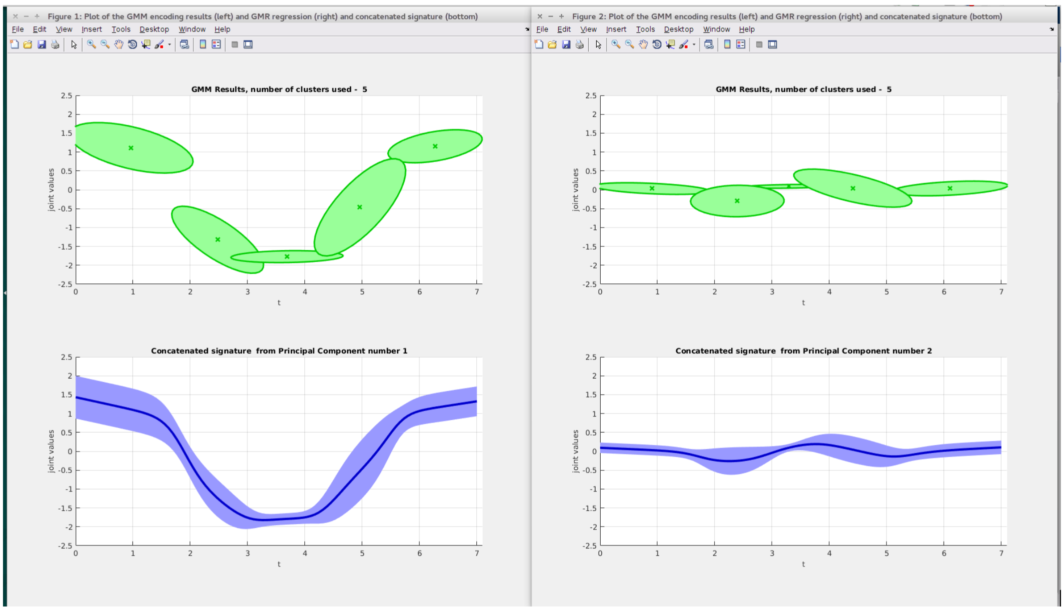

Figure 4: Joint Number 1. Number of cluster used - 5. First Principal component(left) Second Principal Component(right)

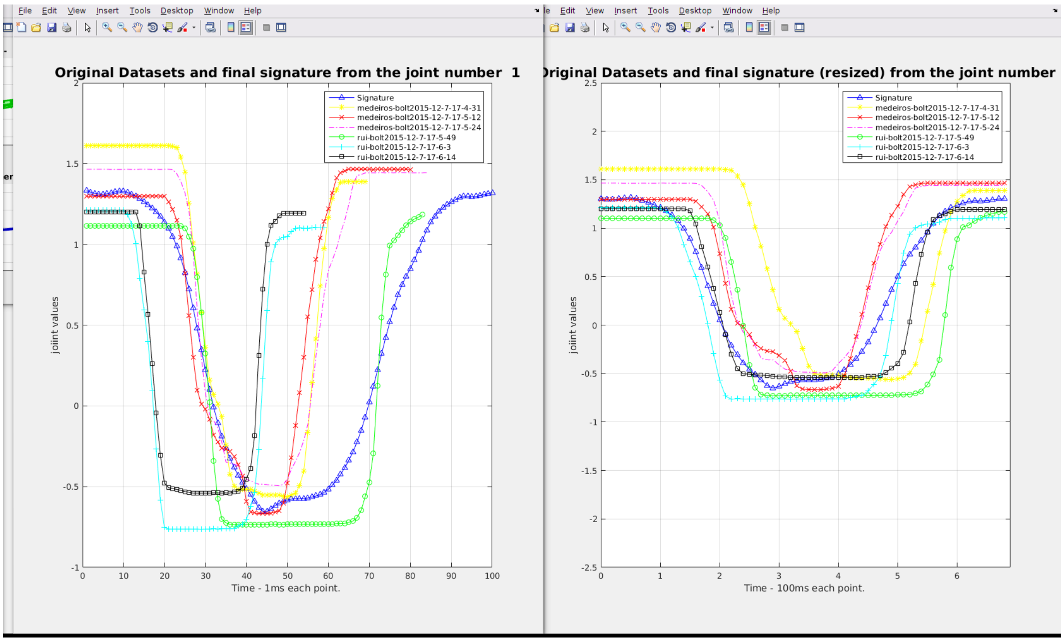

Figure 5: Joint Number 1. Original data and final "signature"(left) Original data and final "signature" resized(right)

4.2 Refults for the Second Approach:

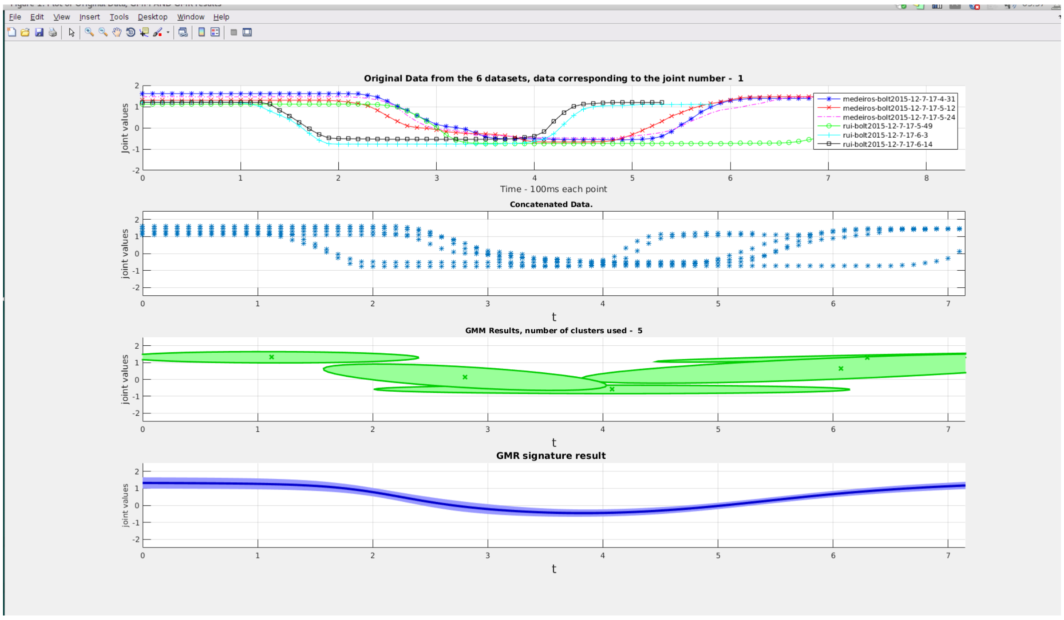

Figure 6: Joint Number 1. Number of cluster used - 5

No pre-procesing steps were performed. As it was explained before the **GMM** and **GMR** performed poorly with this approach.

4.3 Refults for the Third Approach:

Although it is not and efficient approach it has a good performance.

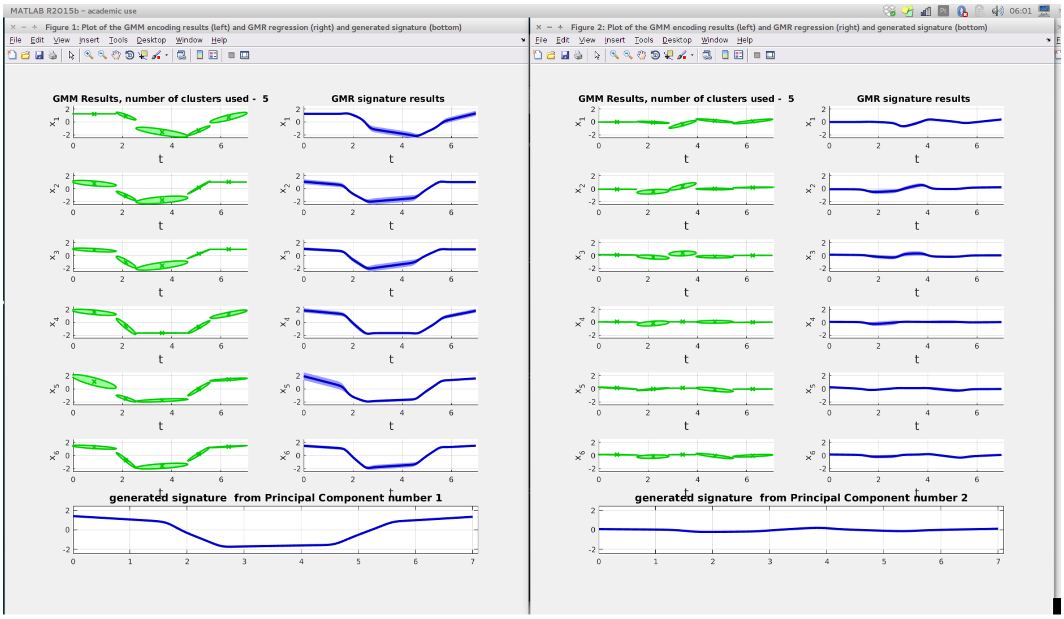

Figure 7: Joint Number 1. Number of cluster used - 5. First Principal component(left) Second Principal Component(right)

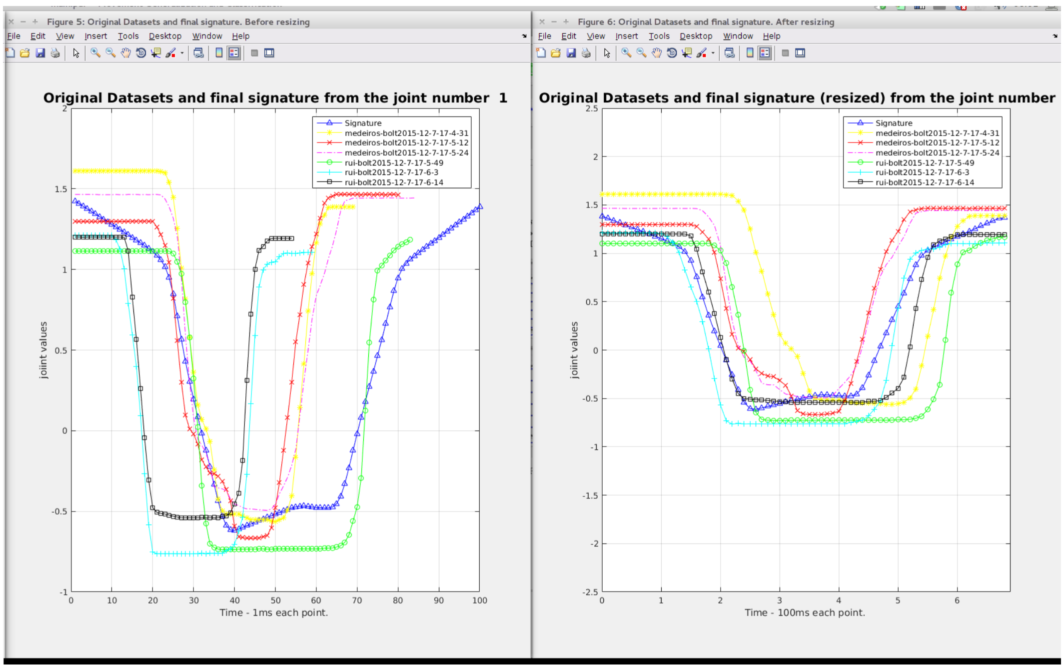

Figure 8: Joint Number 1. Original data and final "signature"(left) Original data and final "signature" resized(right)

4.4 Refults for the Fourth Approach:

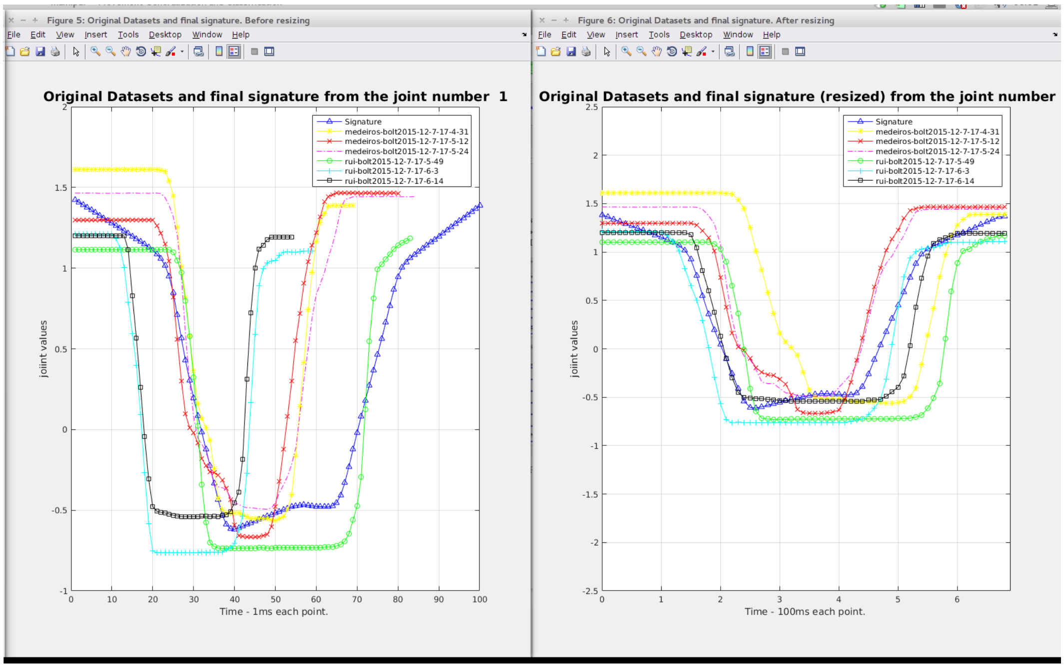

Figure 9: Joint Number 1. Original data and final "signature"(left) Original data and final "signature" resized(right)

The main difference between this approach and the first approach is the time of performing **GMM** since here we apply it to the original dataset, temporally aligned it takes longer time.

5 Conclusion and Future Work

It is possible to observe that the bests approaches are the first and the fourth approach at the same time they are similar to the one taken in the article[1]. By looking into the Principal Components Analysis results it is possible to observe that there is a lot of redundancy.

In the first approach we applied **PCA** and only after temporally align the data with **DTW** since it was the order taken in[1], it would be interesting to perform first the alignment and apply **PCA** after and analyse the results.

In a future work it would be interesting to study which are the most relevant joints for each different movement. It could also be important to analyse the end position of the hands of the robot and study in detail the results.

References

[1] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(2):286–298, 2007.

[2] Pau Micó. Dynamic time warping.