DATA MINING – ASSIGNEMENT 2 REPORT

CSE 572

SPRING 2018

GROUP - 12

Submitted To:

Dr. Ayan Banerjee

Arizona State University

Submitted by:

Aritra Mitra - 1211172453 – amitra8@asu.edu
Arun Karthick Manickam Alagar Muthumanickam - 1213135077 - amanick4@asu.edu
Vinay Bhargav Arni Ragunathan - 1212712330 - varnirag@asu.edu
Giriraj NoLastName - 1213350721 - ggirira2@asu.edu
Deepthi Parvathy Puducode Radhakrishnan - 1213171633 - dpuducod@asu.edu

Contents

1 Introduction:

As a part of Data Mining course requirement project, we are trying to recognize various American Sign language (ASL) gestures by reading the gestures through wrist bands and recognizing them by using Data Mining algorithms. We are trying to extract features which distinguishes one action from other, by further processing these features we ae relating actions to its features. Our final aim is to guess a gesture based on its features.

2 Phase 1

Data Collection:

In the Data Collection phase, we worked on collecting the raw data by performing ASL gestures using wrist bands. In this phase, one member from each team wore the wrist bands and performed the given 20 ASL gestures. Each gesture was performed 20 times and the wrist band readings were read. The 20 gestures which we performed were – About, And, Can, Cop, Deaf, Decide, Father, Find, Go out, Hearing, Here, Hospital, If, Cat, Cost, Day, Gold, Good Night, Hurt and Large. The actions were done for about 3s and were recorded at 15Hz frequency.

The raw data of the wrist bands had data of various sensors such as – Accelerometer – X, Y, Z; EMG pod (0-7); Gyroscope - X, Y, Z; Orientation roll, Pitch, Yaw and Kinnect data for both the hands.

3 Phase 2

In the current Phase 2 of the project, we are first working on reading the data and segregating it for different gestures. The next part of this phase is to extract interesting features of each gesture. We are studying the gesture and intuitively trying to find out its features, then using MATLAB we are trying to extract these features using different feature extraction methods. Post this we are using Principle Component Analysis to perform feature reduction. This gives us the top latent features for an action and the eigen vector analysis shows which features have high variance and define a gesture uniquely.

In this phase we are concentrating only on 10 ASL gestures – About, And, Can, Cop, Deaf, Decide, Father, Find, Go out, Hearing.

3.1 Task 1: Data Preparation

In this task we are reading the raw data from phase 1 and separating the reading for the 10 different gestures. The data contains the actions done by ~37 groups. We are using the data for 5 groups and collating that. The reason for using 5 groups is, collating the data for all the groups is very time consuming as each group has done each gesture 20 times. Secondly, the data for 37 groups is too huge further processing will also be time consuming as we are running the codes in our Personal computers. Finally, as each team has done each action 20 times with 5 persons we are getting the data for each gesture for 20*5 = 100 times, this gives a good person to person variation and will suffice in extracting important features for the action.

Steps done in MATLAB for task 1 (*Task1.m*):

a) Reading the data for gestures: About, And, Can, Cop, Deaf, Decide, Father, Find, Go out, Hearing for folders DM12, DM02, DM03, DM04, DM05 using *xlsread().*

b) From the read excel extracting the numeric data and the text data. The numeric data contains all the sensor values and the text data contains the sensor names.

c) Transposing the numeric data because we need the sensor data row wise. We are taking only 50 readings for each sensor to maintain the consistency. If the readings are less than 50 we do zero padding.

d) Extracting the sensor name for each row of numeric data from the text data and for each action of the gesture adding the action number for the row header. The Row header looks like 'Action 1 ALX'.

e) Appending the data for 20 actions for each group and then appending the data for the 5 groups for individual gesture.

f) This total data is stored in a csv file named as the gestures name 'About.csv' using *xlswrite*() and then stored in a folder named 'Output'.

Some sample output data:

```
Action 1 ALX       0.943848     0.95166     0.95459     0.95459   0.949707 ....
Action 1 ALY       -0.27979    -0.25244     -0.2627    -0.26465     -0.2627....
Action 1 ALZ        0.12207     0.12793     0.12207    0.127441    0.116211....
Action 1 ARX       0.918457    0.918457    0.918457    0.918457    0.918457....
Action 1 ARY       0.310059    0.310059    0.310059    0.310059    0.310059....
Action 1 ARZ       0.130371    0.130371    0.130371    0.130371    0.130371....
Action 1 EMG0L            0          -1          -2          -1          -3....
Action 1 EMG1L            5           2           0          -3          -1....
Action 1 EMG2L            1          -4           1          -2           0....
…….
…….
```

3.2 Task 2 and Task 3: Feature Extraction and PCA

In Task 2 we are extracting the interesting features for every gesture and comparing these features with the intuitive features of the action. The feature which differentiates an action, or which describes a gesture is the feature we are looking for. By observation we can guess the features of a gesture, we are verifying our observation through different feature extraction algorithms in MATLAB.

The Feature Extraction Algorithms used:

a) Fast Fourier Transform (FFT)
b) Variance
c) Discrete Wavelet Transform (DWT)
d) Root Mean Square (RMS)
e) Pearson Correlation Coefficient (PCC)

3.2.1 About

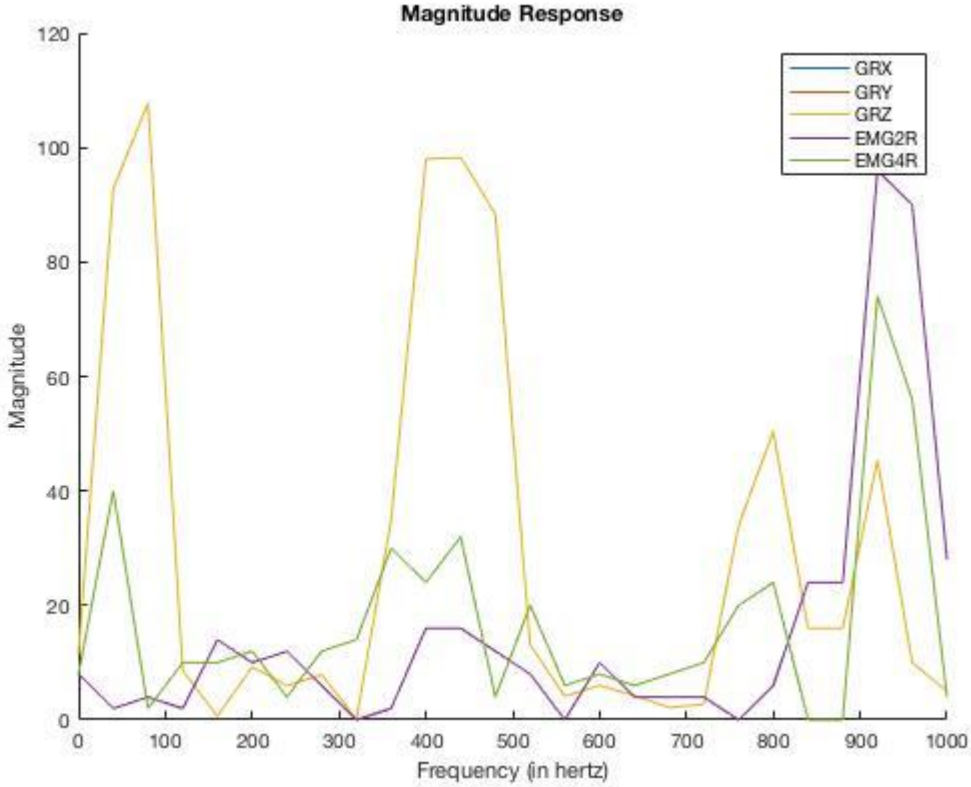
For feature extraction for About action we are using **Fast Fourier Transform**.

a) The raw signals collected from the sensor during the phase 1 is converted into frequency components via Fast Fourier transform. The main idea of using Fourier transform is to identify the ASL action using the spikes in certain features. The advantage of Fast Fourier Transfer is that number of computations is less when compared to other transformations. We are using MATLAB's in-built function for Fast Fourier Transform. The way FFT operates is by decomposing an N point time domain signal into N time domain signals each containing a single point. Next, we are calculating the frequency corresponding to domain signals. For this phase, we consider 18 features from Gyroscope, Accelerometer, Orientation and EMG sensors.

b) For the "ABOUT" ASL action applying FFT to predict what feature values shows variation for the movement. When trying to extract the features manually from the data it looks like there is change in GRX, GRY, GRZ, EMG 2R, EMG 4R as for ABOUT action there is a rotation of right hand. We feed in all these feature parameters for FFT to calculate the frequency.

c) As a first step, extracting the CSV file of the ASL Action (Action.csv), which we got it from the output of task 1. The feature index values are specified to keep track of the value peaks for these specified features. Iterate over every index and send the values for FFT transformation. Consider all possible values of teams 12, 5, 2, 4 and 3 for a feature.

d) Normalizing the data with Z- order transformation. This is to ease the plotting. Plot the Frequency obtained from the Fourier transform against the magnitude and observe the peaks for the action.

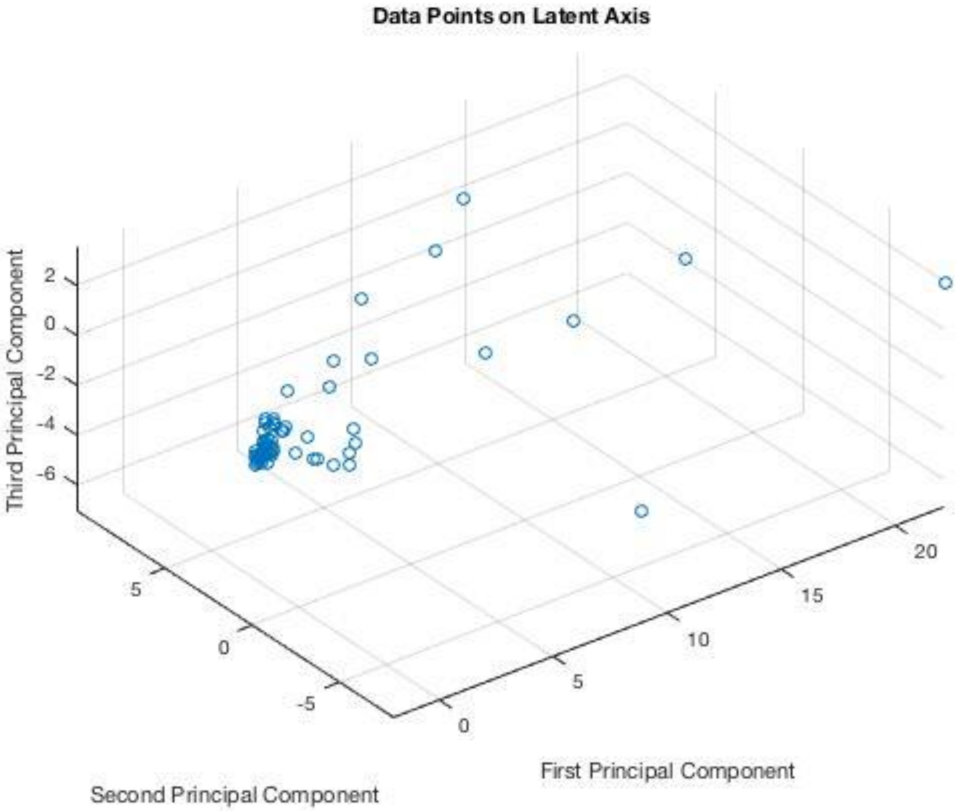


e) For **Task 3 PCA**, the Z- order values are passed to PCA to retrieve the core matrix and the feature matrix against the latent semantics. Plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics. Also, we are plotting the spider plot for eigen vectors obtained from PCA.

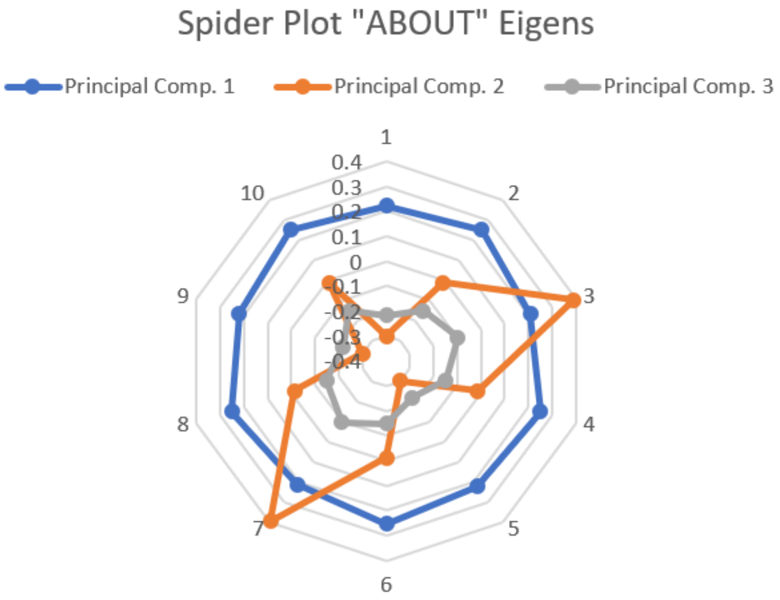
f) Our intuition for features of 'About' gesture is perfectly justified because we observe from the explained parameter of the PCA output that we are able to preserve 91.5% of the variance using the first three eigen vectors. PCA was certainly helpful in this scenario. We can reduce the feature matrix to NX3 and still preserve 91.5% variance in the combined dataset.

g) The relevant Matlab code can be found in About_FFT.m.

PCA:



Spider Plot



3.2.2 And

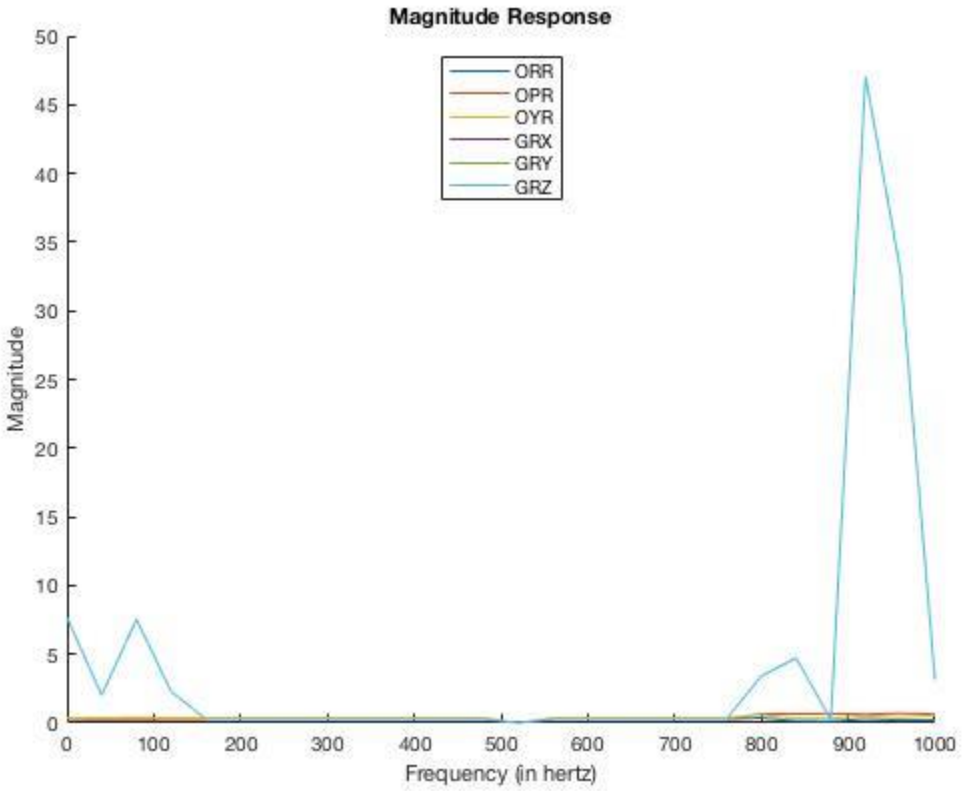
For feature extraction for And action we are using **Discrete Wavelet Transform**.

a) The sensor data is represented as wavelet forms. DWT is applied on data where the frequency component of the signal and its corresponding temporal resolution must be observed. We are particularly interested in decomposing the feature action with orientation and gyroscope, since we see a change in orientation as a part of the ASL action “AND”. We are using DWT function which is inbuilt function of MATLAB to observe the variation of data along the features. We are considering relevant features to the Orientation and Gyroscope like OPR, ORR, OYR, GRX, GRY and GRZ to observe variance in the data points for the action.

b) Here we intend to see a difference in magnitude when right hand is moved. The ideal transformation that we can use for this scenario is discrete wavelet transform, as it represents the frequency components of a signal as well as the corresponding temporal location for those frequencies. This overcomes the disadvantage of the Fast Fourier transform which considers only the frequency components. In this case, we are expected to see high peak of signal and a sudden drop of magnitude at frequency 830 Hz. This corresponds to the right hand moving from the left to right, once it reaches the original position the magnitude drops.

c) The relevant MATLAB code can be found in And_DWT.m.

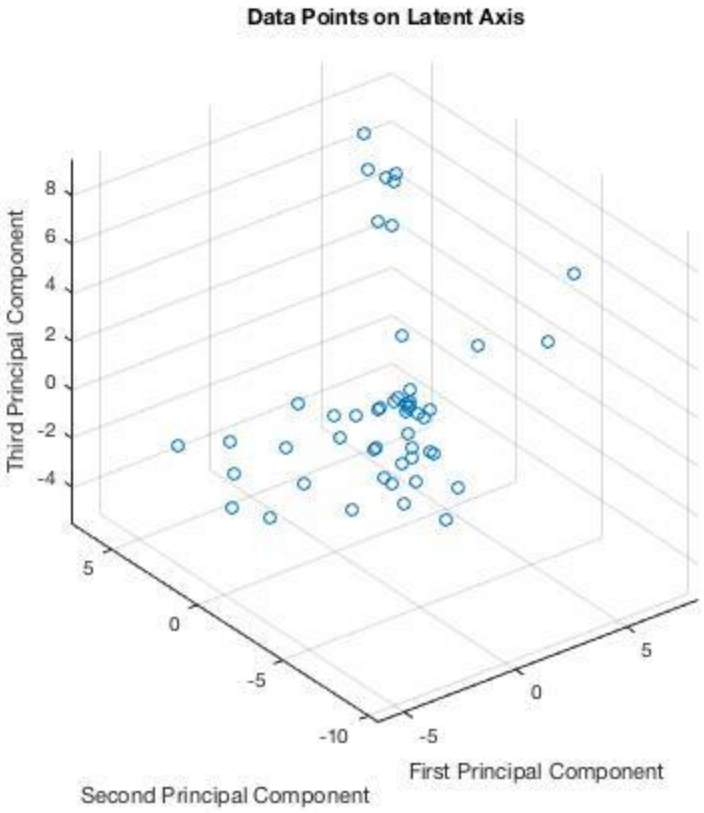
d) Thus, the difference can be observed in the orientation and gyroscope feature of right hand thereby demarcating the ASL action “AND” from all other actions. Compute the Z order score to normalize, and to make the plot look readable. We are plotting the temporal frequency against the magnitude to see the spikes of variation of data.



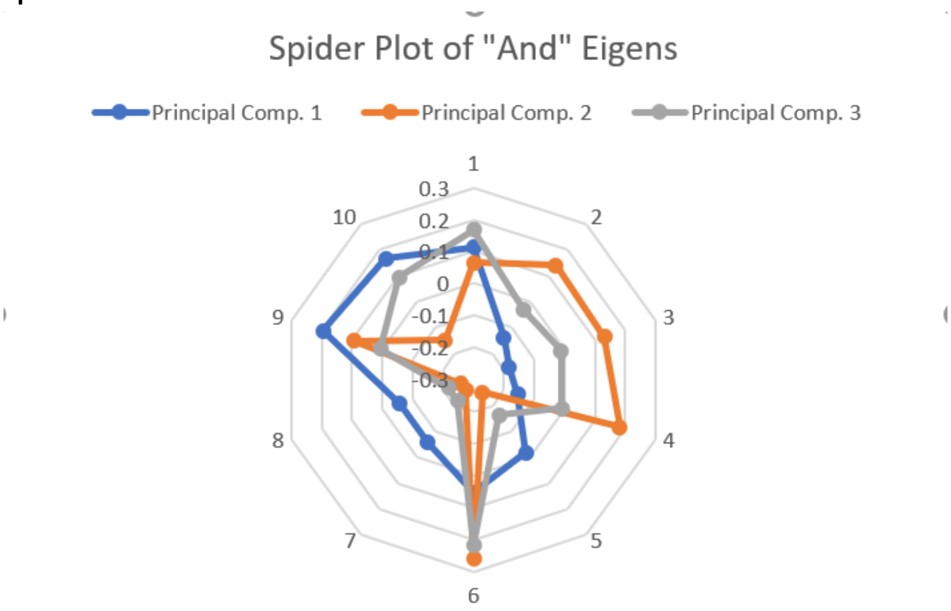
e) For **Task 3**, we pass the Z order values to PCA and retrieve the core matrix and the feature matrix against the latent semantics. Plot the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics.

f) Our intuition for features of ‘And’ gesture is partially justified because we observe from the explained parameter of the PCA output that we are able to preserve only 65% of the variance using the first three eigen vectors. Here we are reducing the feature matrix to NX3 and preserve 65% variance in the combined dataset.

g) Below are the graph plots for the mentioned graph plots.
PCA:



Spider Plot :



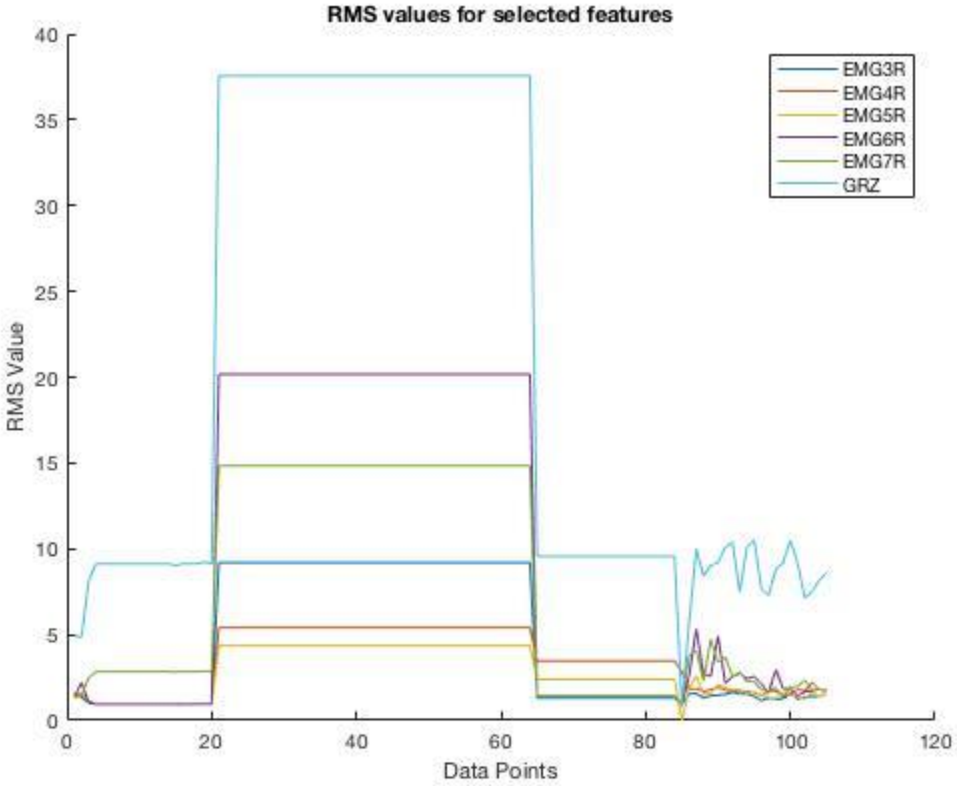
3.2.3 Cop

For feature extraction for Cop action we are using **Root Mean Square**.

a) Generally, Root mean square is used to measure the typical magnitude of the set of features, that is, computed as the square root of the arithmetic/quadratic mean of the squares of data points. Typically, is the measure of imperfection in the data set which is used to distinguish data points.

b) In our case, the data from the various sensors is in form of signal waves, hence applying an RMS on it will discriminate features from one another.

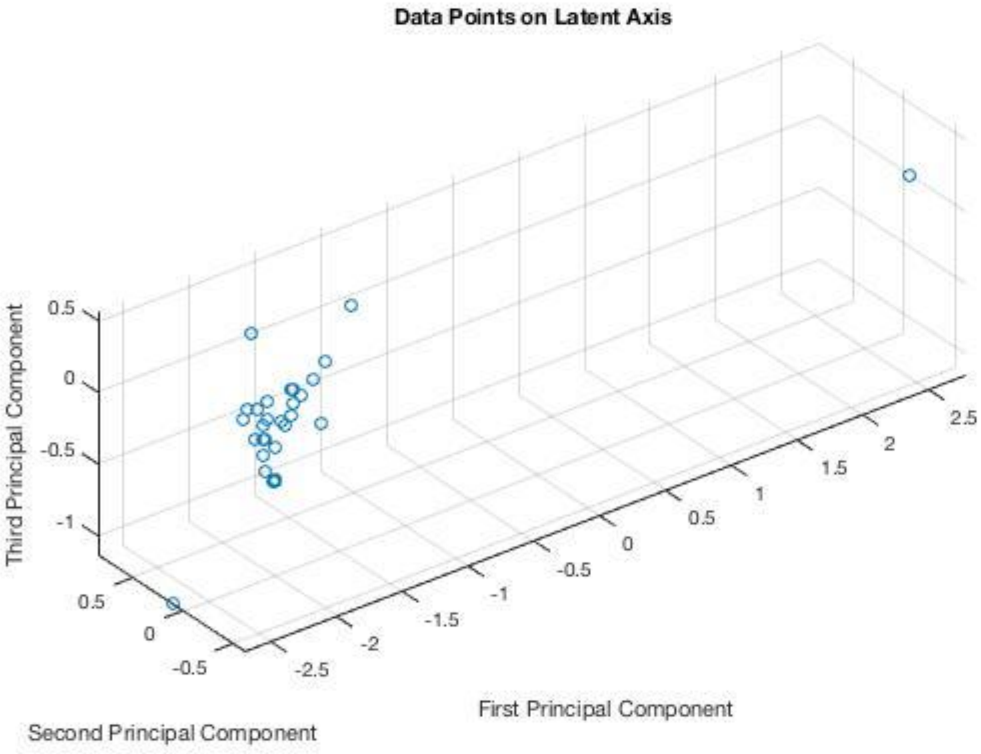
c) From our observation and intuition, the "COP" ASL action has distinctive Electromyography EMG (R - 3,4,5,6,7) and Gyroscope (R - Z) change in the Right hand which are relatively easy to differentiate among the other features.



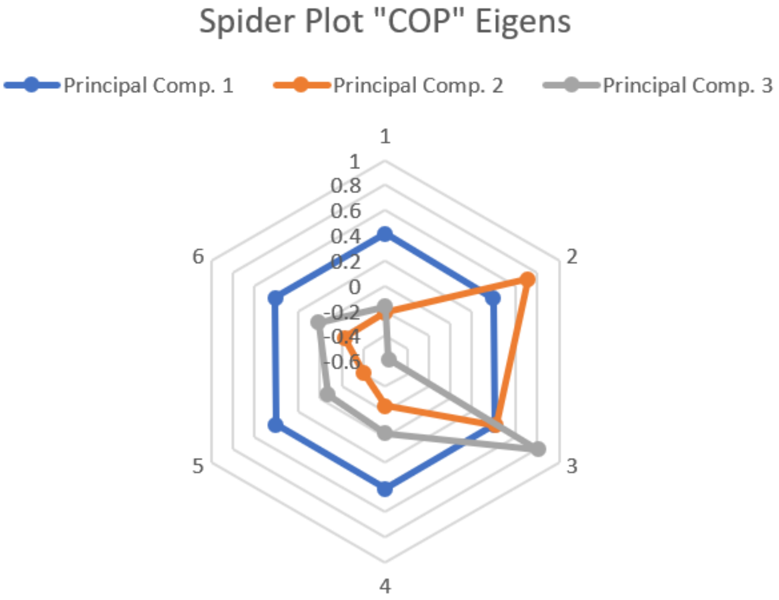
d) For **Task 3 PCA**, the core matrix and the feature matrix are extracted from PCA against the latent semantics, thereby plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of data points in Latent Semantics. Also, we are plotting the spider plot for eigenvectors obtained from PCA.

e) With our intuition for "COP" action, the PCA output was able to preserve 99% of variance using the first three eigenvectors. Holistically, we can prune the feature matrix to Nx3 and still preserve 99% variance in the combined dataset.

f) The relevant Matlab code can be found in Cop_RMS.m. Below are the graph plots for the above-mentioned plot.
PCA:



Spider Plot



3.2.4 Deaf

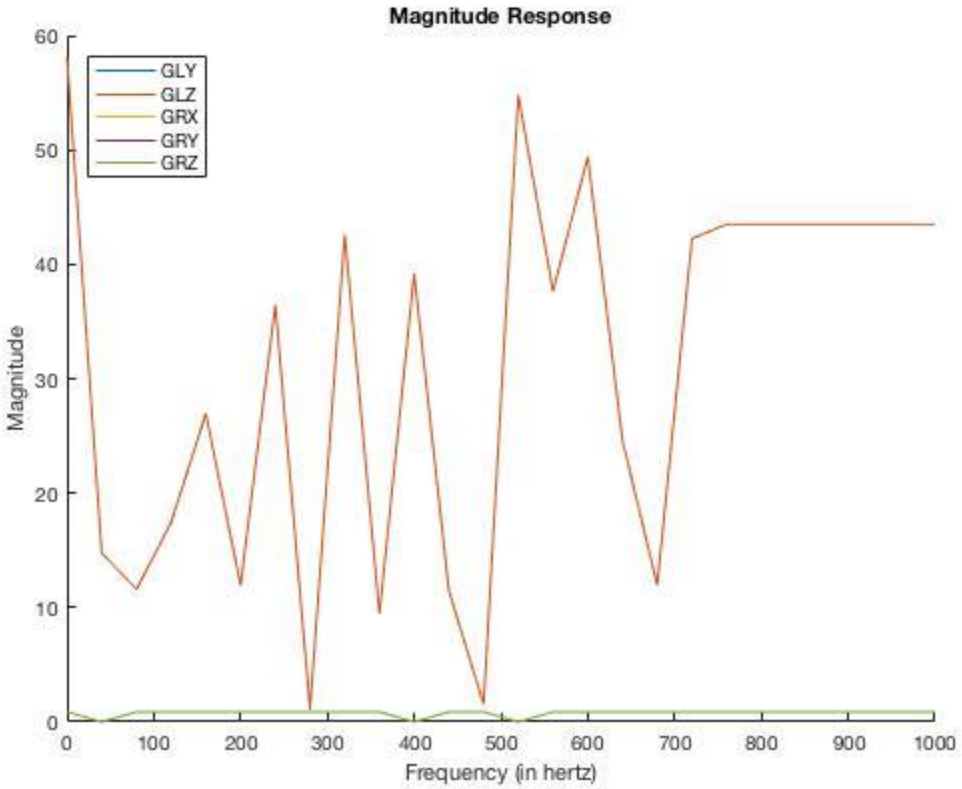
For feature extraction for Deaf action we are using **Discrete Wavelet Transform**.

a) The sensor data is represented as wavelet forms. DWT is applied on data where the frequency component of the signal and its corresponding temporal resolution must be observed. We are particularly interested in decomposing the feature action with orientation and gyroscope, since we see a change in orientation as a part of the ASL action "DEAF". We are using DWT function which is inbuilt function of MATLAB to observe the variation of data along the features. We are considering relevant features to the Orientation and Gyroscope like OPR, ORR, OYR, GRX, GRY and GRZ to observe variance in the data points for the action.

b) Here we intend to see a difference in magnitude when right hand is moved. The ideal transformation that we can use for this scenario is discrete wavelet transform, as it represents the frequency components of a signal as well as the corresponding temporal location for those frequencies. This overcomes the disadvantage of the Fast Fourier transform which considers only the frequency components. In this case, we are expected to see high peak of signal and a sudden drop of magnitude at frequency 475 Hz. This corresponds to the slight movement of right hand from the left to right near the mouth, once the hand stops the magnitude drops.

c) The relevant MATLAB code can be found in Deaf_DWT.m.

d) Thus, the difference can be observed in the orientation and gyroscope feature of right hand thereby demarcating the ASL action "DEAF" from all other actions. Compute the Z order score to normalize, and to make the plot look readable. We are plotting the temporal frequency against the magnitude to see the spikes of variation of data.

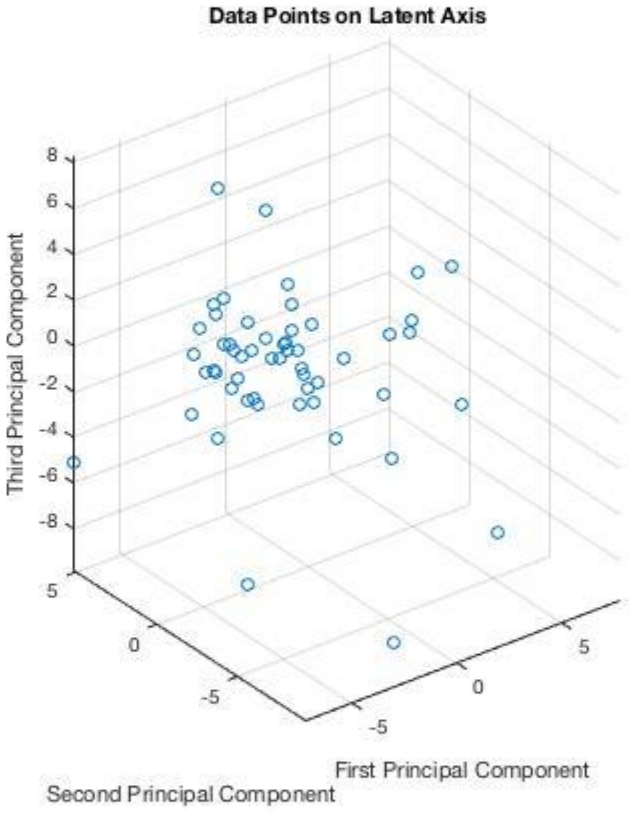


e) For **Task 3**, we pass the Z order values to PCA and retrieve the core matrix and the feature matrix against the latent semantics. Plot the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics.

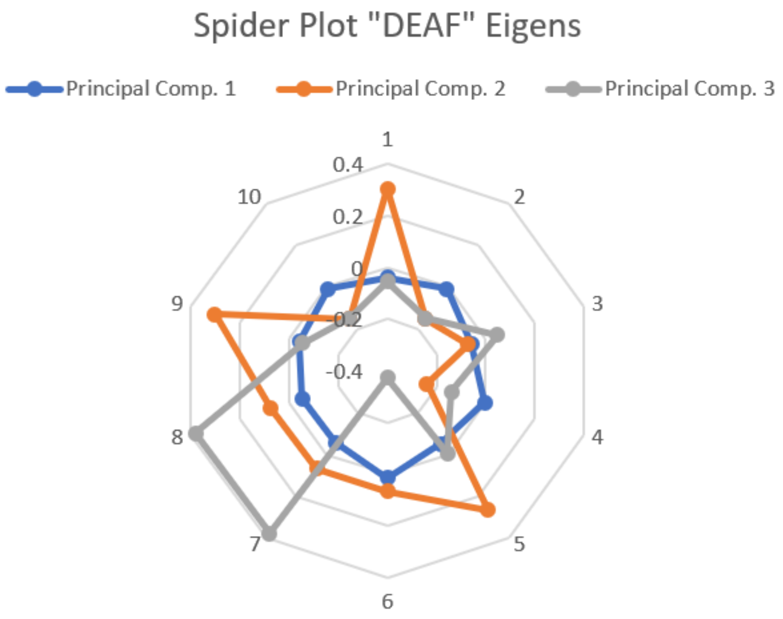
h) Our intuition for features of 'Deaf' gesture is justified because we observe from the explained parameter of the PCA output that we are able to preserve only 80% of the variance using the first three eigen vectors. Here we are reducing the feature matrix to NX3 and preserve 80% variance in the combined dataset.

i) Below are the graph plots for the mentioned graph plots.

PCA :



Spider Plot :



3.2.5 Decide

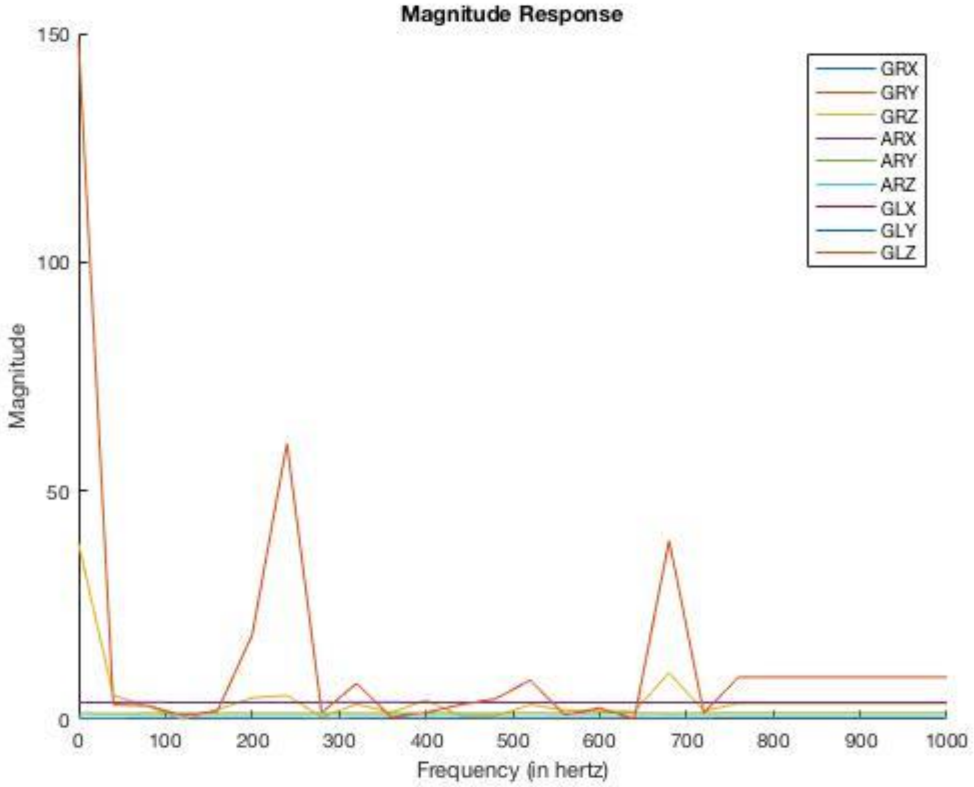
For feature extraction for Decide action we are using **Fast Fourier Transform**.

a) The raw signals collected from the sensor during the phase 1 is converted into frequency components via Fast Fourier transform. The main idea of using Fourier transform is to identify the ASL action using the spikes in certain features. The advantage of Fast Fourier Transfer is that number of computations is less when compared to other transformations. We are using MATLAB's in-built function for Fast Fourier Transform. The way FFT operates is by decomposing an N point time domain signal into N time domain signals each containing a single point. Next, we are calculating the frequency corresponding to domain signals. For this phase, we consider 18 features from Gyroscope, Accelerometer, Orientation and EMG sensors.

b) For the "DECIDE" ASL action applying FFT to predict what feature values shows variation for the movement. When trying to extract the features manually from the data it looks like there is change in GLX, GLY, GLZ, ARX, ARY, ARZ as for DECIDE action there is acceleration in right hand and there is a small angular rotation in left hand. We feed in all these feature parameters for FFT to calculate the frequency. We also tried DWT to extract the features for decide, but it was not giving proper results.

c) As a first step, extracting the CSV file of the ASL Decide (Decide.csv), which we got it from the output of task 1. The feature index values are specified to keep track of the value peaks for these specified features. Iterate over every index and send the values for FFT transformation. Consider all possible values of teams 12, 5, 2, 4 and 3 for a feature.

d) Normalizing the data with Z- order transformation. This is to ease the plotting. Plot the Frequency obtained from the Fourier transform against the magnitude and observe the peaks for the action.

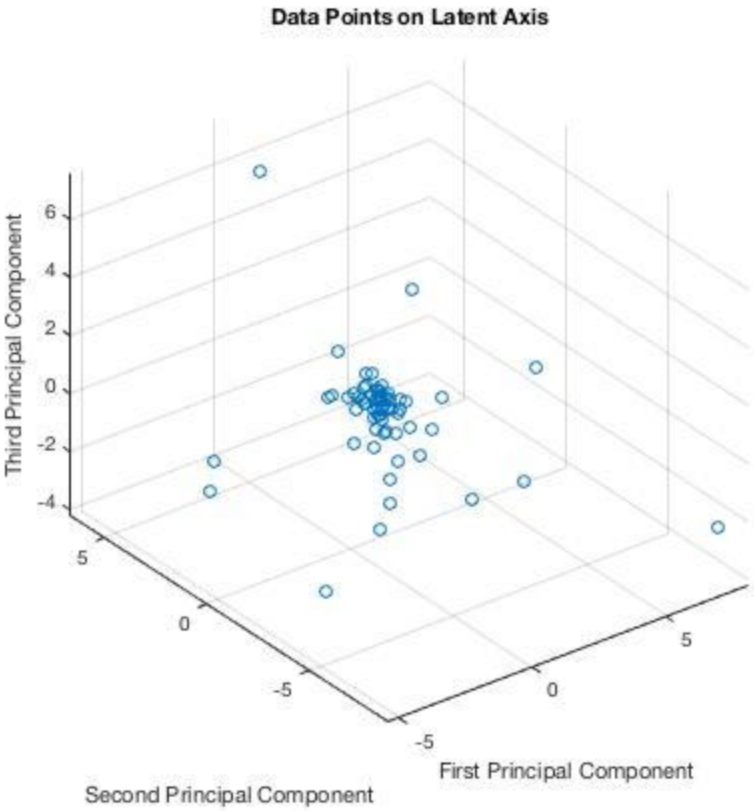


e) For **Task 3 PCA**, the Z - order values are passed to PCA to retrieve the core matrix and the feature matrix against the latent semantics. Plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics.

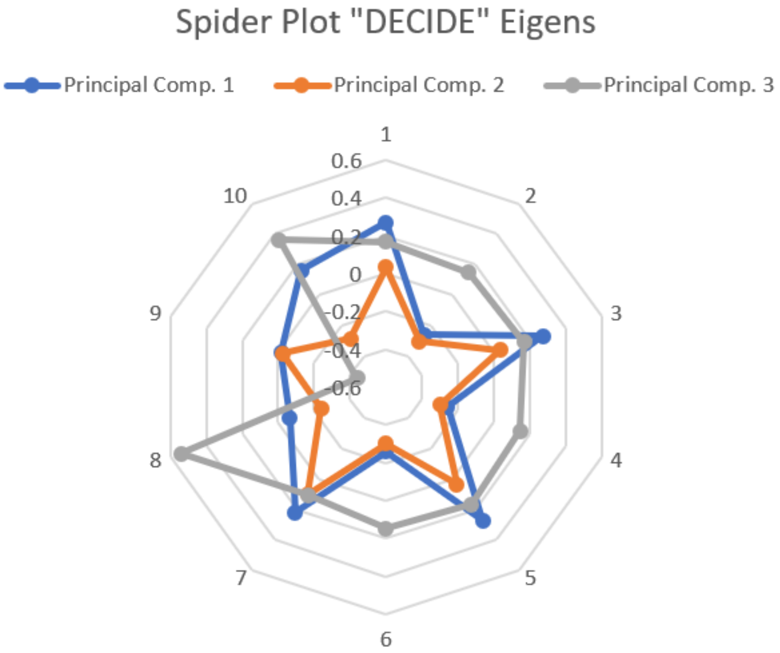
f) Our intuition for features of Decide gesture is partially justified because we observe from the explained parameter of the PCA output that we are able to preserve 76% of the variance using the first three eigen vectors. PCA was certainly helpful in this scenario. We can reduce the feature matrix to NX3 and still preserve 76% variance in the combined dataset. The features which we have selected if it has little variance then we can’t use that feature to explain this action. As our selected features have good variance we are able to get justifiable results

g) The relevant Matlab code can be found in Decide_FFT.m. Below are the graph plots for the above-mentioned plot.

PCA :



Spider Plot :



3.3.6 Father

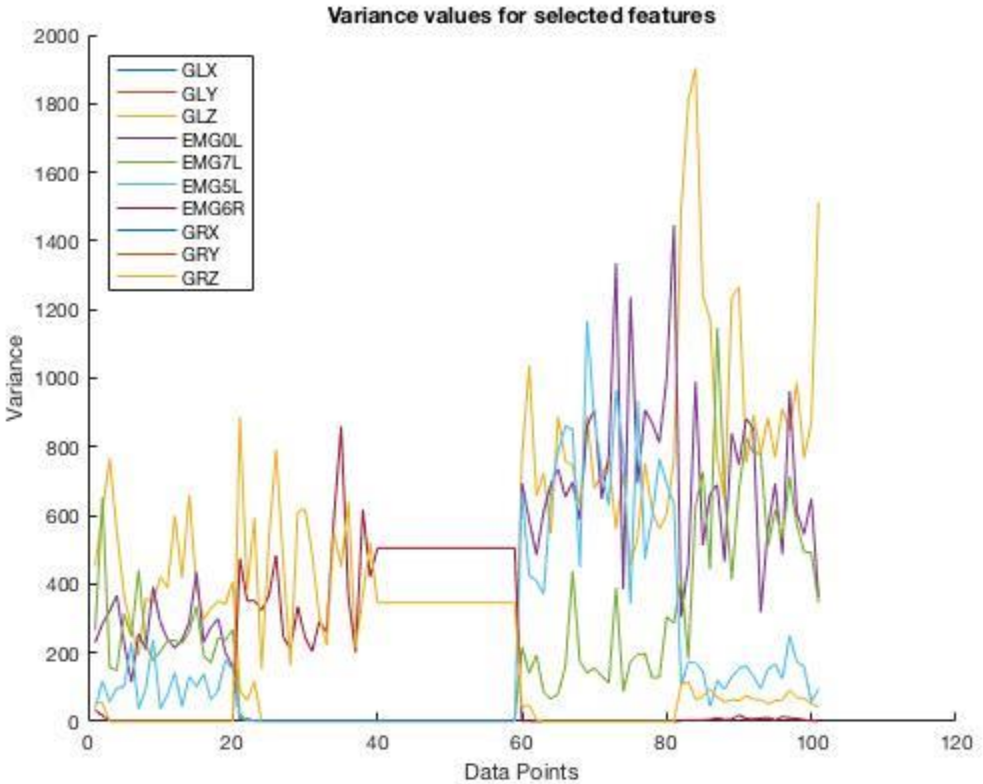
For feature extraction for Father action we are using **Variance**.

a) When it comes to feature extraction, variance plays a critical role. In our case, we take 34 degrees-of-freedom to measure each ASL action which makes it more important to distinguish features which are action specific from the ones which are less informative. Each data segment includes 6 accelerometer data (3 for each hand), 16 EMG Data (8 for each hand), 6 gyroscope data (3 for each hand) and 6 orientation data (3 for each hand). So, by close observation of the data for each action, we tend to extract the feature variance matrix which is of the order Mx34.

b) Basically, out of intuition and set of logic, we choose 7 degrees-of-freedom, in other words, the feature that contributes to maximum variance to distinguish ASL actions. "FATHER" action produces close to zero sensor data for the left hand and there is a decisive change in the EMG and Gyroscope data of the right hand.

c) This variance among data segments can be computed with ease by the MatLab inbuilt feature *var()* which results in precise outcomes.

d) Also, some features do not produce clearer variance between actions. Hence it is important to isolate such data segments to reduce false positives and true negatives.

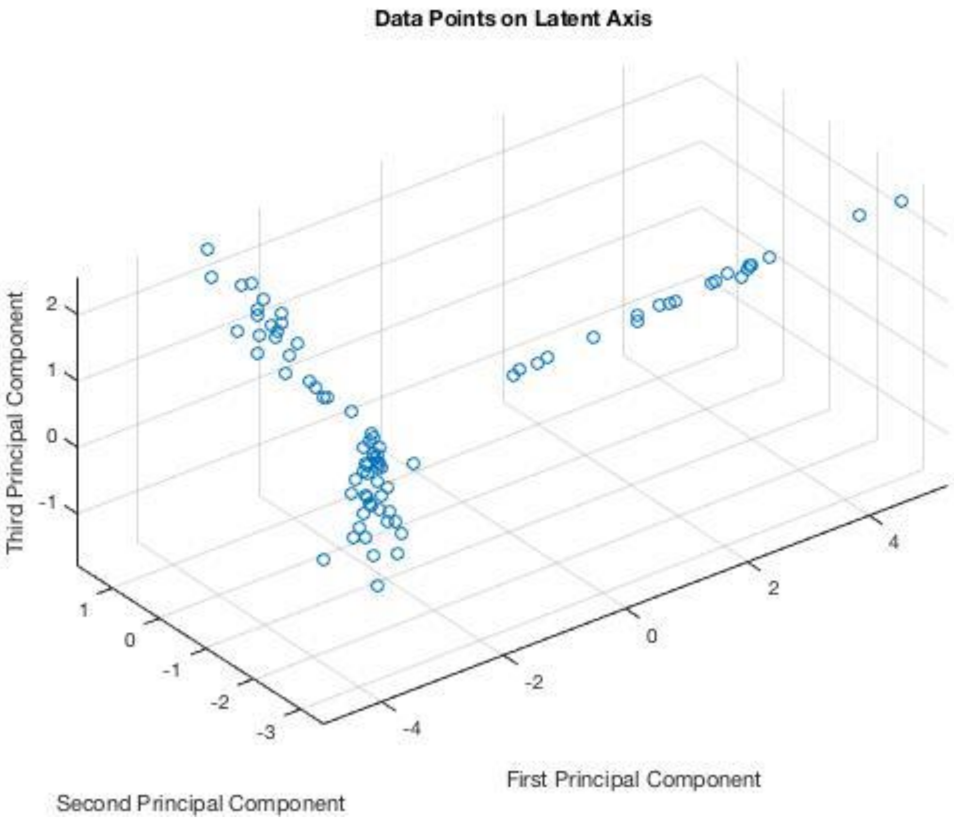


e) For **Task 3 PCA**, the core matrix and the feature matrix extracted are from PCA against the latent semantics. We are Plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics. Also, we are plotting the spider plot for eigen vectors obtained from PCA.

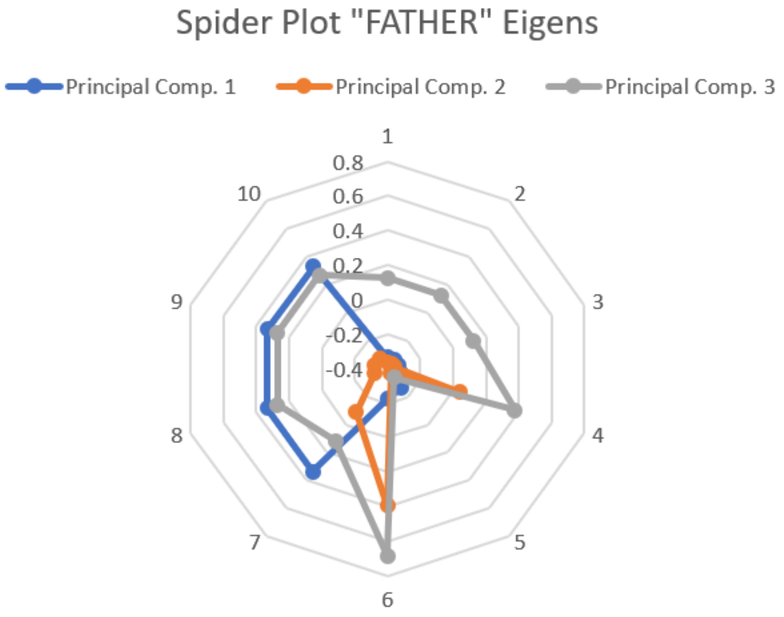
f) With our intuition for "Father" action, the PCA output was able to preserve 93% of variance using the first three eigen vectors. Holistically, we can prune the feature matrix to Nx3 and still preserve 93% variance in the combined dataset.

g) The relevant Matlab code can be found in Father_VAR.m. Below are the graph plots for the above-mentioned plot.

PCA



Spider Plot :



(

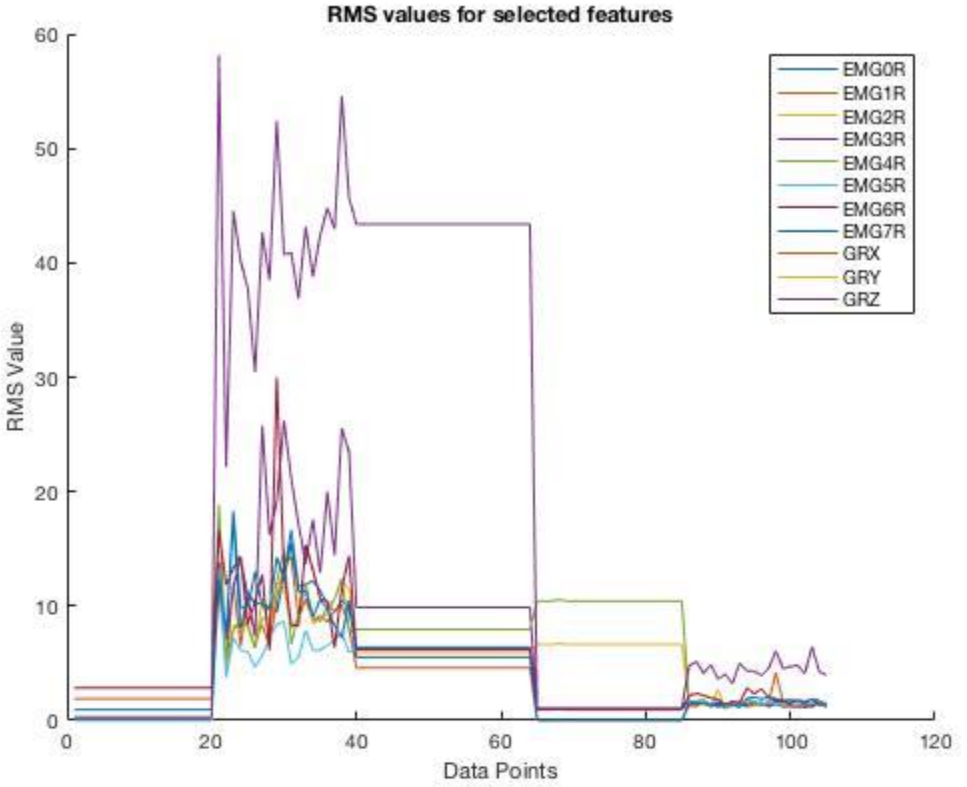
3.3.7 Find

For feature extraction for Find action we are using **Root Mean Square**.

a) Generally, Root mean square is used to measure the typical magnitude of the set of features, that is, computed as the square root of the arithmetic/quadratic mean of the squares of data points. Typically, is the measure of imperfection in the data set which is used to distinguish data points.

b) In our case, the data from the various sensors is in form of signal waves, hence applying an RMS on it will discriminate features from one another.

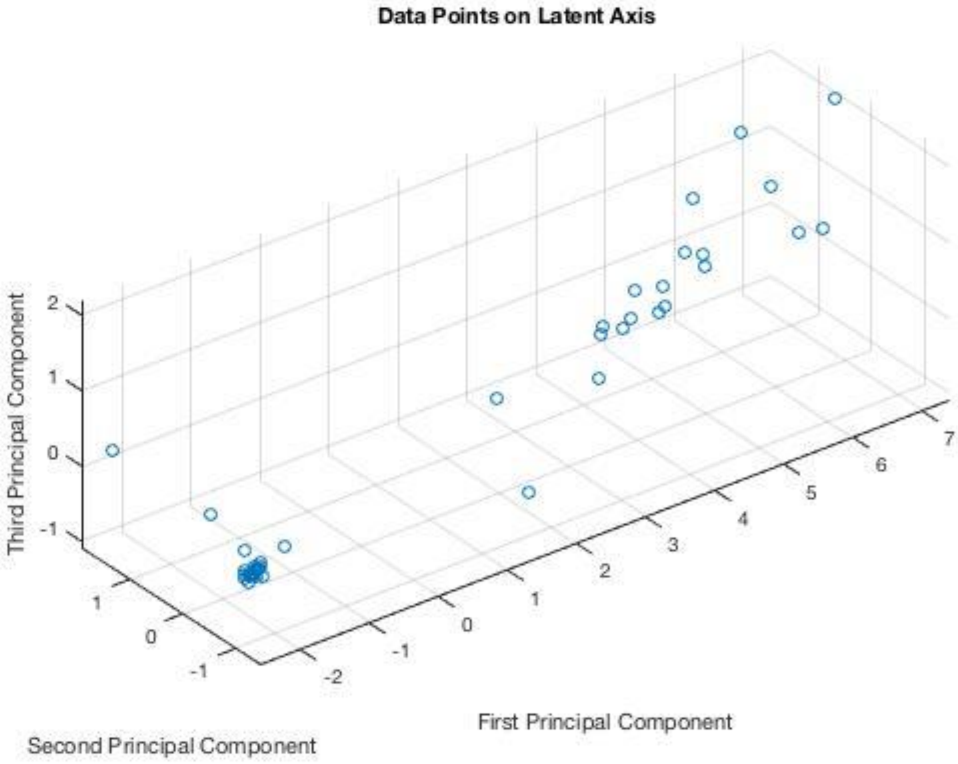
c) From our observation and intuition, the "FIND" ASL action has distinctive EMG and Gyroscope change in the Right hand which is relatively easy to differentiate among the other features.



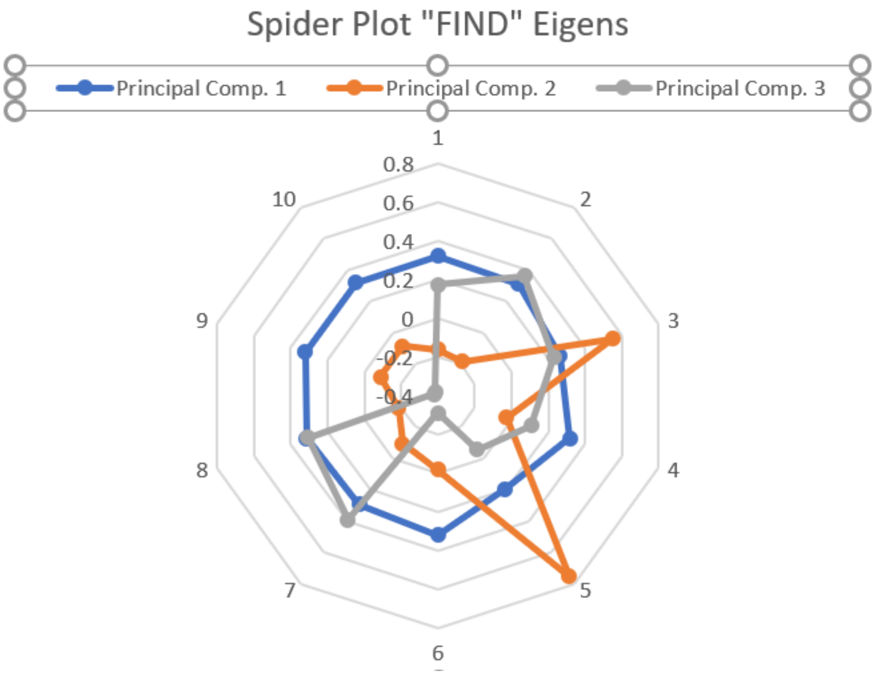
d) For **Task 3 PCA**, the core matrix and the feature matrix are extracted from PCA against the latent semantics, thereby plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of data points in Latent Semantics. Also, we are plotting the spider plot for eigenvectors obtained from PCA.

e) With our intuition for "Find" action, the PCA output was able to preserve 96% of variance using the first three eigenvectors. Holistically, we can prune the feature matrix to Nx3 and still preserve 96% variance in the combined dataset.

f) The relevant Matlab code can be found in Find_RMS.m. Below are the graph plots for the above-mentioned plot.
PCA :



Spider Plot :



3.3.8 Go Out

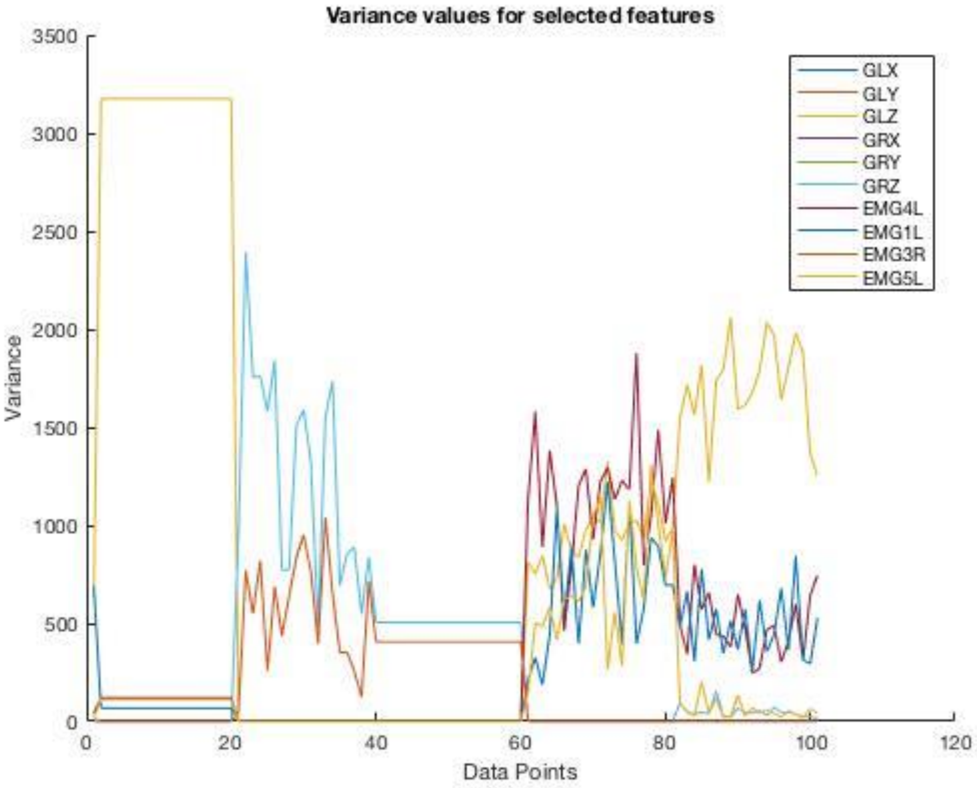
For feature extraction for Go Out action we are using **Variance**.

a) When it comes to feature extraction, variance plays a critical role. In our case, we take 34 degrees-of-freedom to measure each ASL action which makes it more important to distinguish features which are action specific from the ones which are less informative. Each data segment includes 6 accelerometer data (3 for each hand), 16 EMG Data (8 for each hand), 6 gyroscope data (3 for each hand) and 6 orientation data (3 for each hand). So, by close observation of the data for each action, we tend to extract the feature variance matrix which is of the order Mx34.

b) Out of intuition and set of logic, we choose 7 degrees-of-freedom, in other words, the feature that contributes to maximum variance to distinguish ASL actions. "Go Out" action tends to produce more variance with respect to the right-hand EMG and Gyroscope data (GRX, GRY,GRZ, EMG 1L, EMG 5L, EMG 3R, EMG 4R) with fixed left hand. This helps in shortlisting the degrees-of-freedom for "Go Out" action.

c) This variance among data segments can be computed with ease by the MatLab inbuilt feature *var()* which results in precise outcomes.

d) Also, some features do not produce clearer variance between actions. Hence it is important to isolate such data segments to reduce false positives and true negatives.

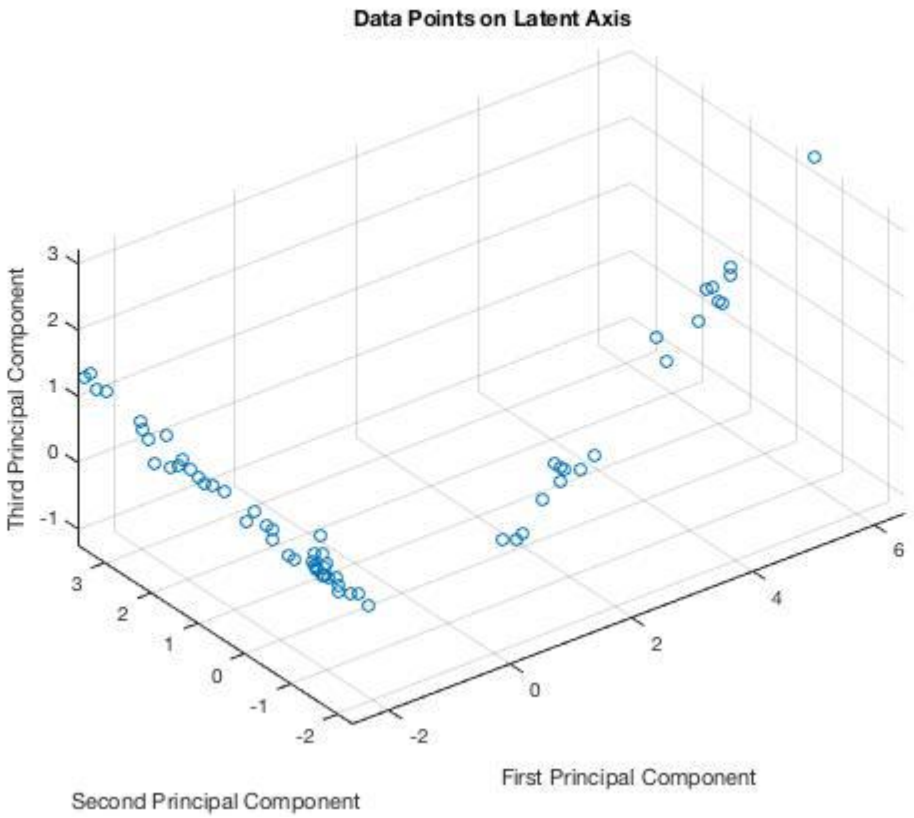


e) For **Task 3 PCA**, the core matrix and the feature matrix are extracted from PCA against the latent semantics. We are Plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics. Also, we are plotting the spider plot for eigen vectors obtained from PCA.

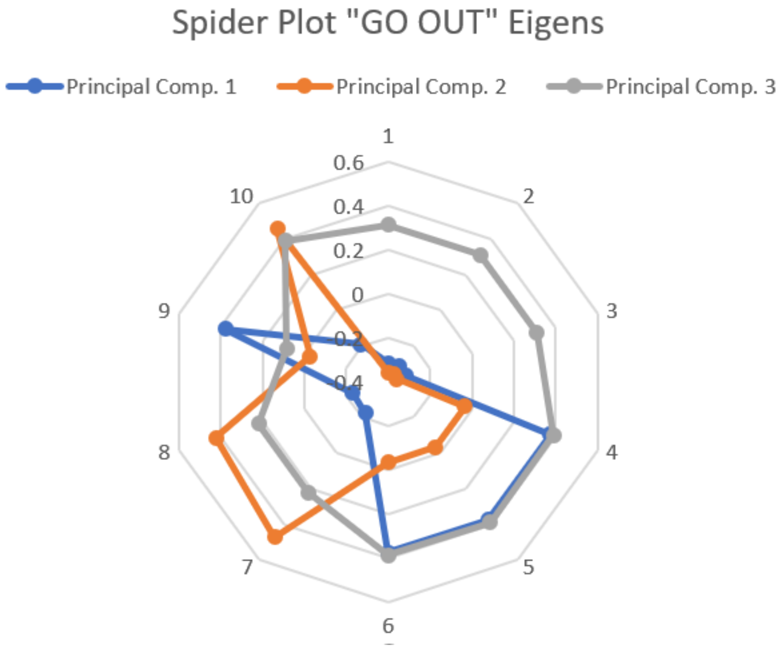
f) With our intuition for Go Out action, the PCA output was able to preserve 92% of variance using the first three eigen vectors. Holistically, we can prune the feature matrix to Nx3 and still preserve 92% variance in the combined dataset.

g) The relevant Matlab code can be found in GoOut_VAR.m. Below are the graph plots for the above-mentioned plot.

PCA :



Spider Plot :



3.2.9 Hearing

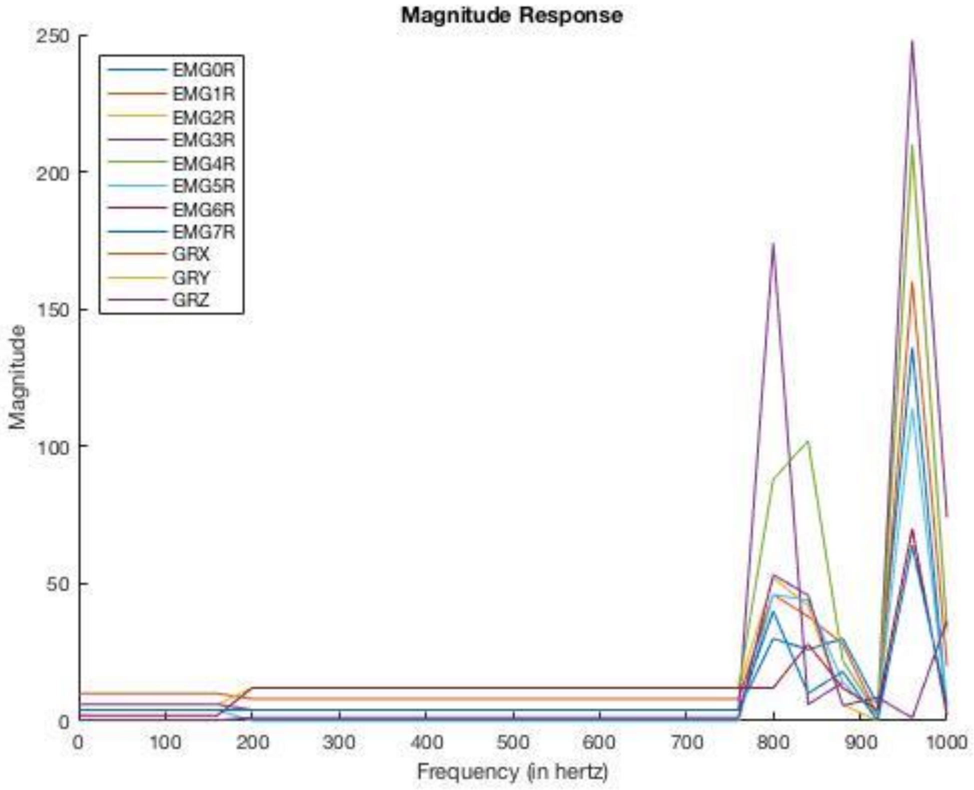
For feature extraction for Hearing action we are using **Fast Fourier Transform**.

a) The raw signals collected from the sensor during the phase 1 is converted into frequency components via Fast Fourier transform. The main idea of using Fourier transform is to identify the ASL action using the spikes in certain features. The advantage of Fast Fourier Transfer is that number of computations is less when compared to other transformations. We are using MATLAB's in-built function for Fast Fourier Transform. The way FFT operates is by decomposing an N point time domain signal into N time domain signals each containing a single point. Next, we are calculating the frequency corresponding to domain signals. For this phase, we consider 18 features from Gyroscope, Accelerometer, Orientation and EMG sensors.

b) For the "HEARING" ASL action applying FFT to predict what feature values shows variation for the movement. When trying to extract the features manually from the data it looks like there is change in EMG 0R – 7R, GRX, GRY, GRZ as for HEARING action there is rotation in right hand. We feed in all these feature parameters for FFT to calculate the frequency.

c) As a first step, extracting the CSV file of the ASL Hearing (Hearing.csv), which we got it from the output of task 1. The feature index values are specified to keep track of the value peaks for these specified features. Iterate over every index and send the values for FFT transformation. Consider all possible values of teams 12, 5, 2, 4 and 3 for a feature.

d) Normalizing the data with Z- order transformation. This is to ease the plotting. Plot the Frequency obtained from the Fourier transform against the magnitude and observe the peaks for the action.

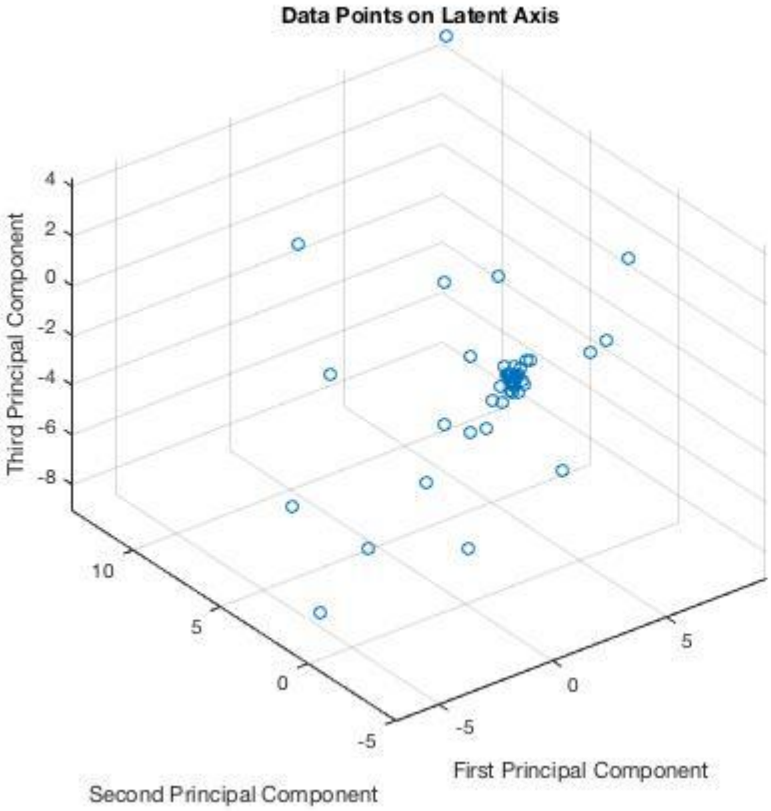


e) For **Task 3 PCA**, the Z - order values are passed to PCA to retrieve the core matrix and the feature matrix against the latent semantics. Plotting the data points in the First, Second and Third Principal Component Axis to get a better understanding of Data Points in Latent Semantics.

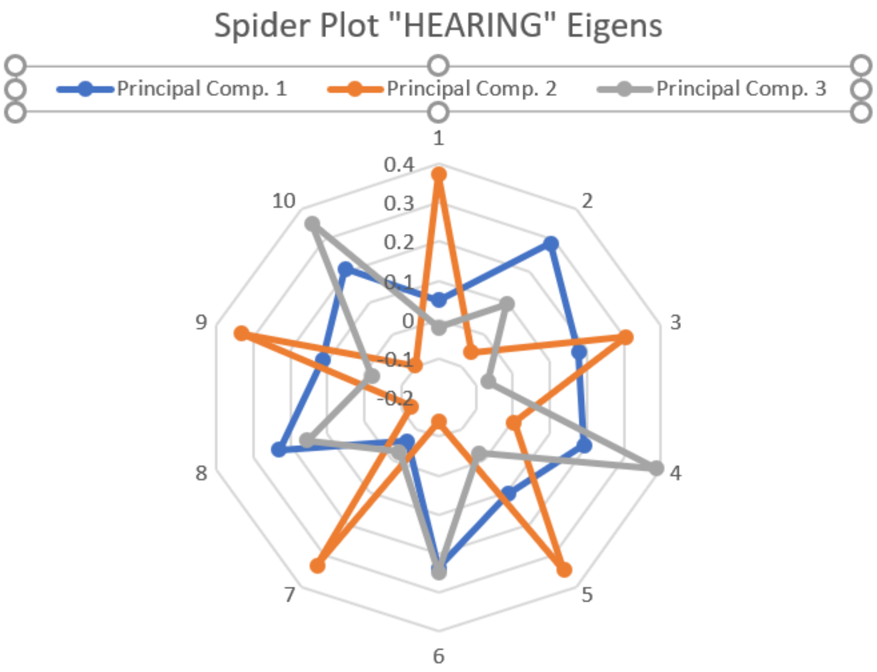
f) Our intuition for features of Hearing gesture is completely justified because we observe from the explained parameter of the PCA output that we are able to preserve 86% of the variance using the first three eigen vectors. PCA was certainly helpful in this scenario. We can reduce the feature matrix to NX3 and still preserve 86% variance in the combined dataset. The features which we have selected if it has little variance then we can't use that feature to explain this action. As our selected features have good variance we are able to get justifiable results

g) The relevant Matlab code can be found in Decide_FFT.m. Below are the graph plots for the above-mentioned plot.

PCA :



Spider Plot :



3.2.10 Can

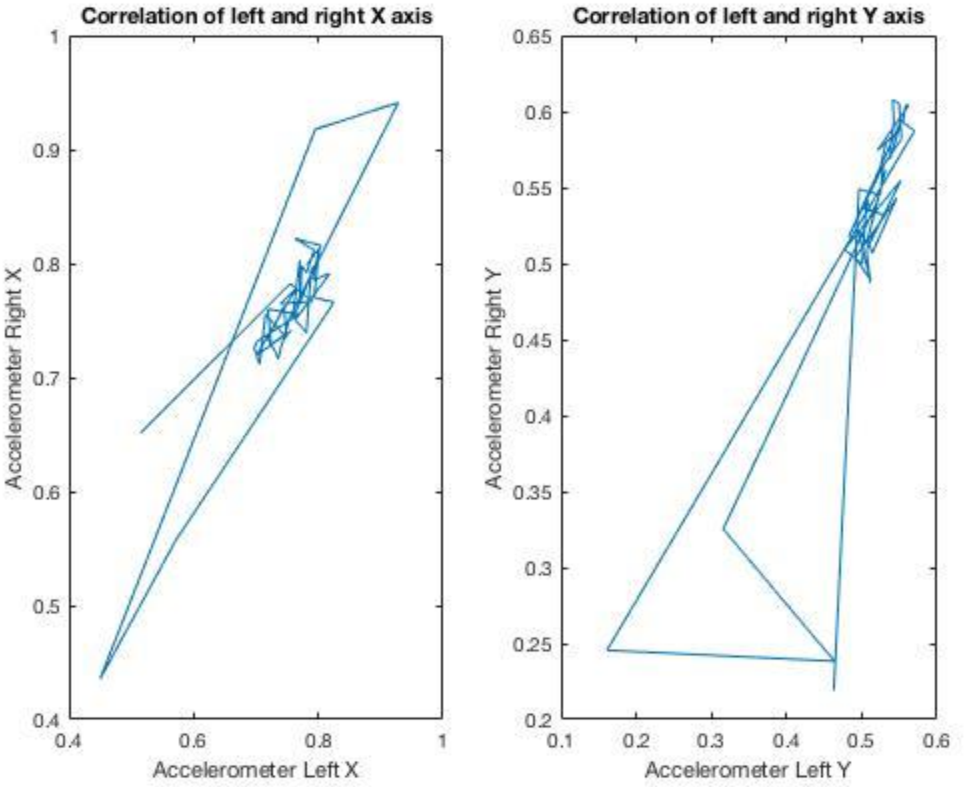
For feature extraction for Hearing action we are using **Pearson Correlation Coefficient**.

a) For the “CAN” ASL action, we are applying PCC to predict what feature values shows variation for the movement. When trying to extract the features manually from the data it looks like there are high fluctuations in Accelerometer data, as for CAN action there is a up down movement of both right and left hands. The movement of one hand is in sync with other hand, so the two sensor values ALX and ARX should be highly correlated.

b) As a first step, extracting the CSV file of the ASL Action (Can.csv), which we got it from the output of task. The feature index values are specified to keep track of the value peaks for these specified features. Iterate over every index and calculate the values for RMS.

c) Now calculate the PCC for ALX and ARX, and for ALY and ARY. We get the following PCC values:
PCCX = 0.852
PCCY = 0.79

d) The high values of PCC suggest that our intuition about the related accelerometer data for right and left hands is in the correct direction.
The above can be explained by the graphs ALX vs ARX and ALY vs ARY.

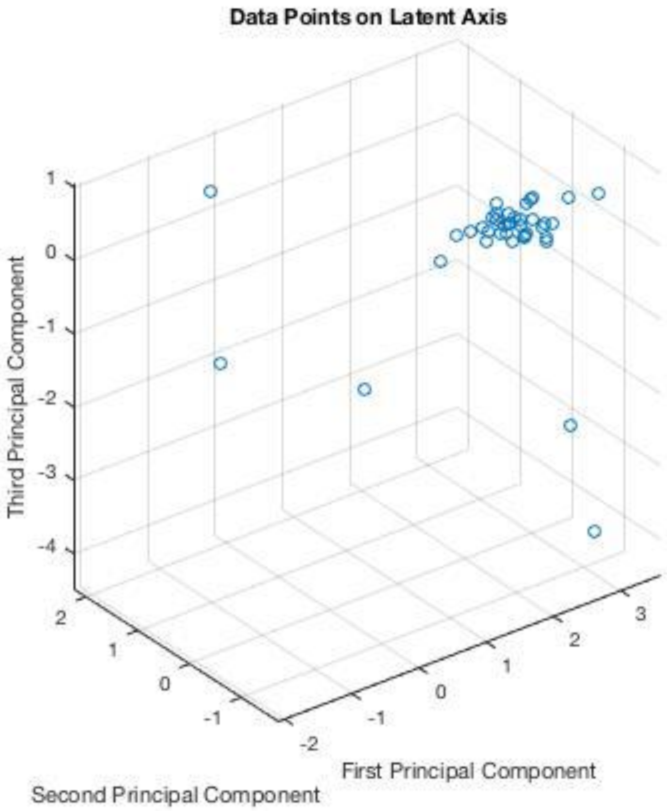


e) For **task 3** for "CAN" we need to take the PCA and discuss about the correctness of the intuition about selecting a perticular feature. To do this we are normalizing the values using z-score function and then getting coefficient matrix from the PCA output. Now we verify the PCA results by multiplying eigen vectors with the old feature matrix.

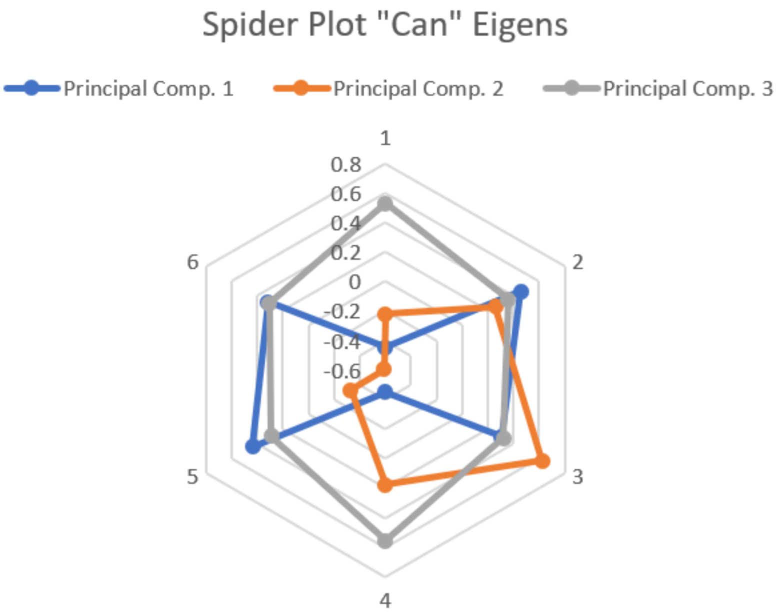
f) Now plot the Data Points in the new feature space. Also plot the eigen vectors in a spider plot.

g) Our intuition for features of "CAN" gesture is perfectly justified because we observe from the explained parameter of the PCA output that we are able to preserve 90.34% of the variance using the first three eigen vectors. PCA was certainly helpful in this scenario. We can reduce the feature matrix to NX3 and still preserve 98.5% variance in the dataset.

h) The relevant Matlab code can be found in Can_PCC.m . Below are the graph plots for the CAN gesture.
PCA :


Data Points on Latent Axis
Third Principal Component
1
0
-1
-2
-3
-4
Second Principal Component
2
1
0
-1
First Principal Component
-2
-1
0
1
2
3

Spider Plot :


Spider Plot "Can" Eigens
Principal Comp. 1
Principal Comp. 2
Principal Comp. 3
1
2
3
4
5
6
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6

4 Task 3: Feature Selection – General Explanation of PCS

In this Task, we analyze the feature extraction methods we chose as per our intuition in Task 2 and analyze which features are responsible the most variance in the data set, according to Principle Component Analysis.

4.1 Arranging Feature Matrix

The goal of this subtask is to organize all of our extracted features into a 2-D matrix. This matrix should be of dimension NxM, where N is the number of times the action is performed and M is the total number of features we have extracted. We have provided the dimensions of each individual feature matrix in Task 2. PCA needs to be computed over the feature matrix. We multiply the Eigen Vectors with the old feature matrix to get the new feature matrix.

4.2 Execution of PCA

The eigenvectors of the top three principal components are shown in the respective actions using Spider Plot.

4.3 Making Sense of Eigen Vectors

The features with the highest coefficients can be extracted from the spider plot. For example in "about" action, the first principal component is able to preserve 75.1% of the variance, while adding the second eigenvector increases this value to 85%. This is how we confirm if our initial intuition is correct or not.

4.4 Results of PCA

We create the new feature matrix by multiplying the old feature matrix with the eigenvectors.

4.5 Final Conclusion of PCA

PCA was certainly helpful in most of the scenarios. We reduce our feature matrix size from NxM to Nx3 and still capture more than 75% of the variance in almost all the cases as shown in Figure 2 of respective actions. We plot the data points in the 3-D latent feature space to show the variance along each of the principal component axes. This is important since features with variance are required to distinguish features. If a feature has very low variance, then it can’t be used to distinguish observations because it doesn’t convey us anything. But if a feature does change and thus exhibits variance then it can. If we use the top three principal components, then we now preserve about 90% of the variance set with an Nx3 feature matrix in almost all actions. We can observe that a small number of features (as determined by the size of coefficients in the principal components) can be used to represent a huge data.