EMG Synergy Extraction

RESEARCH PAPER

Academic year summar 2023

Using Electromyography data for non-invasive naturally-controlled robotic hand prostheses

Waleed Alzamil

Under the supervision of Eng.Hamdy Osama

Contents

Task1: Get Angles

About

In this work we use the Ninapro (Non Invasive Adaptive Prosthetics) database (Data Citations 2)

All the work can be found here

Data description

1. subject: subject number
2. exercise: exercise number
3. acc (36 columns): three-axes accelerometers of the 12 electrodes
4. emg (12 columns): sEMG signal of the 12 electrodes
5. glove (22 columns): uncalibrated signal from the 22 sensors of the cyberglove
6. inclin (2 columns): signal from the 2 axes inclinometer positioned on the wrist
7. stimulus (1 column): the movement repeated by the subject.
8. restimulus (1 column): again the movement repeated by the subject. In this case the duration of the movement label is refined a-posteriori in order to correspond to the real movement
9. repetition (1 column): repetition of the stimulus
10. rerepetition (1 column): repetition of restimulus
11. force (6 columns): force recorded during the third exercise
12. forcecal (2 x 6 values): the force sensors calibration values, corresponding to the minimal and the maximal force.

And the goal of this task is to find the angles of the fingers from the glove (22 sensor) data.

Research

The glove is equipped with up to 22 sensors, including **bend** sensors and **abduction** sensors.

Sensor description of these sensor types

1. **Bend Sensors**: These are the majority of the sensors in the glove. They are located over the knuckles and joints of the fingers and thumb. These sensors measure the amount of bend, or flex, in the joint. The sensor works by changing its resistance as it is bent. The more the sensor is bent, the higher the resistance. This change in resistance is measured and converted into a digital signal that represents the degree of bend.
2. **Abduction Sensors**: These sensors are located between the fingers and measure the angle between adjacent fingers. They work on the same principle as the bend sensors, changing their resistance as the angle between the fingers changes.

The raw data from these sensors are typically in the range of 0 to 255 (8-bit data), representing the minimum to maximum bend or abduction.

Mapping raw data from the CyberGlove to joint angles involves a process of normalization.

1. **Normalization**: The raw data from the CyberGlove sensors are typically in the range of 0 to 255 (8-bit data). To map these to joint angles, data need to be normalized. This involves subtracting the minimum sensor reading from the current reading, and then dividing by the range of sensor readings. This will give a normalized value between 0 and 1.

$$\text{Normalized Data} = \frac{\text{Raw Data} - \text{Min Data}}{\text{Max Data} - \text{Min Data}} \tag{1}$$

2. **Mapping to Angles**: Map this to joint angles. Assume that the joint angles range from 0 to a maximum angle (for example, 0 to 180 degrees for a finger joint), multiply the normalized data by the maximum angle to get the joint angle.

$$\text{Joint Angle} = \text{Normalized Data} \times \text{Max Angle} \tag{2}$$

NOTE, the exact mapping from raw data to joint angles may vary depending on the specific CyberGlove model and the individual wearing the glove. It's important to perform a thorough calibration process to ensure accurate results. Maybe we will need more complxe model to deal with the non-linearity in the sensors.

Acquisition setup

The acquisition setup included several sensors, designed to record hand kinematics, dynamics and the corresponding muscular activity. The sensors were connected to a laptop responsible for data acquisition. Hand kinematics was measured using a 22-sensor CyberGlove II dataglove (CyberGlove Systems LLC). The CyberGlove is a motion capture data glove, instrumented with joint angle measurements. It uses proprietary resistive bend-sensing technology to accurately transform hand and finger motions into real-time digital joint-angle data. The device returns 22 8-bit values proportional to these angles for an average resolution of less than one degree depending on the size of the subject's hand.

Max Angles

The human hand is a complex system of bones, muscles, and joints. Each joint has a specific range of motion, which is the measure of the maximum angle that the joint can move in a particular direction.

The range of motion for each joint in the human hand varies based on several factors, including the individual's age, sex, and overall health.

General Ranges that can be used as a guideline:

1. **Metacarpophalangeal (MCP) Joints**: These are the knuckles where the fingers meet the hand. The MCP joints can flex (bend) to approximately 90 degrees and extend (straighten) to about 30 degrees.
2. **Proximal Interphalangeal (PIP) Joints**: These are the middle joints of the fingers. The PIP joints can flex to approximately 100-120 degrees.
3. **Distal Interphalangeal (DIP) Joints**: These are the joints closest to the fingertips. The DIP joints can flex to approximately 80-90 degrees.
4. **Thumb Carpometacarpal (CMC) Joint**: This joint allows the thumb to move toward the palm (opposition). The CMC joint can flex to approximately 15 degrees and extend to about 5 degrees.
5. **Thumb Metacarpophalangeal (MCP) Joint**: This joint allows the thumb to move up and down. The MCP joint can flex to approximately 50 degrees and extend to about 0 degrees.
6. **Thumb Interphalangeal (IP) Joint**: This joint allows the thumb to bend. The IP joint can flex to approximately 80 degrees.

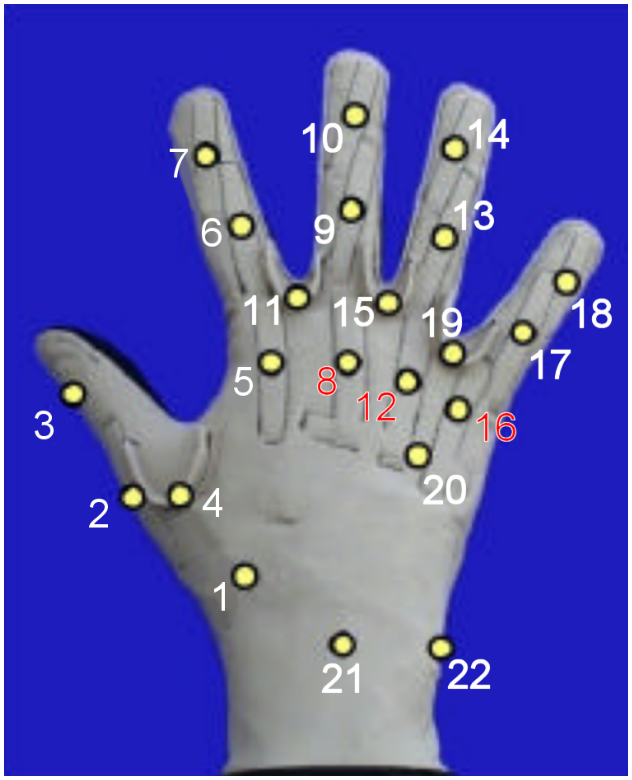


Figure 1: cyberglove

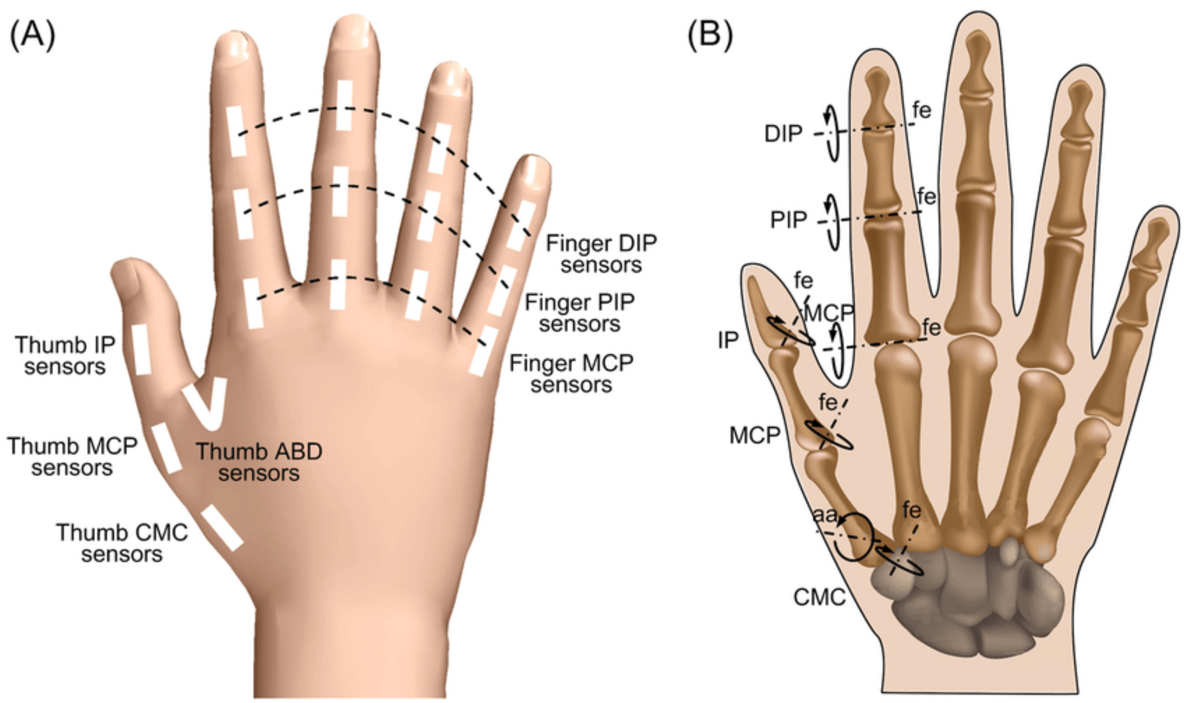


Figure 2: kinematic model of human hand

Problem

1. Datasheets which I found are useless
2. Issues with the original data Figure 3

index	count	mean	std	min	25%	50%	75%	max
glove_1	1808331.0	50.39799880981445	28.616668701171875	-23.671113967895508	34.77608108520508	54.258480072021484	63.99967956542969	142.9033966064453
glove_2	1808331.0	10.256361961364746	11.402695655822754	-18.94945526123047	5.205894470214844	9.370610237121582	15.201211929321289	69.34251403808594
glove_3	1808331.0	10.817085266113281	14.966017723083496	-21.063579559326172	1.6779863238334656	11.303863525390625	19.595340728759766	65.90132904052734
glove_4	1808331.0	4.590909481048584	14.567110061645508	-35.89632034301758	-2.790359497070312 5	1.939063310623169	9.506139755249023	52.07094573974609 4
glove_5	1808331.0	30.701602935791016	19.153051376342773	-25.831825256347656	21.88593292236328	37.48596954345703	42.074214935302734	68.68604278564453
glove_6	1808331.0	28.497417449951172	32.2488899230957	-22.392051696777344	-9.042371273040771	31.608074188232422	52.488121032714844	102.1682357788086
glove_7	1808331.0	13.172595024108887	7.800526142120361	-18.534347534179688	8.599761962890625	12.976231575012207	17.35270118713379	76.4350357055664
glove_8	1808331.0	42.84450149536133	25.043014526367188	-18.24286651611328	25.64004611968994	48.91493606567383	57.936134338378906	98.0303497314453
glove_9	1808331.0	45.46136474609375	50.24419403076172	-44.53389358520508	-11.498798370361328	54.5716476440429 7	83.79511260986328	156.2182159423828
glove_10	1808331.0	4.677371978759766	7.992690563201904	-33.44902038574219	1.153414487838745 1	4.795775890350342	7.2240166664123535	61.25238037109375
glove_11	1808331.0	461.3294677734375	257.8214721679687 5	-561.1754760742188	427.0580749511719	497.6462097167969	582.3519287109375	892.9396362304688
glove_12	1808331.0	34.84281539916992	27.24970245361328	-33.17074966430664	24.664857864379883	37.64320755004883	45.99407386779785	112.18421173095703
glove_13	1808331.0	43.19628143310547	35.74754333496094	-17.8292179107666	14.789659023284912	49.30435562133789	71.41460037231445	106.87225341796875
glove_14	1808331.0	-0.7870416641235352	10.94864559173584	-21.700496673583984	-7.634825706481934	-2.2422194480895996	2.5800061225891113	77.85547637939453
glove_15	1808331.0	3.594827175140381	5.244267463684082	-18.889404296875	2.922351837158203	5.251762866973877	6.734115123748779	13.510582923889 16
glove_16	1808331.0	33.544708251953125	20.932373046875	-14.016033172607422	21.504051208496094	34.9440803527832	42.62409973144531	97.34423065185547
glove_17	1808331.0	30.282367706298828	33.53587341308594	-26.391592025756836	2.813242197036743	37.4767951965332	55.09565734863281	90.33338928222656
glove_18	1808331.0	4.437711715698242	9.064489364624023	-10.249415397644043	-1.2141180038452148	1.6094123125076294	7.256473064422607	47.91530990600586
glove_19	1808331.0	13.991806030273438	1.7681303024291992	2.985877275466919	13.7152709960937 5	14.279975891113281	14.844680786132812	19.079967498779297
glove_20	1808331.0	15.404114723205566	18.994918823242188	-40.749271392822266	6.135543823242187 5	17.7120418548584	28.08722686767578	59.96625900268555
glove_21	1808331.0	9.016514778137207	14.123442649841309	-94.94100952148438	5.294108390808105 5	10.626374244689941	15.176444053649902	54.70578765869140 6
glove_22	1808331.0	13.119962692260742	10.897356033325195	1.6941215991973877	7.157502412796020 5	12.705911636352539	16.094154357910156	112.6590881347656 2

Figure 3: Original Data describtion

index	count	mean	std	min	25%	50%	75%	max
glove_1	1808331.0	0.44466039538383484	0.17179499566555023	0.0	0.35087719559669495	0.46783626079559326	0.5263157486915588	1.0
glove_2	1808331.0	0.3307868242263794	0.12914758920669556	0.0	0.27358490228652954	0.3207547068595886	0.3867924213409424	1.0
glove_3	1808331.0	0.36659228801727295	0.17209260165691376	0.0	0.26150280237197876	0.3721897006034851	0.46753251552581787	1.0
glove_4	1808331.0	0.4602533280849457	0.16559693217277527	0.0	0.37634408473968506	0.4301075339317322	0.5161290168762207	1.0
glove_5	1808331.0	0.5981243848800659	0.20263947546482086	0.0	0.5048543810844421	0.6699029207229614	0.7184465527534485	1.0
glove_6	1808331.0	0.4085530638694763	0.2589018642902374	0.0	0.10717444866895676	0.43352600932121277	0.6011560559272766	1.0
glove_7	1808331.0	0.33386480808258057	0.08213727176189423	0.0	0.2857142984867096	0.3317972421646118	0.37788018584251404	1.0
glove_8	1808331.0	0.5253776907920837	0.21538077294826508	0.0	0.3774120658636093	0.5775862336158752	0.6551724076271057	1.0
glove_9	1808331.0	0.4482905864715576	0.2502797842025757	0.0	0.16455665230751038	0.4936712682247162	0.6392411589622498	1.0
glove_10	1808331.0	0.4025959372520447	0.08439886569976807	0.0	0.36538463830947876	0.4038461446762085	0.4294871985912323	1.0
glove_11	1808331.0	0.7031802535057068	0.1773047149181366	0.0	0.6796116232872009	0.7281553149223328	0.7864077091217041	1.0
glove_12	1808331.0	0.46791353821754456	0.18747004866600037	0.0	0.39789222180843353	0.4871794879436493	0.5446310341358185	1.0
glove_13	1808331.0	0.4893726110458374	0.2866649627685547	0.0	0.26157572865486145	0.5383543372154236	0.7156597077846527	1.0
glove_14	1808331.0	0.21006736159324646	0.10997477918863297	0.0	0.1412840485572815	0.19545063376426697	0.24388796091079712	1.0
glove_15	1808331.0	0.6939576864242554	0.16186018288135529	0.0	0.673202633857727	0.7450980544090271	0.7908496856689453	1.0
glove_16	1808331.0	0.4270891547203064	0.1879698634147644	0.0	0.31896552443504333	0.4396551549434662	0.5086206793785095	1.0
glove_17	1808331.0	0.4855341613292694	0.287306725978851 3	0.0	0.25020208954811096	0.5471698045730591	0.698113203048706	1.0
glove_18	1808331.0	0.25250929594039917	0.15584169328212738	0.0	0.155339807271957 4	0.20388349890708923	0.3009708523750305	1.0
glove_19	1808331.0	0.6838492751121521	0.10986208915710449	0.0	0.6666667461395264	0.7017544507980347	0.736842155456543	1.0
glove_20	1808331.0	0.557544469833374	0.18859970569610596	0.0	0.46551722288131714	0.5804597735404968	0.6834744811058044	1.0
glove_21	1808331.0	0.6946861743927002	0.0943785235285759	0.0	0.6698113083839417	0.7054436802864075	0.7358490824699402	1.0
glove_22	1808331.0	0.10296802222728729	0.09820537269115448	0.0	0.04923518747091293	0.0992366373538971	0.12977097928524017	1.0

Figure 4: Normalized Data describtion

3. the stimulus has numbers from 0 to 17 which means we have 18 Figure 6 movements but in fact we just have 17 Figure 5 so what does this mean??

   After asking Eng.Hamdy Osama, he answered: ”the extra gesture (which is number 0) is meant for hand in rest state -¿ no gesture is being made”

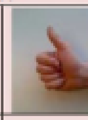
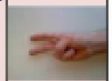
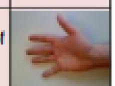
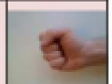
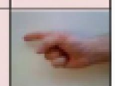
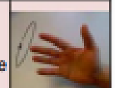
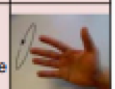
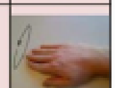
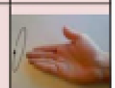
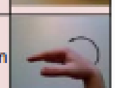
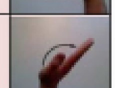
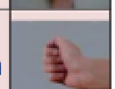
Exercise B		
1	Thumb up	
2	Extension of index and middle, flexion of the others	
3	Flexion of ring and little finger, extension of the others	
4	Thumb opposing base of little finger	
5	Abduction of all fingers	
6	Fingers flexed together in fist	
7	Pointing index	
8	Adduction of extended fingers	
9	Wrist supination (axis: middle finger)	
10	Wrist pronation (axis: middle finger)	
11	Wrist supination (axis: little finger)	
12	Wrist pronation (axis: little finger)	
13	Wrist flexion	
14	Wrist extension	
15	Wrist radial deviation	
16	Wrist ulnar deviation	
17	Wrist extension with closed hand	

Figure 5: movements in the Original Data

```
pd.unique(stimulus1.iloc[:,0])
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17], dtype=int8)
```

Figure 6: encoded movements in the Original Data

My approach now

1. give for each joint the approximated max angle then Applying equation 2
2. figure out the type of each sensor in the glove (Bend or Abduction)
3. try to figure out why there are values greater than 255 in the glove data

Results

- glove {7, 10, 14, 18} -> DIP finger sensors -> 90
- glove {6, 9, 13, 17} -> PIP finger sensors -> 120
- glove {11, 15, 19} -> MCP finger sensors -> 90
- glove {4, 5, 8, 12, 16, 20, 21, 22} -> are not finger sensors
- glove 3 -> IP thump sensors -> 80
- glove 2 -> MCP thump sensors -> 50
- glove 1 -> MCP thump sensors -> 15

Figure 7: Max angles

	angles_glove_1	angles_glove_2	angles_glove_3	angles_glove_4	angles_glove_5	angles_glove_6	angles_glove_7	angles_glove_8	angles_glove_9
0	7.807017	14.622641	32.207797	0.0	0.0	45.086707	22.396314	0.0	48.607618
1	7.807017	14.622641	32.207797	0.0	0.0	45.086707	22.396314	0.0	48.607618
2	7.807017	14.622641	32.207797	0.0	0.0	45.086707	22.396314	0.0	48.607618
3	7.807017	14.622641	32.207797	0.0	0.0	45.086707	22.396314	0.0	48.607618
4	7.807017	14.622641	32.207797	0.0	0.0	45.086707	22.396314	0.0	48.607618
...	...	...	...	...	...	...	...	...	...
1808326	7.192861	15.566038	20.779223	0.0	0.0	83.236992	41.474655	0.0	80.506411
1808327	7.191763	15.566038	20.779223	0.0	0.0	83.236992	41.474655	0.0	80.506411
1808328	7.190666	15.566038	20.779223	0.0	0.0	83.236992	41.474655	0.0	80.506411
1808329	7.189569	15.566038	20.779223	0.0	0.0	83.236992	41.474655	0.0	80.506411
1808330	7.188471	15.566038	20.779223	0.0	0.0	83.236992	41.474655	0.0	80.506411

1808331 rows × 22 columns

Figure 8: sample of the E1

Does the results of the angle modelling make sense?

```
E1.iloc[4318 : 4320].T # thump up
```

	4318	4319
angles_glove_1	6.099506	6.098557
angles_glove_2	6.603774	6.603774
angles_glove_3	13.506495	13.506495
angles_glove_4	0.000000	0.000000
angles_glove_5	0.000000	0.000000
angles_glove_6	97.014470	97.044489
angles_glove_7	40.008787	40.013275
angles_glove_8	0.000000	0.000000
angles_glove_9	95.238261	95.262916
angles_glove_10	37.768636	37.774875
angles_glove_11	58.076767	58.086213
angles_glove_12	0.000000	0.000000
angles_glove_13	99.567168	99.599354
angles_glove_14	23.340755	23.334819
angles_glove_15	61.724276	61.737000
angles_glove_16	0.000000	0.000000
angles_glove_17	93.884454	93.908944
angles_glove_18	33.143821	33.162728
angles_glove_19	58.734391	58.760017
angles_glove_20	0.000000	0.000000
angles_glove_21	0.000000	0.000000
angles_glove_22	0.000000	0.000000

Figure 9: results of the angle modelling for movements 1

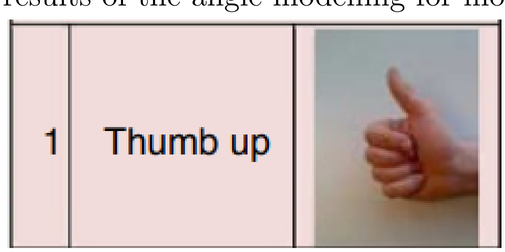


Figure 10: movements 1

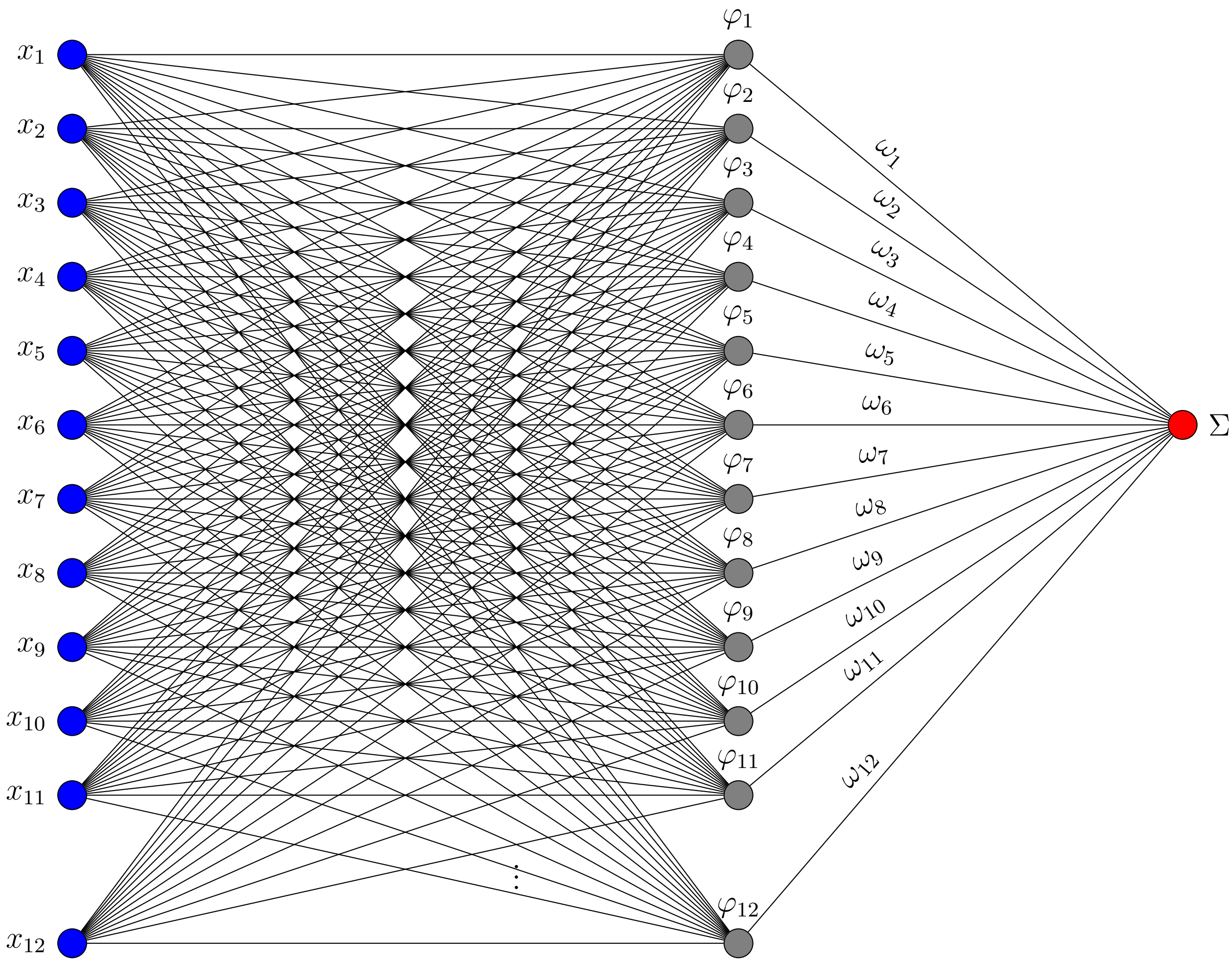
Task2: Build different architectures to predict the angles (Neural Networks)

prepare the data for training

We will reduce the dimentionality of the input signals (12 signal) but for instance we will use it as a whole to see the result. so the input layer will have 12 neuron

Drop glove 4, 5, 8, 12, 16, 20, 21, 22 and leave the rest now our neural networks has $22 - 8 = 14$ neuron in the output layer

DNN based layers



Note this is not the actual Architecture I used it's just for illustration

Training process

The results is so bad and the process is computationally expensive and I need to tune the Architecture so the following is not the final model but it's the base for further development

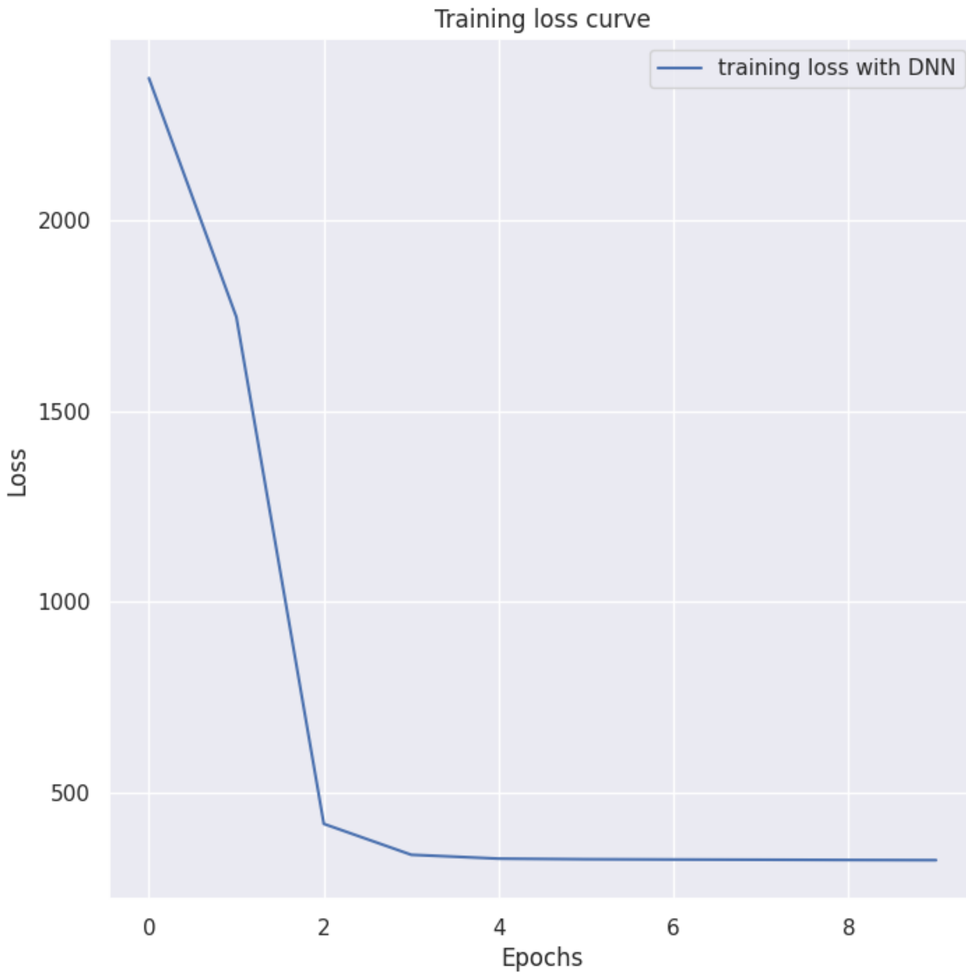


Figure 11: The Loss of this model During the training process

Sooooo bad!!

Our next step is to try CNNs and RNNs we need to prepare the data in some way to feed it to some architecture based on CNN or LSTM for example.

LSTM-DNN based layers

almost I used the same architecture in the previous model but 3 LSTM layers were added in the first of the architecture.
We need to consider preparing the data for the sequence model (padding to the maximum length and truncating to the minimum length) (using all the samples or using in only the samples of motion as this may make a big difference as the samples with no motion is half the data and thus due to the way the data collected)

Training process

The results are so bad and the process is computationally expensive and I need to tune the Architecture so the following is not the final model but it's the base for further development

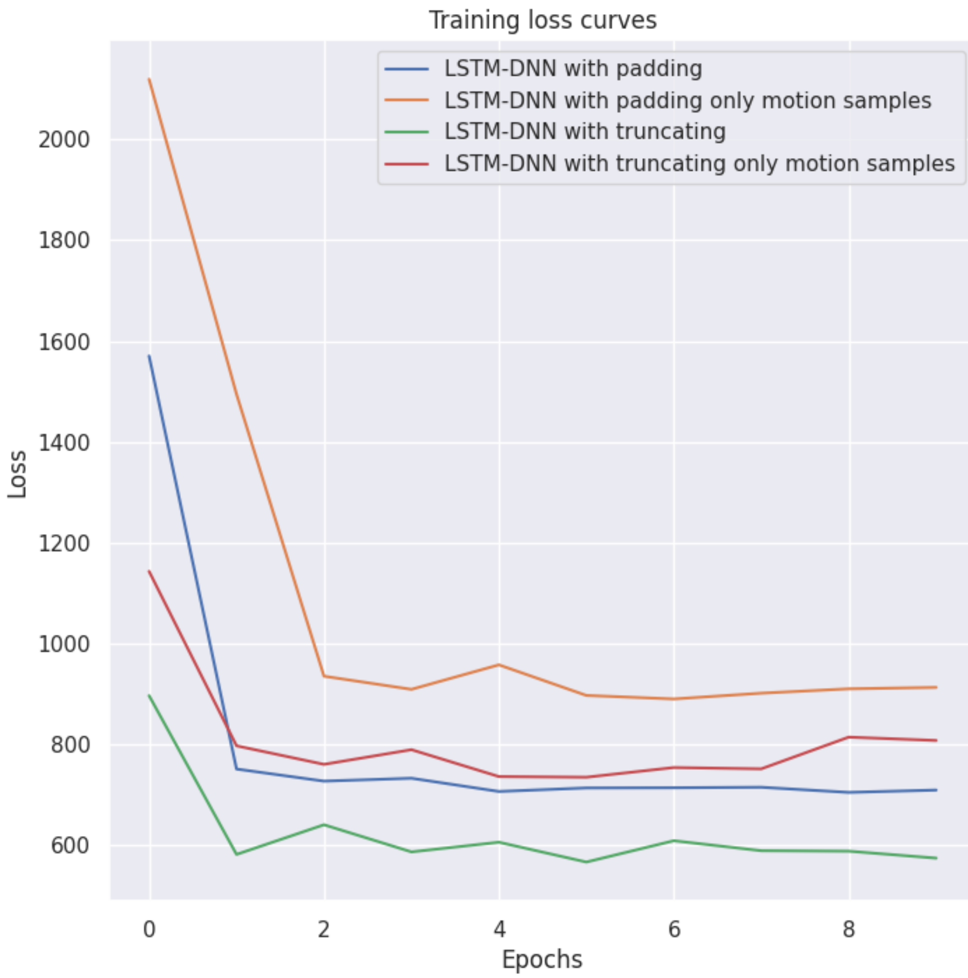


Figure 12: The Loss of this Model During the training process

Unbelievable!!

Worse than the DNN based layers!!

we need to consider other architectures like transformers (state-of-art)

conclustion

1. I tried to change and tweak the hyper-parameters in the architecture but nothing goes below loss = 300.
2. Tried to build a more complex architecture but it was in vain (no improvement).
3. The only thing to do is to try some transformer-based models but ...... From previous insights I can say that there is something wrong with the data
   a) Data is not sufficient
   b) Maybe my linear model for the angles is the problem but I don't think so.
   c) The data and the problem are so complicated and I couldn't distinguish between them using autoencoders in 2D space as can be shown in the following plot.

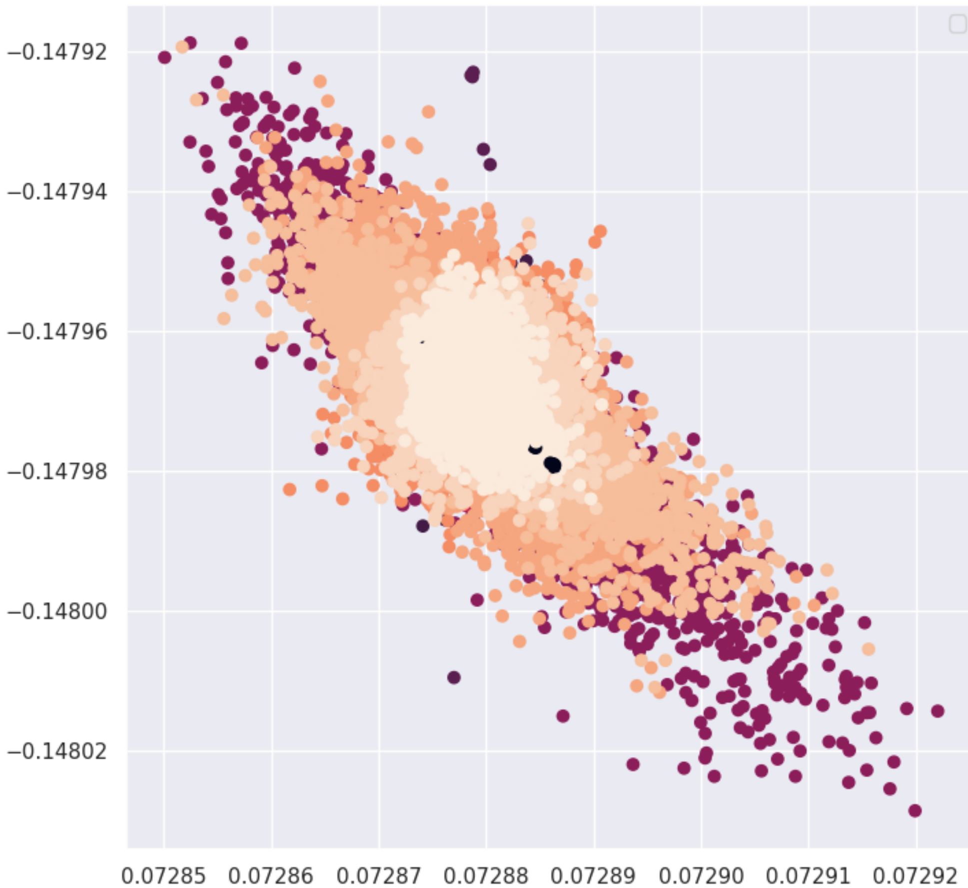


Figure 13: The Loss of this model During the training process

d) When applied the PCA I can see the data is so difficult to be presented in a low dimensional space.

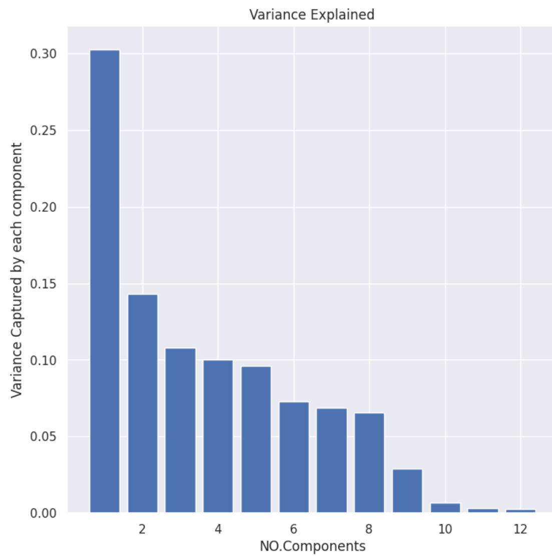


Figure 14: the variance explained by each principal component

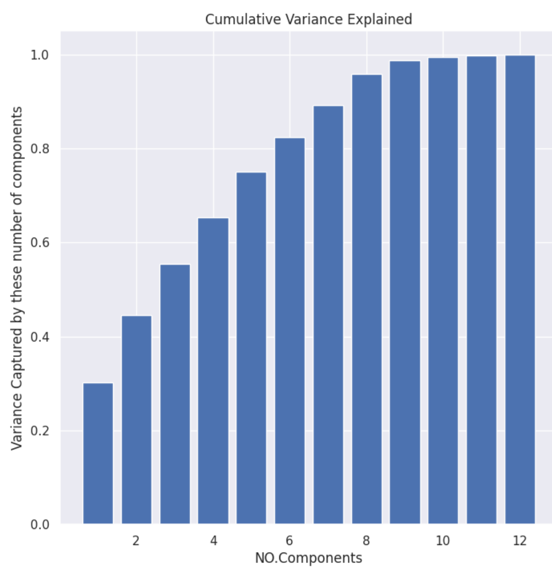


Figure 15: the Cumulative variance explained the principal components