PROSTHETIC VOICE

PROJECT REPORT

Submitted by

K PRATHIK 13EE139
SANJEEV N 13EE144
K SHASHANK HEGDE 13EE224

Under the guidance of

Dr. C M C KRISHNAN
Assistant Professor

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL
SRINIVASNAGAR, MANGALORE 575025
KARNATAKA, INDIA

APRIL 2017

Certificate of Completion

This is to certify that the project report titled “Prosthetic Voice” is a bonafide work carried out by:-

K Prathik 13EE139
Sanjeev N 13EE144
K Shashank Hegde 13EE224,

the students of VIII semester, B.Tech Electrical and Electronics Engineering, National Institute of Technology Karnataka Surathkal. This project has been completed under the guidance of Dr. C M C Krishnan, Assistant Professor, NITK Surathkal, during the academic year 2016-17 as Major Project - 2 (EE499).

Dr. C M C Krishnan
Assistant Professor,
Department of Electrical and Electronics Engineering
NITK

Abstract

Deaf-mute individuals have an arduous task in communicating to a normal individual if that person is unaware of sign language. This project aims to develop a product that will convert the signs shown by the deaf-mute individual to an artificially generated voice output that maps onto the respective sign shown through the phonetics in the message(s) conveyed. Here, SEMG signals are extracted to provide the basis for mapping to the phonetics. Electromyographic signals have been widely employed as a control signal in rehabilitation and a means of diagnosis in health care. Commercial computer software such as Matlab and CoolTerm/Arduino were used for data acquisition software development and data analysis.

Introduction

EMG is a nerve conducting test, performed by measuring the bioelectric signals from the muscle of a human body. Muscular movements cause the action of muscles and nerves, which provide electrical currents. These currents are generated by the interchange of ions across the muscles which make a part of the signalling process for the muscle fibres to contract [2]. It can be measured by applying conductive elements or electrodes to the skin surface, or invasively within the muscle. Measurement of surface EMG is dependent on the amplitude of the surface EMG signal. The signal varies from μV to mV range. Since the signal level is too low to capture on the display, it is required to amplify the signal level to a TTL level (between -5 volts to +5 volts). Many critical factors should be considered before the signal is displayed properly, such as the electrical signals are distorted by noises and artefacts. Additional DC current could also provide offset to the EMG signal. Without a proper ground reference, the signal could be misleading [1].

Forearm Muscles

The forearm is a structure on the upper limb. The forearm consists of two bones, the radius and ulna. It contains many muscles such as the extensor carpi radialis, extensor digitorum communis, extensor carpi ulnaris, flexor carpi radialis and a few more.

Extensor carpi radialis

The extensor carpi radialis is one of the five main muscles that control movements at the wrist. This muscle is quite long, starting on the lateral side of the humerus, and attaching to the base of the second metacarpal bone. As the name suggests, this muscle is an extensor at the wrist joint and travels along the radial side of the arm, so will also abduct (radial abduction) the hand at the wrist.

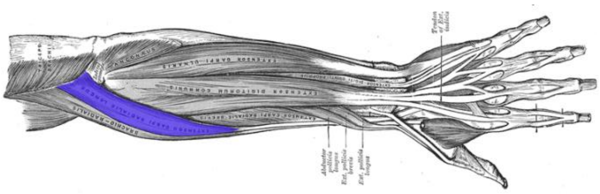

Fig 1: Extensor carpi radialis

Extensor digitorum communis

The extensor digitorum muscle (also known as extensor digitorum communis) is a muscle of the posterior forearm present in humans and other animals. It extends the medial four digits of the hand. The extensor digitorum communis extends the phalanges, then the wrist, and finally the elbow. It tends to separate the fingers as it extends them.

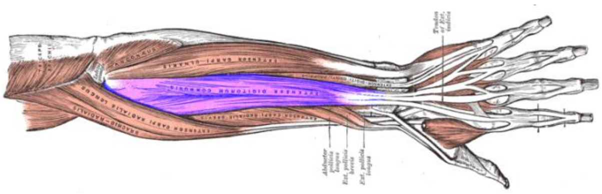

Fig 2: Extensor digitorum communis

Extensor carpi ulnaris

The extensor carpi ulnaris is a skeletal muscle located on the ulnar side of the forearm. It acts to extend and adduct at the carpus/wrist from anatomical position. The extensor carpi ulnaris extends the wrist, but when acting alone inclines the hand toward the ulnar side; by its continued action it extends the elbow-joint.

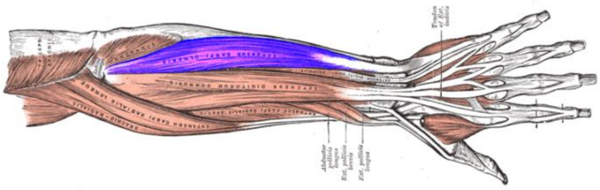

Fig 3: Extensor carpi ulnaris

Flexor carpi radialis

Flexor carpi radialis is a muscle of the human forearm that acts to flex and (radial) adduct the hand [1].

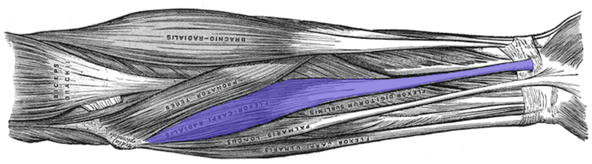

Fig 4: Flexor carpi radialis

Monitoring Muscle Activity

This muscle sensor from Advancer Technologies measures a muscle's activity by monitoring the electric potential generated by muscle cells. The sensor amplifies and processes the complex electrical activity of a muscle and converts it into a simple analog signal that can easily be read by any microcontroller with an analog-to-digital converter (ADC), such as an A-Star or Arduino – or even a Maestro servo controller. As the target muscle group flexes, the sensor's output voltage increases. The exact relationship between the output voltage and the muscle activity can be fine-tuned using an on-board gain potentiometer. In order to attach to skin, the sensor requires three electrodes that snap into the sensor's snap-style connectors, which make it easy to attach and detach electrodes. Two connectors are located directly on the PCB, and the third is located at the end of the attached reference electrode cable. The board's pins have a 0.1″ pitch and work with 0.1″ male headers and 0.1″ female headers [4]. The connection pattern to extract the EMG signal is shown below.

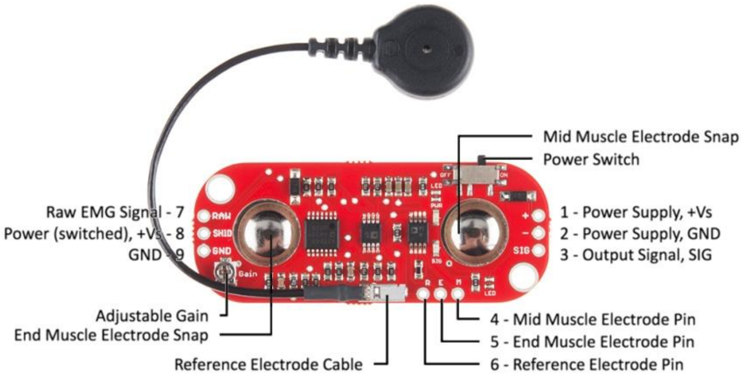

Fig 5: MyoWare Muscle Sensor

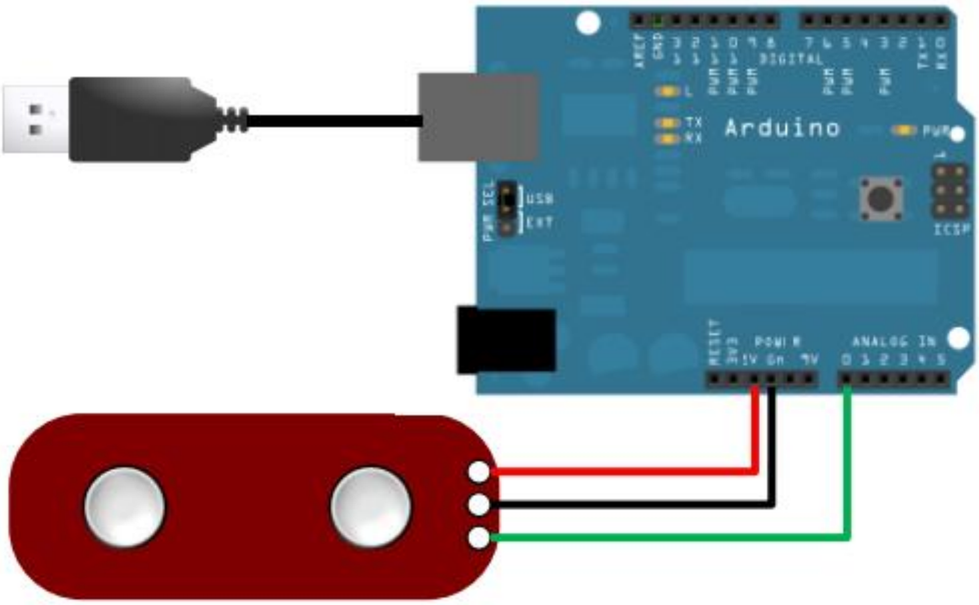

Fig 6: Circuit Diagram

Gelled electrodes use an electrolytic gel as a chemical interface between the skin and the metallic part of the electrode. Oxidative and reductive chemical reactions take place in the contact region of the metal surface and the gel. The metallic layer allows current from the muscle to pass more freely across the junction between the electrolyte and the electrode. This introduces less electrical noise into the measurement, as compared with equivalent metallic electrodes (e.g. Ag) [5]. Gelled electrodes can either be disposable or reusable. Disposable electrodes are the most common since they are very light. Disposable electrodes come in a wide assortment of shapes and sizes, and the materials comprising the patch and the form of the conductive gel varies between manufacturers. With proper application, disposable electrodes minimize the risk of electrode displacement even during rapid movements.

Position and orientation of the muscle sensor electrodes has a vast effect on the strength of the signal. The electrodes should be place in the middle of the muscle body and should be aligned with the orientation of the muscle fibres. Placing the sensor in other locations will reduce the strength and quality of the sensor's signal due to a reduction of the number of motor units measured and interference attributed to crosstalk [4].

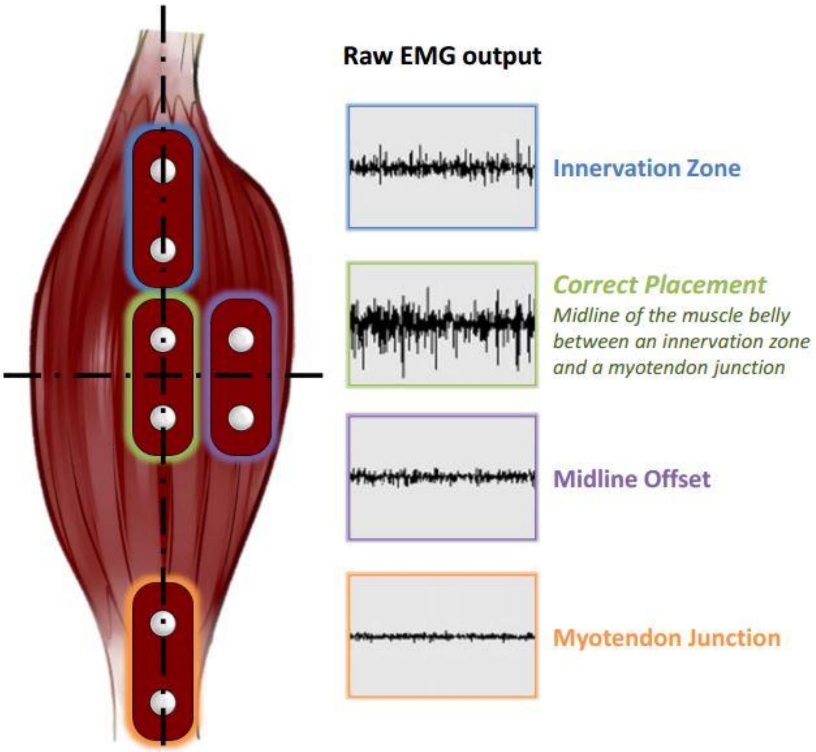

Fig 7: Electrode placement

Methodology

To start building a data set, it is necessary to have the building blocks of the same ready before extracting. In order to do this, 10 different hand signs are considered.

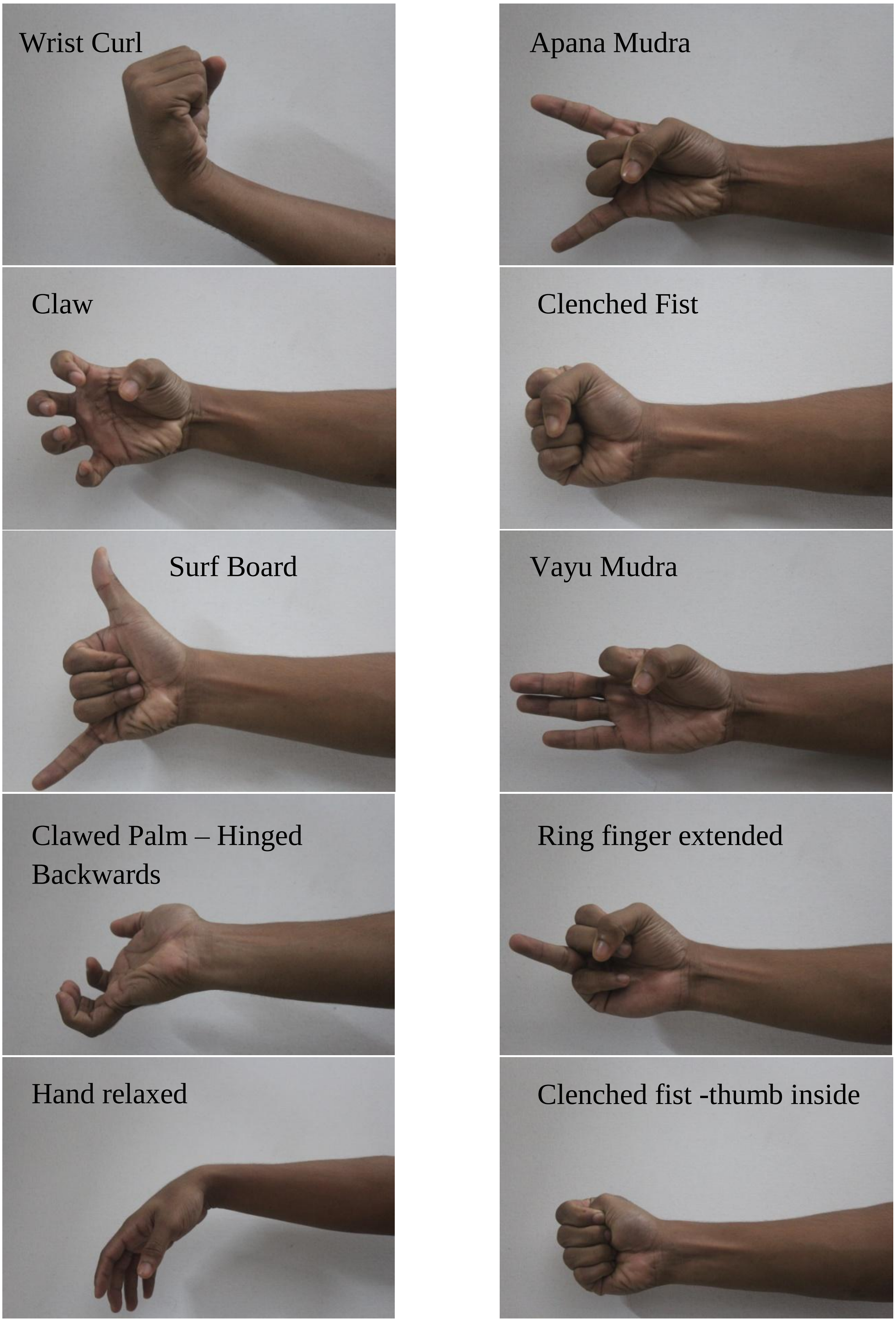

Fig 8: Hand signs

As the aim of this project is to enable deaf-mute individuals to communicate via sign language, the selected actions may use any muscle required.

The sensors were placed on 3 locations – one each on the dorsal and ventral side of the fore arm and the third, place on the bicep of the same arm. These locations were chosen in accordance with the hand signs selected, as maximum muscle activities are observed at these points. Further, the locations of the sensors were strictly kept towards the elbow due to 2 main reasons. First, the muscle bellies are located towards end of the fore arm. As a result, tapping of signals are much easier than the tapering end towards the wrist, where it is difficult to gauge and pin point the muscle whose activities might overlap with neighbouring ones. Second, each MyoWare sensor has a reference electrode which is ideally placed at a bony area. In order to give a common body ground, the reference electrodes are placed around the elbow joint.

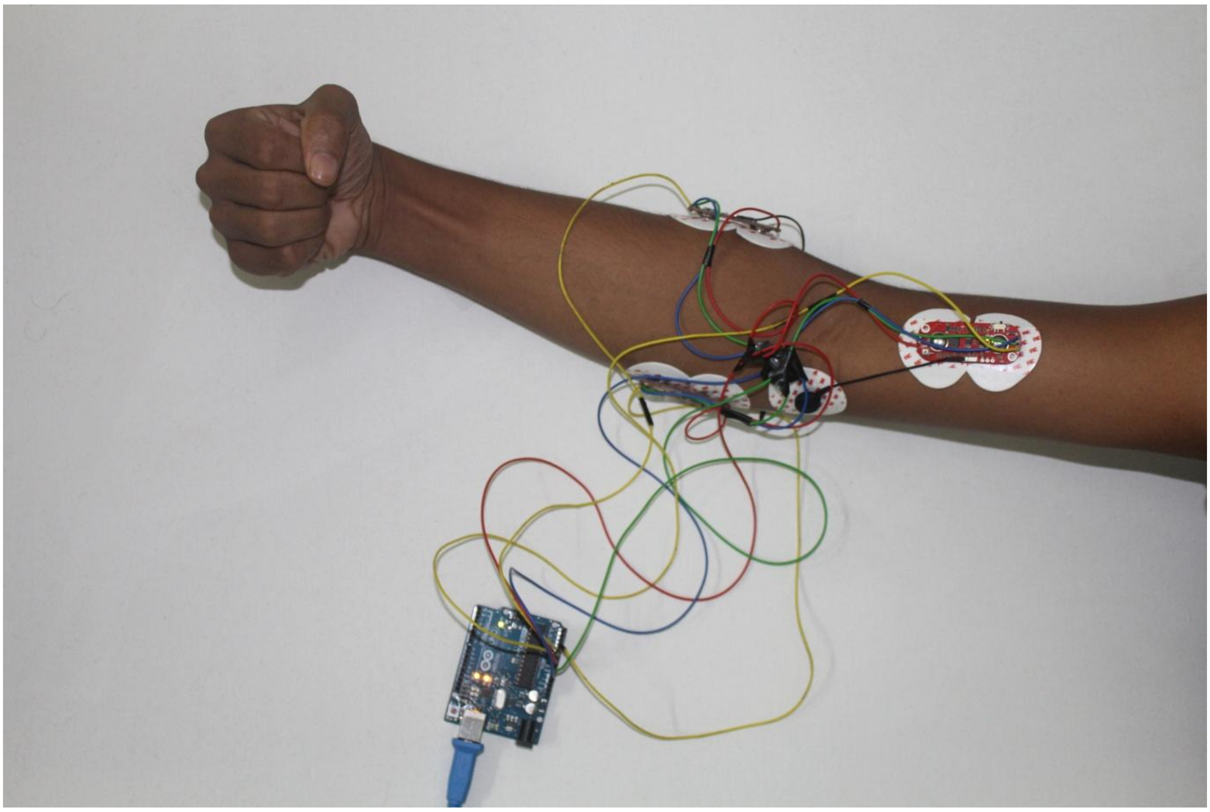

Fig 9: Final set up with all electrodes

Multi stranded wires were soldered to the MyoWare and the output from the same were fed to Arduino Uno's analog pins. Data extraction was done though CoolTerm, an open source software that publishes serial monitor values to a text file (.asc format). Each of the actions was performed in a trial for 65 seconds – alternating between rest and flexion for a period of 5 seconds. Signals were taken in the above mentioned way for 13 individuals – 8 males and 5 females, each of them having given a signed consent to participate in the research activity. The actions were taken in such a way that the transition between rest and flexion was minimal. Also, the first 15 seconds of the signal collection was taken as a rest period in order to avoid transients when the Arduino reads the signals through its Analog Input pins.

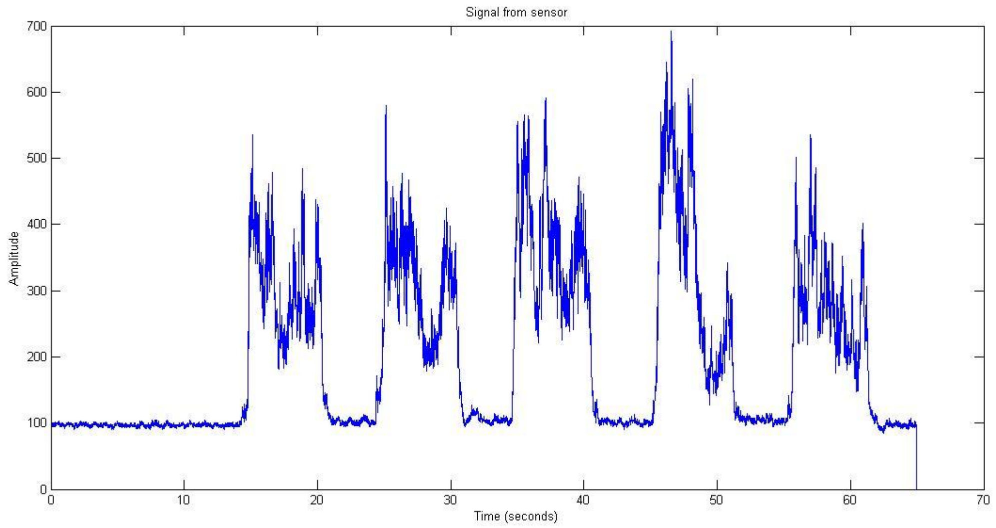

Fig 10: Rectified EMG Output for 5 action repetitions over 65 seconds**.**

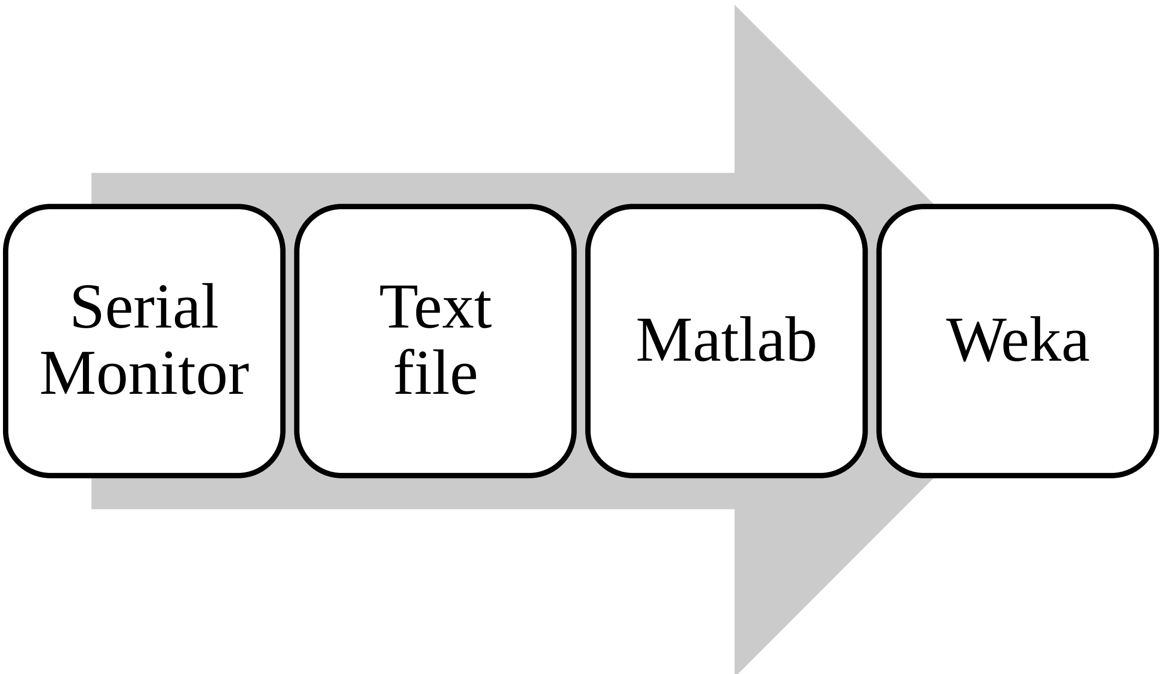

Fig 11: Flow of Signal for data collection and analysis

9 sounds were recorded and programmed to be triggered when the corresponding action was predicted.

Signal Conditioning

Having collected the signals from the test subjects, the data was concatenated into a matrix (subject vs. data set). In order to process this data suitable for classification, noise was removed as a first step. The entire data set was passed through a third order FIR filter and then a 30^{th} order Kaiser filter with cut-off frequency as 60Hz and 150Hz.

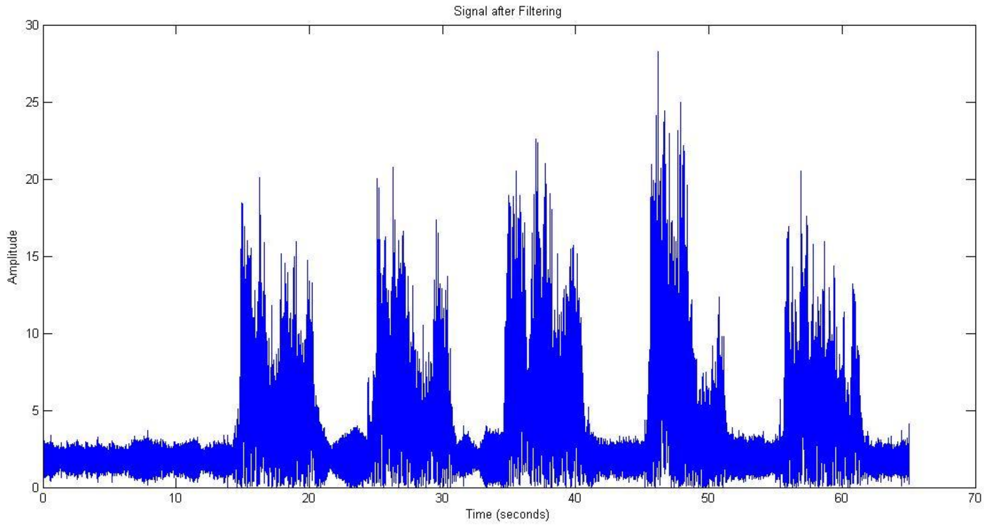

Fig 12: Signal post filtering

The next part of the algorithm was signal detection. The conditioned signal, having a total of 15 transients in all need to be split into individual transients, minimising the noise to zero. This gives a binary like input for the classification algorithm, high when signal is detected, low when it encounters a zero. In order to detect this signal, a custom algorithm was employed. Here, a window of desired size is considered which runs through the entire signal, split for each sensor (de-multiplexing the data set).

- Consider a window of desired length
- Window runs from start through end of signal
- Compute mean of each window, $\bar{X} = \frac{1}{n}\Sigma(x(n))$ and choose the sensor that has the maximum mean
- Compute energy of each window, $E = (x(n))^2$
- If $E_{win} > 15\,\bar{X}_{win}$, cut the signal at the start of the window, through the next 5 seconds

Run this process for all such transients. Store this new set in another matrix. The data set was ready for feature extraction.

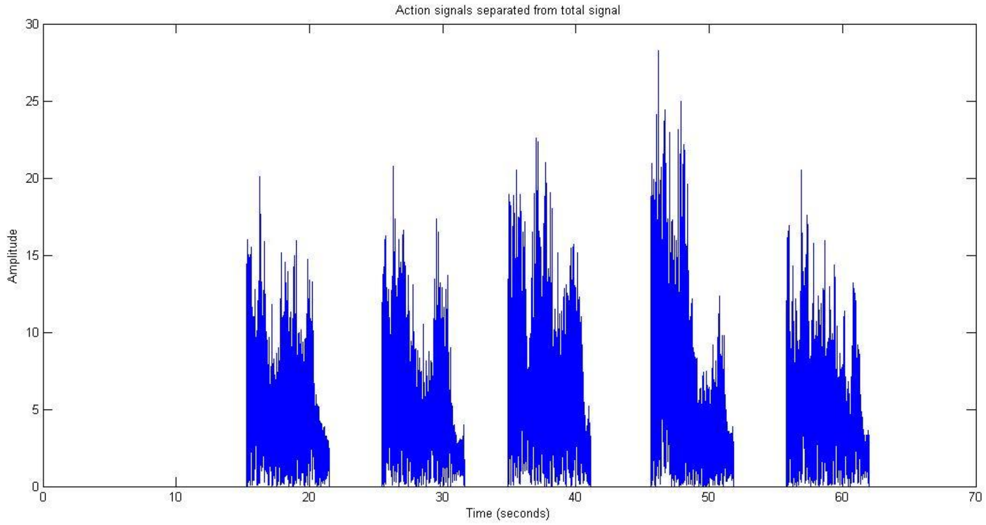

Fig 13: Signal after removing no-action noise

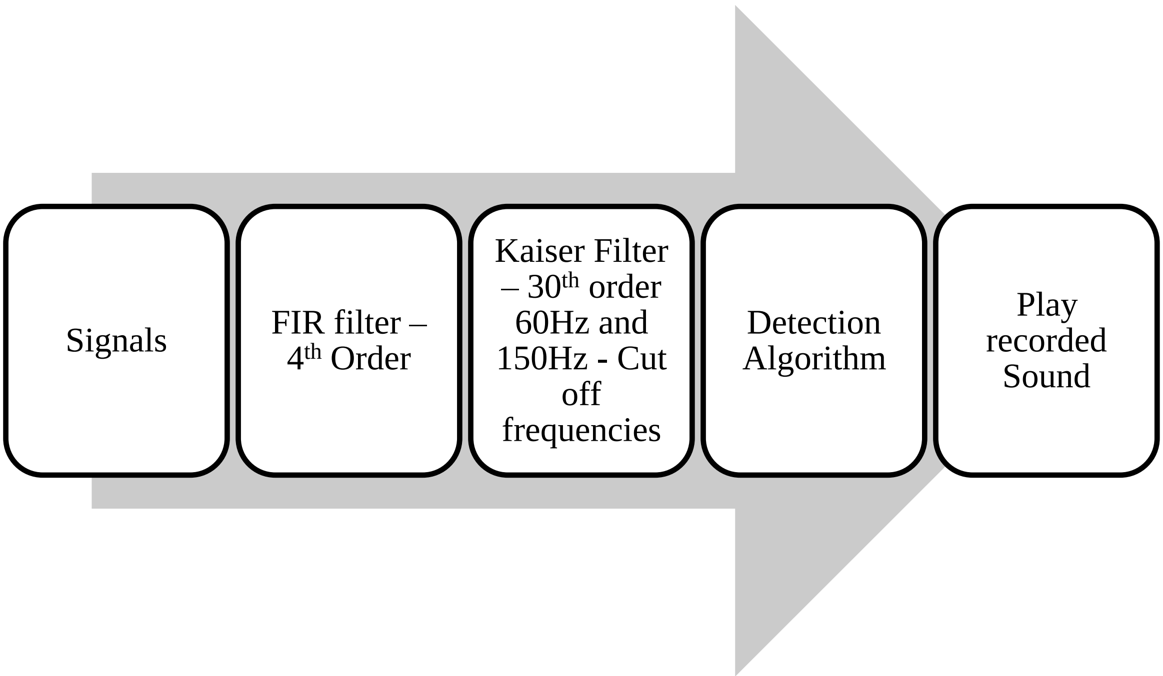

Fig 14: Process Flowchart

Feature Extraction

In machine learning and pattern recognition, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved.

In this project, time domain, frequency domain and several other parameters were chosen to extract information from the refined data set collected. Most of the features so chosen have two other parallel versions for each sensor.

Features in time domain have been widely used in medical and engineering practices and researches. Time domain features are used in signal classification due to its easy and quick implementation. Furthermore, it does not need any transformation for the features that are calculated based on raw EMG time series. Non-stationary property of EMG signal, changing in statistical properties over time has come to be a disadvantage for the features in time domain, but time domain features assume the data as a stationary signal. However, compared to frequency domain and time-frequency domain, time domain features have been widely used because of their performances of signal classification in low noise environments and their lower computational complexity.

They are as follows:

1. **Mean**: The means of each sensor data were computed.
2. **RMS**: The Root Mean Squares of each sensor were computed. RMS is defined as the square root of the mean over time of the square of the vertical distance of the graph from the rest state, related to the constant force and non-fatiguing contraction of the muscle.
3. **Variance**: Variance represents the extent of fluctuation of a signal from its mean. Variance is a property widely used especially when contractions of muscles are extremely strong and visible changes in signal patterns are observed. Variance uses power of a signal as a feature. Since variance is the mean value of the square of that variable, it can be computed as $\mathrm{Var} = \frac{1}{N-1}\sum_{n=1}^{N} \mathrm{x}_\mathrm{n}^2$ [7] .
4. **Power**: The power of a signal is the sum of the absolute squares of its time-domain samples divided by the signal length, or, equivalently, the square of its RMS level.
5. **Amplitude**: Maximum amplitude (MAX) is defined as the peak amplitude of a signal. It is often used in areas where the measured signal is not sinusoidal, where the signal swings above and below a zero value [11].
6. **Spectral roll-off**: The roll-off is a measure of spectral shape useful for distinguishing actual signal from noise that is a part of the signal. The frequency below which 85% of the magnitude distribution of the spectrum is concentrated is known as Roll-Off. That is, if K is the largest bin that fulfils, $\sum_{k=1}^{N/2} |\mathrm{X_r(k)}| \le 0.85 \sum_{k=1}^{N/2} |\mathrm{X_r(k)}|$[15].
7. **Spectral centroid**: Centroid is the gravity of the spectrum, where the sign function is defined by $\mathrm{C_r} = \frac{\sum_{k=1}^{N/2} \mathrm{f(k)}|\mathrm{X_r(k)}|}{\sum_{k=1}^{N/2} |\mathrm{X_r(k)}|}$, where N is a number of FFT points, X_r [k] is the STFT of frame x_r, and f [k] is a frequency at bin k. Centroid models the sound sharpness. Sharpness is related to the high frequency content of the spectrum [15].
8. **Mean frequency**: It is an average frequency which is calculated as the sum of product of the EMG power spectrum and the frequency divided by the total sum of the power spectrum. MNF= $\frac{\sum_{j=1}^{M} \mathrm{f_j P_j}}{\sum_{j=1}^{M} \mathrm{P_j}}$, where fj is the frequency value of EMG power

spectrum at the frequency bin j, Pj is the EMG power spectrum at the frequency bin j, and M is the length of frequency bin. In the analysis of EMG signal, M is usually defined as the next power of 2 from the length of EMG data in time-domain [10].

9. **Spectral Deformation**: The Ω ratio is sensitive to changes in spectral symmetry and provides an indication of spectral deformation. It is computed as: $\Omega = \frac{\sqrt{M_{2}/M_{0}}}{M_{1}/M_{0}}$, where M_n is the n^{th} spectral moment defined as: $M_n = \sum_{i=0}^{i_{max}} P_i f_i^n$, where P_i is the power spectral density value at frequency f_i [9].
10. **4^{th} order polynomial regression coefficients for normalised data set**: The polynomial $ax^3 + bx^2 + cx + d$ was used to fit the data set in the above curve. These four coefficients, a, b, c, d were used as features of each sensor.
11. **2^{nd} order exponential curve fit coefficients**: The polynomial $ae^{bx} + ce^{dx}$ was used to fit the data set in the above curve. These four coefficients, a, b, c, d were used as features of each sensor.
12. **1^{st} order Linear Predictive Coefficients**: This model is a powerful tool in representing the spectral envelop of a digital signal in a compressed form. The prediction of current sample as a linear combination of past *p* samples form the basis of linear prediction analysis where *p* is the order of prediction. The primary objective of LP analysis is to compute the LP coefficients which minimized the prediction error e (n). The popular method for computing the LP coefficients by least squares auto correlation method. This achieved by minimizing the total prediction error.
13. **Maximum to minimum drop in power density ratio**: An estimation of the power spectral density (PSD) of noise is a crucial part to retrieve signal in a noisy environment. The DPR is the ratio between the highest mean power density and lowest mean power density between 50 and 250 Hz. The mean power density is computed by averaged 13 consecutive points in the EMG power spectrum. The DPR is used to indicate whether the power spectrum was adequately peaked in the expected frequency range and can be used to detect the absence of EMG activity. It also ensures that the EMG spectrum drops off for higher frequencies, which would enable detection of high frequency noise and aliasing due to under sampling[6].
14. **Height of test subjects**: Height of each test subject was measured in centimetre.
15. **Weight of test subjects**: Weight of each test subject was measured in kilogram.
16. **Circumference of relaxed forearm of test subjects**: Measured in centimetre.
17. **Circumference of flexed bicep of test subjects**: Measured in centimetre.

Principal Component Analysis (PCA)

PCA seeks to find a linear transformation, $\bar{y} = W^T \bar{x}$, where $\bar{x} \in \Re^m$ and $\bar{y} \in \Re^n$ and $m > n$, such that the variance of the data is maximized in the projected space. Mathematically, PCA is a transformation that diagonalizes the covariance matrix of the global data set. It is also an unsupervised transformation which does not require labelled training data for finding the transformation. While m is the dimensionality of the original feature space, n is the desired dimension of the projected space and is usually determined as the number of significant Eigen values in the spectral decomposition of the global covariance matrix.

A general approach to the PCA is to first solve the characteristic polynomial equation for all Eigen values and then find their corresponding eigenvectors to produce principal components (PCs) in accordance with descending order of Eigen values.

For the given data set, the various n values selected were 30,40,45,50 and 61(feature matrix without principal component analysis). It was seen that for n=40, there was a higher coefficient of correlation.

Weka attribute selector

WEKA(Waikato Environment for knowledge analysis) is a popular suite of machine learning software written in Java. It is a collection of machine learning algorithms, for solving real-world data mining problems.

Advantages of Weka

- Free availability under the GNU General Public License.
- It is portable, as it runs on any platform and is fully implemented in Java.
- A comprehensive collection of data pre-processing and modelling techniques.
- Ease of use due to its graphical user interfaces.
- WEKA supports several standard data tasks, specifically, data pre-processing, classification, clustering, regression, feature selection.
- Run individual experiments.
- Builds KDD phases.

Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. In some cases, accuracy on future classification can be improved; in others, the result is a more compact, easily interpreted representation of the target concept. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively.

Weka provides an attribute selection tool. The process is separated into two parts:

Attribute Evaluator: Method by which attribute subsets are assessed.

Search Method: Method by which the space of possible subsets is searched.

The Attribute Evaluator is the method by which subsets of attributes are assessed. For example, they may be assessed by building a model and evaluating the accuracy of the model.

Some examples of attribute evaluation methods are:

1. **CfsSubsetEval**: Values subsets that correlate highly with the class value and low correlation with each other.
2. **ClassifierSubsetEval**: Assesses subsets using a predictive algorithm and another dataset that you specify.
3. **WrapperSubsetEval**: Assess subsets using a classifier that you specify and n-fold cross validation.

The Search Method is the structured way in which the search space of possible attribute subsets is navigated based on the subset evaluation. Baseline methods include Random Search and Exhaustive Search, although graph search algorithms are popular such as Best First Search.

Some examples of attribute evaluation methods are:

1. **Exhaustive**: Tests all combinations of attributes.
2. **BestFirst**: Uses a best-first search strategy to navigate attribute subsets.
3. **GreedyStepWise**: Uses a forward (additive) or backward (subtractive) step-wise strategy to navigate attribute subsets.
   In this project, the algorithm performs a greedy forward search through the space of attribute subsets. It starts with no attributes in the space and stops when the addition/deletion of any remaining attributes results in a decrease in evaluation.

Correlation based Feature Selection (CFS):

We addressed the problem of feature selection for machine learning through a correlation based approach. The central hypothesis is that good feature sets contain features that are highly correlated with the class, yet uncorrelated with each other. A feature evaluation formula, based on ideas from test theory, provides an operational definition of this hypothesis. CFS is an algorithm that couples this evaluation formula with an appropriate correlation measure and a heuristic search strategy. CFS searches feature subsets according to the degree of redundancy among the features. The evaluator aims to find the subsets of features that are individually highly correlated with the class but have low inter-correlation. The subset evaluators use a numeric measure, such as conditional entropy, to guide the search iteratively and add features that have the highest correlation with the class.

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

,

where M_S is the heuristic "merit" of a feature subset S containing k features, r_{cf} is the mean feature-class correlation (f ∈ S), and r_f f is the average feature-feature inter-correlation.

CFS evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Correlation coefficients are used to estimate correlation between subset of attributes and class, as well as inter-correlations between the features [18,19].

Classification

The selected features were then used for classifying the actions studied. A series of classification algorithms were used to train the data from first nine subjects. The algorithms that had coefficient of correlation, between actual class and predicted class, above 0.9 were considered. The following classifiers were studied.

Decision Tree:

A decision tree gets its name because it is shaped like a tree and can be used to make decisions. Technically, a tree is a set of nodes and branches and each branch descends from a node to another node. The nodes represent the attributes considered in the decision process and the branches represent the different attribute values. To reach a decision using the tree for a given case, we take the attribute values of the case and traverse the tree from the root node down to the leaf node that contains the decision [8]. The basic concerns in a decision tree classifier are the separation of groups at each non-terminal node and the choice of features that are most effective in separating the group of classes. In designing a decision tree classifier it is desirable to construct an optimum tree so as to achieve the highest possible classification accuracy with the minimum number of calculations. The binary tree classifier is considered a special case of a decision tree classifier.

Random Forest and Bagging:

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. It is defined as follows:

A random forest is a classifier consisting of a collection of tree structured classifiers $\{h(x, \Theta_k), k=1,...\}$ *where the* $\{\Theta_k\}$ *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.*[14]

Bagging (Bootstrap Aggregating) is an ensemble method that creates separate samples of the training dataset and creates a classifier for each sample. The results of these multiple classifiers are then combined (such as averaged or majority voting). The trick is that each sample of the training dataset is different, giving each classifier that is trained, a subtly different focus and perspective on the problem [14]. Bagging is similar to random forests. The fundamental difference is that in Random Forests, only a subset of features are selected at random out of the total and the best split feature from the subset is used to split each node in a tree, unlike in bagging where all features are considered for splitting a node.

Neural Network:

A neural network comprises of three layers:

- Input Layer – This is the part of ANN where input is fed to the network.

- Output Layer – This is the place where we obtain the output that is calculated by the network.
- Hidden Layers – These are the transition layers between the input and output layers where various calculation occurs which leads to the generation of output [20].

Transfer functions are used to convert the given input values to values ranging from zero to one. Each layer contains a specific number of values. Each value is called a node. Apart from these one of the important components of a neural network are weights. Weights are numbers which are multiplied with every node of a particular layer to form the next layer. The layers and the nodes both are represented in a form matrix. Matrix multiplication of a layer matrix to its assigned weight matrix leads to formation of next layer. Cost function is a mathematical equation relating the actual output to the calculated output. The cost function is directly proportional to the error. Therefore to minimize the error, cost function has to be minimized.

Results and Discussions

Once, the features were extracted, the entire feature set was fed to a k Nearest Neighbour (kNN) classifier. The correlation coefficient (0.63) was found to be unsatisfactory. To overcome this problem, the feature matrix was subject to Principal Component Analysis. The following were observed for kNN:-

- For dimension length of 30, correlation coefficient was 0.52
- For dimension length of 40, correlation coefficient was 0.71
- For dimension length of 45, correlation coefficient was 0. 61
- For dimension length of 50, correlation coefficient was 0.58

It is evident from above that for a dimension length of 40, the correlation coefficient was highest. In order to further improve, the efficiency of classification, feature selection was implemented. A correlation based feature selection algorithm was used. To find the set of features which gave the best results for the above mentioned evaluator, a greedy stepwise search was realized. This search resulted in ranking the first 10 valid features that would result in high classification efficiency. The selected features were:

- Sensor 3 – Polynomial Regression coefficients 1,2,3,4
- Sensor 1 – Exponential fit coefficients – 4
- Sensor 1 – 1^{st} order LPC
- Sensor 3 – 1^{st} order LPC
- Height of Subjects
- Circumference of forearm
- Circumference of bicep

This reduced feature matrix was classified using classification algorithms such as Random Forest, Bagging and Neural Network. k-Fold cross validation iteration with k =10 was employed for each of the classifier. The confusion matrices with the respective f-measures are listed below.

Random Forest - 10 features - cross validation											
	Predicted										
Actual		Action1	Action2	Action3	Action4	Action5	Action6	Action7	Action8	Action9	Action10
	Action1	74.47	25.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action2	8.33	69.44	22.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action3	0.00	8.33	88.89	2.78	0.00	0.00	0.00	0.00	0.00	0.00
	Action4	0.00	0.00	2.38	88.10	9.52	0.00	0.00	0.00	0.00	0.00
	Action5	0.00	0.00	0.00	0.00	92.59	7.41	0.00	0.00	0.00	0.00
	Action6	0.00	0.00	0.00	0.00	7.50	90.00	2.50	0.00	0.00	0.00
	Action7	0.00	0.00	0.00	0.00	0.00	6.12	87.76	6.12	0.00	0.00
	Action8	0.00	0.00	0.00	0.00	0.00	0.00	4.55	95.45	0.00	0.00
	Action9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	33.33	66.67	0.00
	Action10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.28	83.72
	F-Measures	**0.82**	**0.66**	**0.83**	**0.93**	**0.85**	**0.89**	**0.91**	**0.93**	**0.33**	**0.91**

Table 1: Confusion Matrix of Random Forest - 10 features - cross validation.

Bagging - 10 features - cross validation											
	Predicted										
Actual		Action1	Action2	Action3	Action4	Action5	Action6	Action7	Action8	Action9	Action10
	Action1	82.98	17.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action2	2.78	80.56	16.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action3	0.00	11.11	88.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action4	0.00	0.00	7.14	83.33	9.52	0.00	0.00	0.00	0.00	0.00
	Action5	0.00	0.00	0.00	14.81	74.07	11.11	0.00	0.00	0.00	0.00
	Action6	0.00	0.00	0.00	0.00	10.00	90.00	0.00	0.00	0.00	0.00
	Action7	0.00	0.00	0.00	0.00	0.00	12.24	77.55	10.20	0.00	0.00
	Action8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
	Action9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	33.33	0.00	66.67
	Action10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
	F-Measures	**0.90**	**0.75**	**0.83**	**0.86**	**0.73**	**0.85**	**0.87**	**0.94**	**0.00**	**0.98**

Table 2: Confusion Matrix of Bagging - 10 features - cross validation.

NN - 40 features - cross validation											
		Predicted									
		Action1	Action2	Action3	Action4	Action5	Action6	Action7	Action8	Action9	Action10
Actual	Action1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action2	8.51	61.70	14.89	10.64	0.00	4.26	0.00	0.00	0.00	0.00
	Action3	2.78	19.44	47.22	25.00	2.78	0.00	0.00	0.00	2.78	0.00
	Action4	0.00	0.00	14.29	65.71	14.29	2.86	2.86	0.00	0.00	0.00
	Action5	0.00	0.00	0.00	4.76	83.33	11.90	0.00	0.00	0.00	0.00
	Action6	0.00	0.00	0.00	0.00	18.52	66.67	14.81	0.00	0.00	0.00
	Action7	0.00	0.00	0.00	0.00	0.00	5.00	82.50	12.50	0.00	0.00
	Action8	0.00	0.00	0.00	0.00	2.04	0.00	16.33	65.31	16.33	0.00
	Action9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.67	71.43	11.90
	Action10	0.00	0.00	0.00	0.00	0.00	33.33	66.67	0.00	0.00	0.00
	F-Measures	**0.00**	**0.70**	**0.52**	**0.62**	**0.79**	**0.64**	**0.75**	**0.69**	**0.74**	**0.00**

Table 3: Confusion Matrix of NN - 40 features - cross validation.

From the above confusion matrices, it is observed that classification efficiency was highest for the bagging classification at 86.1%. Random forest method was a close second at 85.28%. Neural network, as expected, gave a mediocre classification efficiency of 67.6%. However, on close observation, we find that in the random forest – cross validation method, all actions are classified to a good extent, while in bagging – cross validation method, Action 9 was not classified. In order to make the classification more valid, a random forest – split method was employed. Here, data from the first 9 subjects were used to train the classifier and the data from the remaining 4 subjects were used to test the classifier. The confusion matrix of this classifier is shown below.

Random Forest 10 features 9-4 split											
		Predicted									
		Action1	Action2	Action3	Action4	Action5	Action6	Action7	Action8	Action9	Action10
Actual	Action1	14.29	85.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action2	0.00	60.00	40.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Action3	0.00	0.00	87.50	12.50	0.00	0.00	0.00	0.00	0.00	0.00
	Action4	0.00	0.00	0.00	58.33	41.67	0.00	0.00	0.00	0.00	0.00
	Action5	0.00	0.00	0.00	0.00	90.91	9.09	0.00	0.00	0.00	0.00
	Action6	0.00	0.00	0.00	0.00	0.00	92.31	7.69	0.00	0.00	0.00
	Action7	0.00	0.00	0.00	0.00	0.00	0.00	70.59	29.41	0.00	0.00
	Action8	0.00	0.00	0.00	0.00	9.09	90.91	0.00	0.00	0.00	0.00
	Action9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
	Action10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
	F-Measures	**0.25**	**0.43**	**0.74**	**0.70**	**0.74**	**0.67**	**0.80**	**0.00**	**1.00**	**1.00**

Table 4: Confusion Matrix of Random Forest 10 features 9-4 split.

This method, though does not have a high classification efficiency (63.63%), it is a proof that the features extracted from the signal can be used for effective classification of actions. It was also found that Sensor 2 (bicep) had no effect on the classification of the above studied actions.

Future Work and Scope for Improvement

There is a lot of scope for improvement in this project. A few notable ones have been listed below.

- As the actions selected for study here involve only open elbow, data from Sensor 2 was found to be inconsequential during classification. Using closed elbow in combination with the actions mentioned, more actions could be classified using Sensor 2.
- Most of the parameters used in the above classifiers such as maximum depth of trees, weight of vote from each tree in random forest, etc were not modified and experimented on Weka. Tweaking of these values may improve classification efficiency.
- Training data can be expanded to a larger size by involving more subjects. Algorithms like Neural Network and random forest show improvement on increasing the training set.
- The ADC used in this project was nearly 1 kHz, while the band of interest is between 60-150 Hz. A faster ADC could be used in order to sample the signal more effectively.
- As the sampling rate of ADC used was low, an anti-aliasing filter with a cut-off frequency of 400Hz should be used.
- Hardware issues can be handled better by building a casing for the set-up.

Conclusion

Random forests are an effective tool in prediction. Because of the Law of Large Numbers they do not over fit. Injecting the right kind of randomness makes them accurate classifiers and regressors. Furthermore, the framework in terms of strength of the individual predictors and their correlations gives insight into the ability of the random forest to predict. Using out-of-bag estimation makes concrete the otherwise theoretical values of strength and correlation. In this project, the aim was to create an action to speech convertor. Among the actions studied in this project, all actions are identified using the features mentioned with an acceptable average accuracy of 86.1%. The sensors used in this experiment cost no more than ₹3,000 and the ADC and Multiplexor of an Arduino UNO would cost about ₹1,500. The total cost of the marketable product would be at the most, ₹10,000. This minimalistic approach is simpler, lower cost and less bulky. It consumes less energy, and can be more intuitive for the user. The final outcome of the experiments can be seen as an illustrative step towards gaining useful knowledge that enables to decide which algorithm to use in certain situations.

References

1. Tanzim Kawnine, “A Radial-Ulnar Deviation and Wrist-Finger Flexion Analysis Based on Electromyography”, Mälardalen University, Sweden, 2008.

2. Jingpeng Wang, Liqiong Tang, John E Bronlund, “Surface EMG Signal Amplification and Filtering”, International Journal of Computer Applications (0975-8887) Volume 82- No1, November 2013.

3. Rami N Khushaba, Sarath Kodagoda, Maen Takruri, Gamini Dissanayake, “Towards Improved Control of Prosthetic Fingers using Surface Electromyogram (EMG) Signals, Elsevier, Expert Systems with Applications 39 (2012) 10731-10738.

4. https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf

5. Dr. Scott Ray, “Important Factors in Surface EMG Measurement”, Bortec Biomedical Ltd.
6. Xin Dang, Takayoshi Nakai and Md. Iqbal Aziz Kha , “Noise Power Spectral Density Estimation based on Maximum a Posteriori and Generalized Gamma Distribution”, International Journal of Advanced Science and Technology Vol. 54, May, 2013.
7. Angkoon Phinyomark, Chusak Limsakul, and Pornchai Phukpattaranont, “A Novel Feature Extraction for Robust EMG Pattern Recognition, JOURNAL OF COMPUTING, VOLUME 1, ISSUE 1, DECEMBER 2009, ISSN: 2151-9617
8. Bhanu Prakash Battula, KVSS Rama Krishna and Tai-hoon Kim,” An Efficient Approach for Knowledge Discovery in Decision Trees using Inter Quartile Range Transform”, International Journal of Control and Automation Vol. 8, No. 7 (2015), pp. 325-334.
9. Wan Mohd Bukhari Wan Daud, Abu Bakar Yahya, Chong Shin Horng, Mohamad Fani Sulaima, and Rubita Sudirman , “Features Extraction of Electromyography Signals in Time Domain on Biceps Brachii Muscle”, International Journal of Modeling and Optimization, Vol. 3, No. 6, December 2013.
10. Amjed S. Al-Fahoum and Ausilah A. Al-Fraihat,” Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains”, ISRN Neuroscience Volume 2014, Article ID 730218, 7 pages.
11. L.Ladha, T.Deepa,”FEATURE SELECTION METHODS AND ALGORITHMS”, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397 Vol. 3 No. 5 May 2011.
12. Mohini D Patil, Dr. Shirish S. Sane,” Effective Classification after Dimension Reduction: A Comparative Study”, International Journal of Scientific and Research Publications, Volume 4, Issue 7, July 2014 1 ISSN 2250-3153.
13. Angkoon Phinyomark, Chusak Limsakul, Pornchai Phukpattaranont,” A Novel Feature Extraction for Robust EMG Pattern Recognition”, Journal of Computing, Volume 1, Issue 1, pp 71-80, December 2009.
14. Leo Breiman, “RANDOM FORESTS”, pp. 2-3, 6-11.

15. Shruti Vaidya, Dr. Kamal Shah,” Audio Denoising, Recognitionand Retrieval”, IOSR IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. II (Mar-Apr. 2014), pp 107-112.
16. Sachin Sharma, Gaurav Kumar, Sandeep Kumar, Debasis Mohapatra, ” Techniques for Feature Extraction from EMG Signal”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 1, January 2012 ISSN: 2277 128X.
17. Mahmoud Tavakoli, Carlo Benussi, Joao Luis Lourenco, “Single channel surface EMG control of advanced prosthetic hands: A simple, low cost and efficient approach”, Expert Systems With Applications 79 (2017) 322–332.
18. M. A. Hall (1998). Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand.
19. D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.
20. Deepak P, Kshitij Shrivastava, Prathik K, Gautham Ganesh, Puneet S, Vijay Mishra, “Artificial Neural Network For Automated Gas Sensor Calibration”, Proceedings of 29th IRF International Conference, 24th July, 2016, Bengaluru, India, ISBN: 978-93-86083-69-2