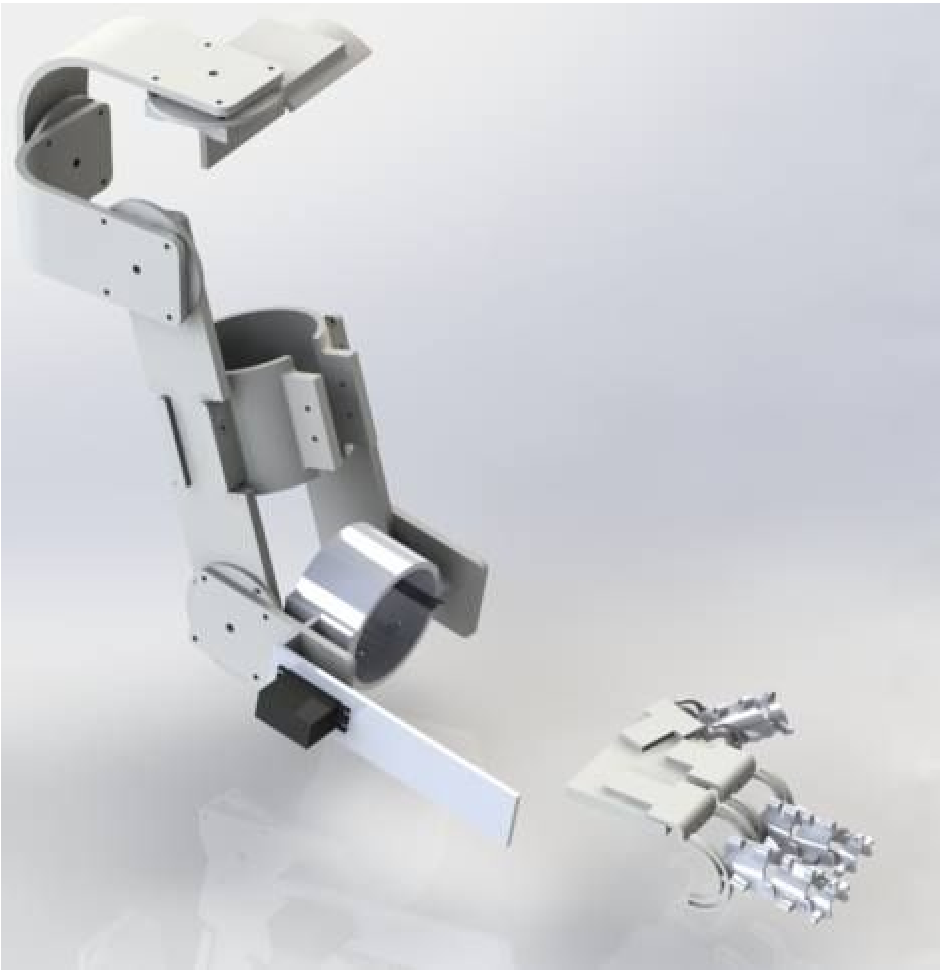
TEAM 6

ArMyo



1. Introduction

Paralysis is the loss of muscle function in some part of the body. It happens when something goes wrong with the way messages pass between your brain and muscles. Some common causes of limb paralysis are:

1. Stroke: This is the condition in which a person's brain is unable to communicate properly with a muscle resulting from loss of motor control of otherwise functional limbs. A stroke can be classified into two categories:
    a. Ischemia: due to lack of blood flow, and
    b. Haemorrhaging, due to internal bleeding
2. Spinal cord injury: It can cause a disruption of the communication channels between the brain and the peripheral nerve endings causing the signals to not go through completely.

There were about 5.35 million people living with paralysis in the United States alone which represents 1.7 % of the population. This problem is much more common in lower-income households (28% of households with a paralyzed member earn less than $15000 per year). The leading cause of paralysis is stroke (33.7%) followed by spinal cord injury (27.3%).

Impairments in reaching movements occur in about two-thirds of stroke survivors: upper limb functions are altered in the 73–88 % of first-time stroke survivors, and in the 55–75 % of chronic post-stroke patients. Indeed, in most of the cases post-stroke subjects remain:

- Unable to use their paretic limb to execute even basic actions
- Losing their independence in carrying out everyday activities.
- All those have a heavy and long-term financial burden imposed on both families and health care systems.

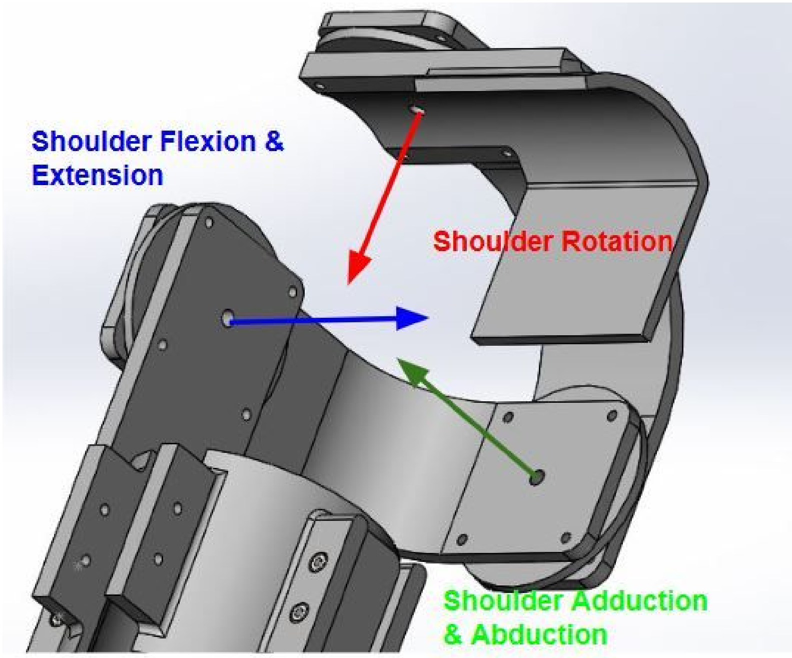
There exists, therefore an emerging need for intelligent and non-intrusive equipment to assist and support motion lost by stroke survivors by utilising the limited signals available by the nerve endings in the limbs. With the recent advances in robotics, mycobiology and artificial intelligence, such solutions are now within our grasp. Our product **ArMyo** aims to tackle the paralysis of the hands and arms by utilizing the weak signals produced in the paralyzed arm muscles to control the arm and the hand without the involvement of other parts of the body.
A robotic exoskeleton system is a noble man-machine intelligent system. It is an orthotic device with corresponding joints and links to the human joints and links. The muscle fibres release calcium ions when they are excited even when they are weak. These signals can be used to detect the intent for motion and then assist the weak muscles in performing the intended action. It is important to develop exoskeleton systems to assist and/or rehabilitate physically weak people in the present society in which a considerable percentage of the population is aged and physically weak.

Medical devices like our **ArMyo** help them hold and move which is highly desirable to improve their quality of life.

2. Mechanical Model

2.1 SHOULDER:

Mechanically, the objective at the shoulder was to provide a 3-DOF joint with sufficient workspace for the movement of the arm. Accordingly, we designed a kinematic model with 3-DOF for shoulder rotation, shoulder adduction & abduction and shoulder flexion & extension. Subsequently, we designed the links for a 3-Revolute DOF serial manipulator system. The orientation was done such that the axes of all the 3 Revolute joints was coincident at the point. It provided a joint close to ball-and-socket type joint, which is actually present in the shoulder. Each DOF is powered by a servo motor. For the servo motor, we chose a torque rating greater than 100 kg-cm, considering the weight of the subject's hand.



The orientation of the degrees of freedom, as can be seen in the figure, are in the order first shoulder rotation, followed by shoulder adduction & abduction, followed by shoulder flexion & extension. All the joints are designed such that they will provide passive freedom in movement when the servo motors are not attached.

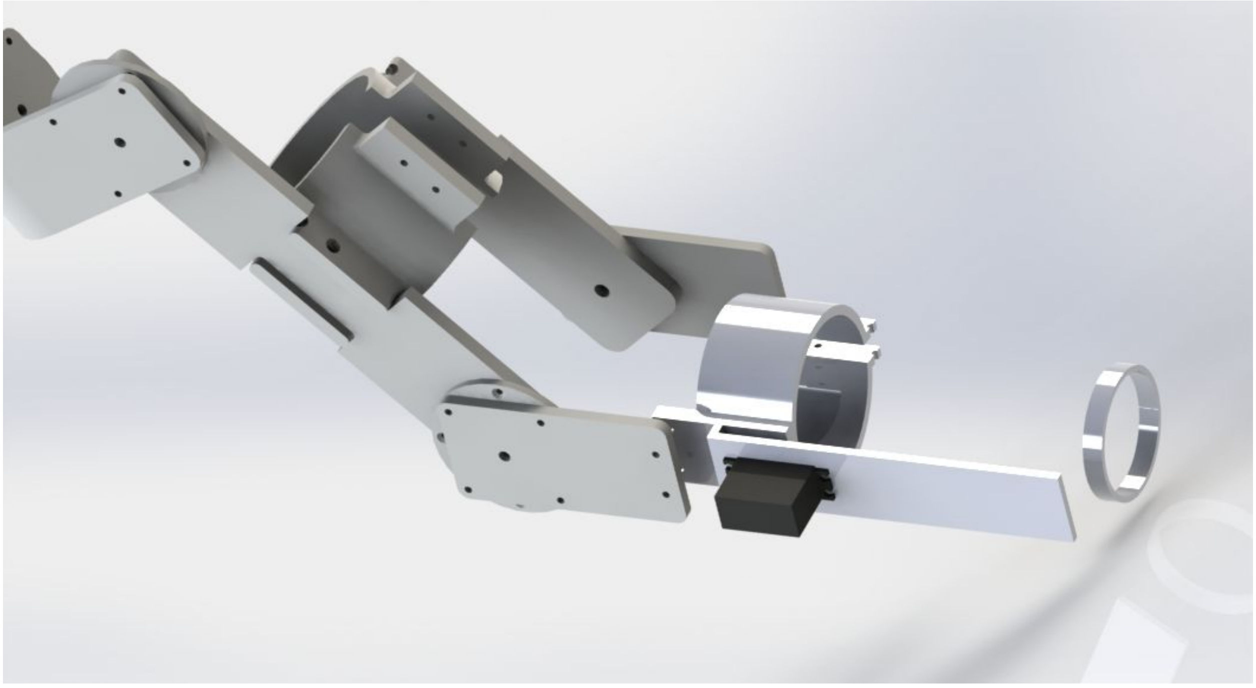
All the parts are manufactured using Aluminium to provide sufficient strength as well as possess lightweight.

2.2 Elbow Joint

The exoskeleton representing the elbow comprises of 2 co-axial revolute joints. The axis of the revolute joint has been made coincident with the axis of the hinge joint of the elbow. The revolute joint consists of 2 links and a disc in between with a hole of 6mm diameter. The pin is a standard steel clevis and cotter pin joint.
The pin joint connects the shoulder exoskeleton and the connecting link to the wrist joint. The material chosen for the connecting links is 6mm thick Aluminum 6061-T6 plates.
The shaft of a servo motor will be attached to the outer plate (connecting link to the wrist) while its body will be fixed to the upper link, thus imparting the relative rotary motion between the 2 links.

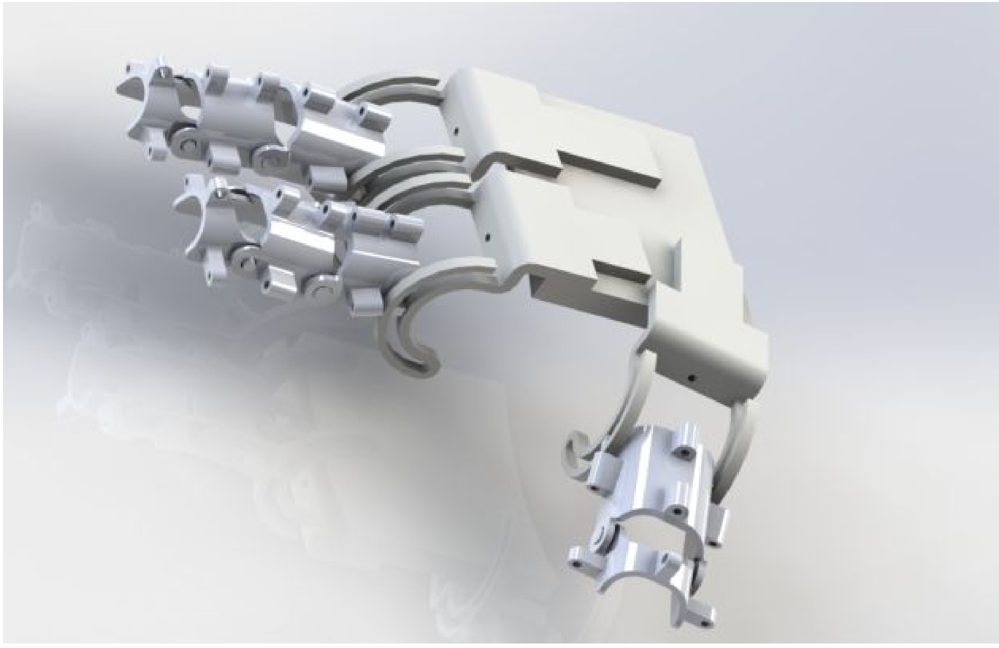
2.3 Wrist Joint

The wrist joint consists of two parts. One is fixed to the wrist and the other is fixed to the upper end of the forearm. Our objective is to create a actuate an angular motion between these two parts. A hose clamp is used to fasten the first part to the wrist. Two strings are wound over the clamp to actuate the twisting motion of the wrist. The clamp near the elbow has an extended section up to the wrist. The strings pass over pulleys attached to this extension. One of the strings is connected to a servo motor. When the string is pulled using the motor the string pulls the clamp which results in the rotation of the wrist. During this process, the other string which is wound in the opposite direction is pulled towards the clamp. And extension spring is attached to that string such that the energy is stored during this process. As soon as the torque from the servo motor is withdrawn the spring compresses to its natural length and the wrist twists back to its original position.



2.4 HAND:

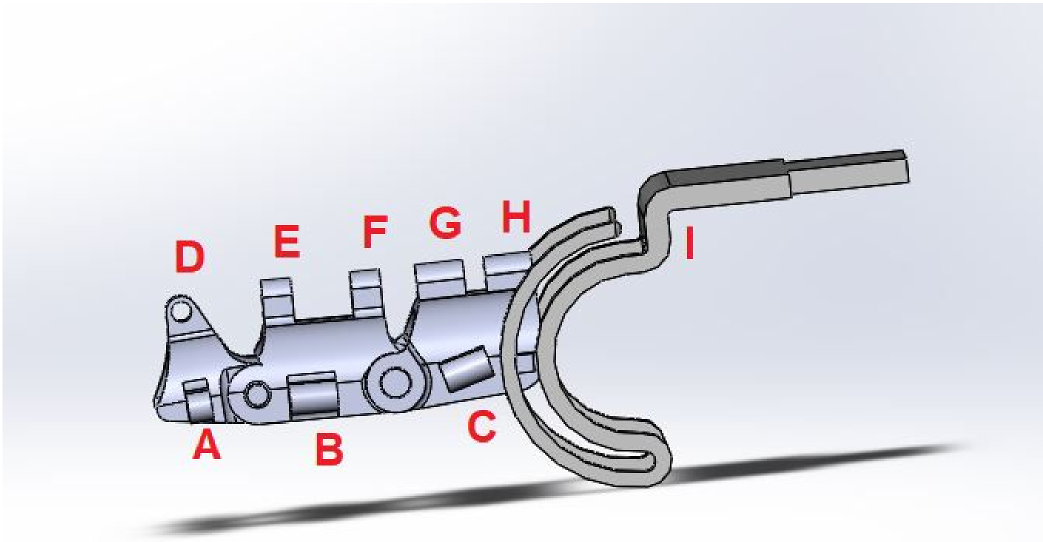
The objective of the hand for our exoskeleton is to grasp and hold regular shaped objects firmly. It was observed that the thumb, the forefinger and the middle finger of a hand was sufficient enough to provide force to hold many objects although not very heavy. Accordingly, kinematic links for the two fingers and one thumb for each hand was designed.



Hand Involving Thumb, Forefinger and Middle Finger

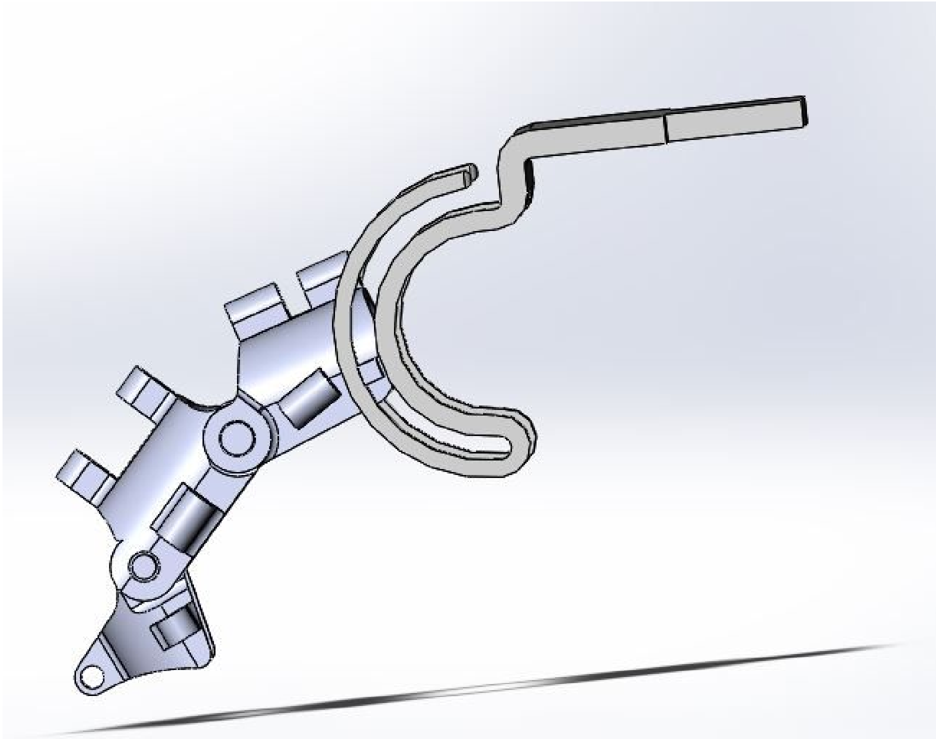
All the joints between the links of the fingers are simple revolute joints and the axis of these revolute joints are superimposed with the respective finger or thumb joints. However, for the last joint, it was not feasible to design a revolute joint and superimpose with the axis of the actual finger joint. Hence, we designed a C-shaped slot joint for it. These give the fingers complete freedom at every joint. The fore-finger and the middle finger is s 3-DOF system whereas the thumb is a 2-DOF system.

A mechanism using strings were used to close all the fingers while holding the objects. The mechanism is explained below.



Forefinger Unfolded

In the figure shown of the forefinger, a non-elastic string is attached at loop A. The string then passes through loop B and loop C and is finally attached to a servo. Similarly, another elastic string is attached at loop D and is guided through loop E, F, G and H, then finally attached at loop I. The servo pulls the string and the finger gets folded subsequently. At the same time, the string from D to I, being elastic, goes under tension. A Similar movement takes place at the middle finger and the thumb as well. This completes the action of folding the fingers of the hand to hold an object. When the servo releases the string AC, the elastic string DI under tension tries to get back all the links of the finger to the initial straight position. In this way, all the fingers are unfolded.



Forefinger folded

We have designed the A, B and C loops on both sides of each finger. When strings from both the sides are pulled, it will avoid any lateral movement of the fingers and the thumb.
All the parts are 3D printed using FDM technique. The material used for all the parts is PLA.

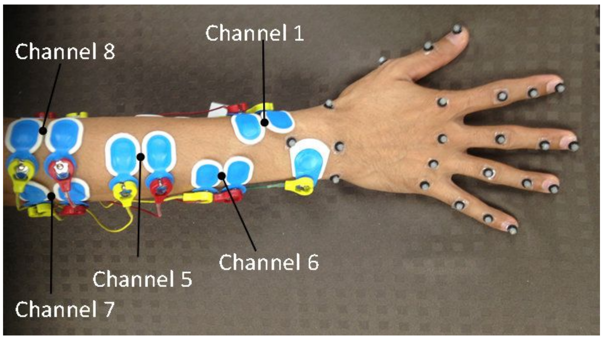
3. EMBEDDED SYSTEMS

The mechanical design of the Powered Exoskeleton incorporates various actuators and motor which needs to be controlled properly and precisely, also the device needed to have the capability of acquiring and classifying EMG signals. The EMG signal is a biomedical signal that measures electrical currents generated in muscles during its contraction representing neuromuscular activities. The nervous system always controls muscle activity (contraction/relaxation). Hence, the EMG signal is a complicated signal, which is controlled by the nervous system and is dependent on the anatomical and physiological properties of muscles.EMG detector, particularly if it is at the surface of the skin, collects signals from different motor units at a time which may generate interaction of different signals.

Electrical and electronic peripherals were added to help the Exoskeleton achieve the desired action helping the paralytic person to regain control of that lost entity. The complete Embedded System is divided into various parts described in detail as subsections below:

3.1 EMG Sensor

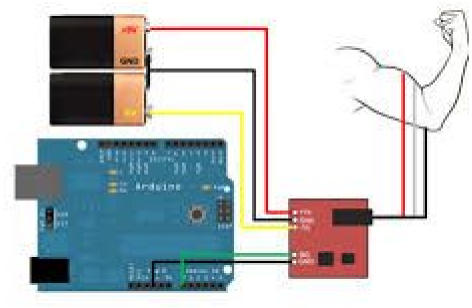
Measuring muscle activation via electric potential, referred to as electromyography (EMG), has traditionally been used for medical research and diagnosis of neuromuscular disorders. However, with the advent of ever shrinking yet more powerful microcontrollers and integrated circuits, EMG circuits and sensors have found their way into prosthetics, robotics and other control systems.
Gelled EMG electrodes contain a gelled electrolytic substance as an interface between skin and electrodes. Oxidation and reduction reactions take place at the metal electrode junction. Silver-silver chloride (Ag-AgCl) is the most common composite for the metallic part of gelled electrodes.
Surface EMG can be recorded by a pair of electrodes or by a more complex array of multiple electrodes. More than one electrode is needed because EMG recordings display the potential difference (voltage difference) between two separate electrodes.

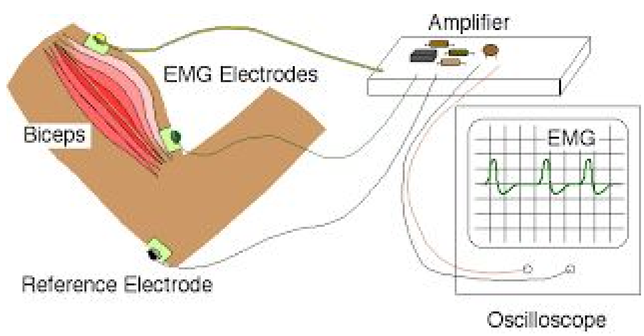


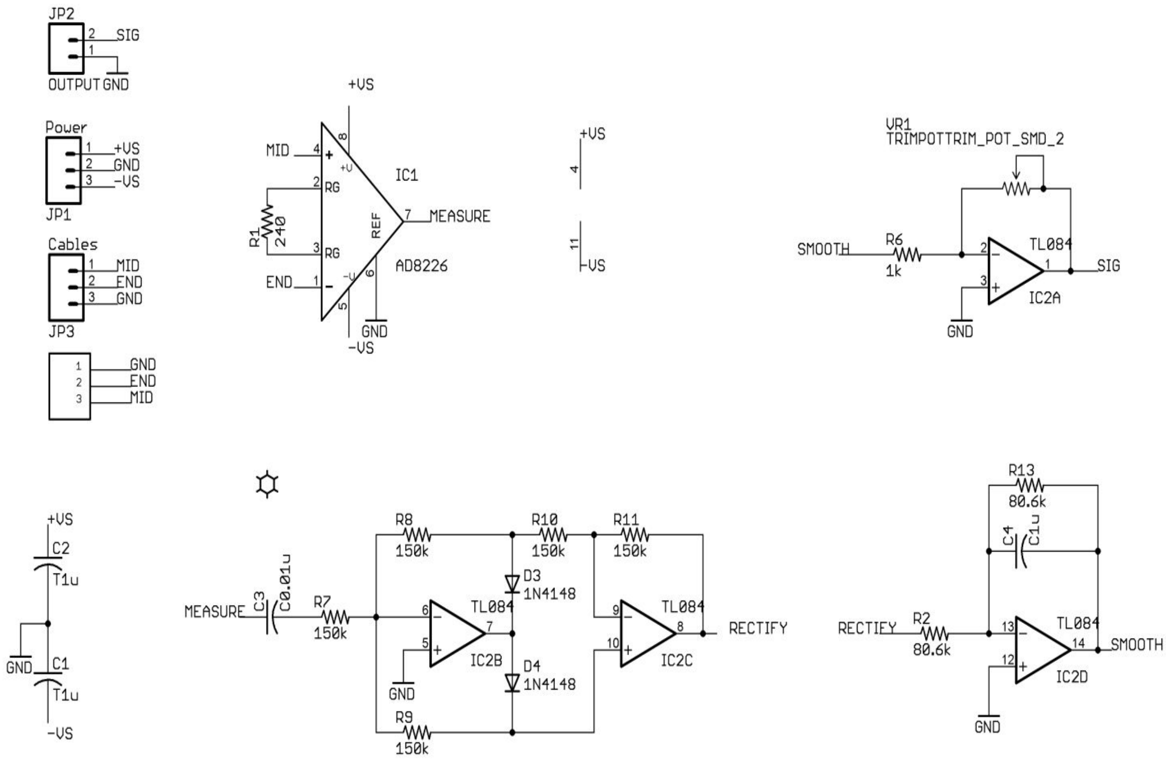
Available EMG Points on Hand :

Working: For each muscle action point, three probes are required for monitoring the signal. One of the electrodes is the reference while the other two electrodes are for carrying the signals.
The signals are in microvolts, which is amplified by the range 0-5V. The amplifier is based on the basic working of the instrumentation amplifier. This amplified signal is then further used as the

input signal to the Arduino Board. The amplifier also allows adjustable gain, so the user can customize as per the signal magnitude.







3.2 The Computing Unit

The model houses a Raspberry Pi 2 and an Arduino Mega 2560, which are responsible for efficiently operating the model.

3.2.1 RASPBERRY Pi 2

The Raspberry Pi 2 is a credit card sized single-board computer powered by ARM Cortex A7 processor. It has the following specifications:

- A 900MHz quad-core 32-bit ARM Cortex-A7 CPU
- 1GB RAM with 4 USB ports and 40 GPIO pins
- Full HDMI port
- Ethernet port
- Camera interface (CSI)

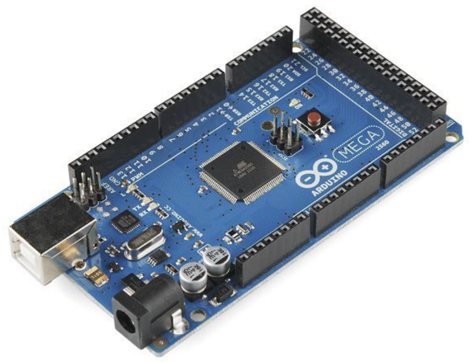
These capabilities of the Raspberry Pi 2 play a pivotal role in the working of our device. The Pi is responsible for running the deep convolutional neural network which extracts features from the acquired EMG signals. It also shares a strong communication link with the low-level controller - Arduino Mega to communicate the real-time information of the desired action to be performed by the arm which handles the controls of the motors.

3.2.2 ARDUINO Mega

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove but in a different package. There are totally 14 digital Pins and 8 analog pins 1 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a Mini-USB connection, an ICSP header, and a reset button.

It is used as a low-level controller in the system accomplishing the job of getting the position of different servo motors for a feedback-based control model and driving appropriate signals according to inputs received from the Raspberry Pi 3 to the motor drivers which actuate the motors.
It is mainly used to perform the following tasks :

- Sensor Data Collection
- Controls of the motors



EMG Data Collection and Communication with Pi:

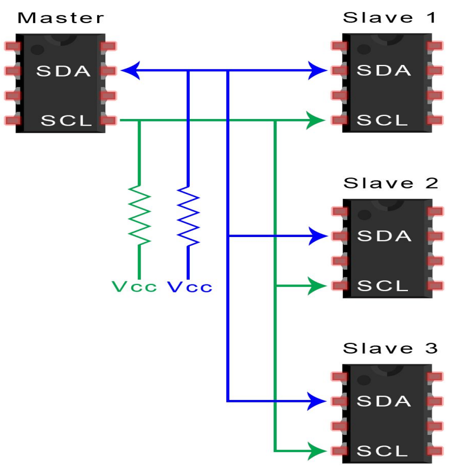
The amplified signals from the EMG electrodes are fed into the Arduino mega. Thus this the time-varying signal acts input feature to the Deep Neural Net Classifier in the Pi. This communication takes place through the Serial Communication Protocol.
Serial Communication is further used for transmitting the output of the classifier to the Arduino to perform the required action as per the signal.

Controls of the motor

Four RMCS-2201 High-Torque Encoder DC Servo Motors are controlled using I2C communication protocol with the help of Arduino Mega.
Four High-Torque MG996R Digital Servo motors are controlled using PWM of required Duty Cycle with the help of Arduino Mega.

I2C protocol

I2C protocol is used to control the high torque servo motors. I2c allows all motors to be connected in the same bus unlike protocols like UART ,accomplishing this by *addressing*.The master sends the address of the slave it wants to communicate with to every slave connected to it. Each slave then compares the address sent from the master to its own address. If the address matches, it sends a low voltage ACK bit back to the master. If the address doesn't match, the slave does nothing and the SDA line remains high.



3.3 Actuators

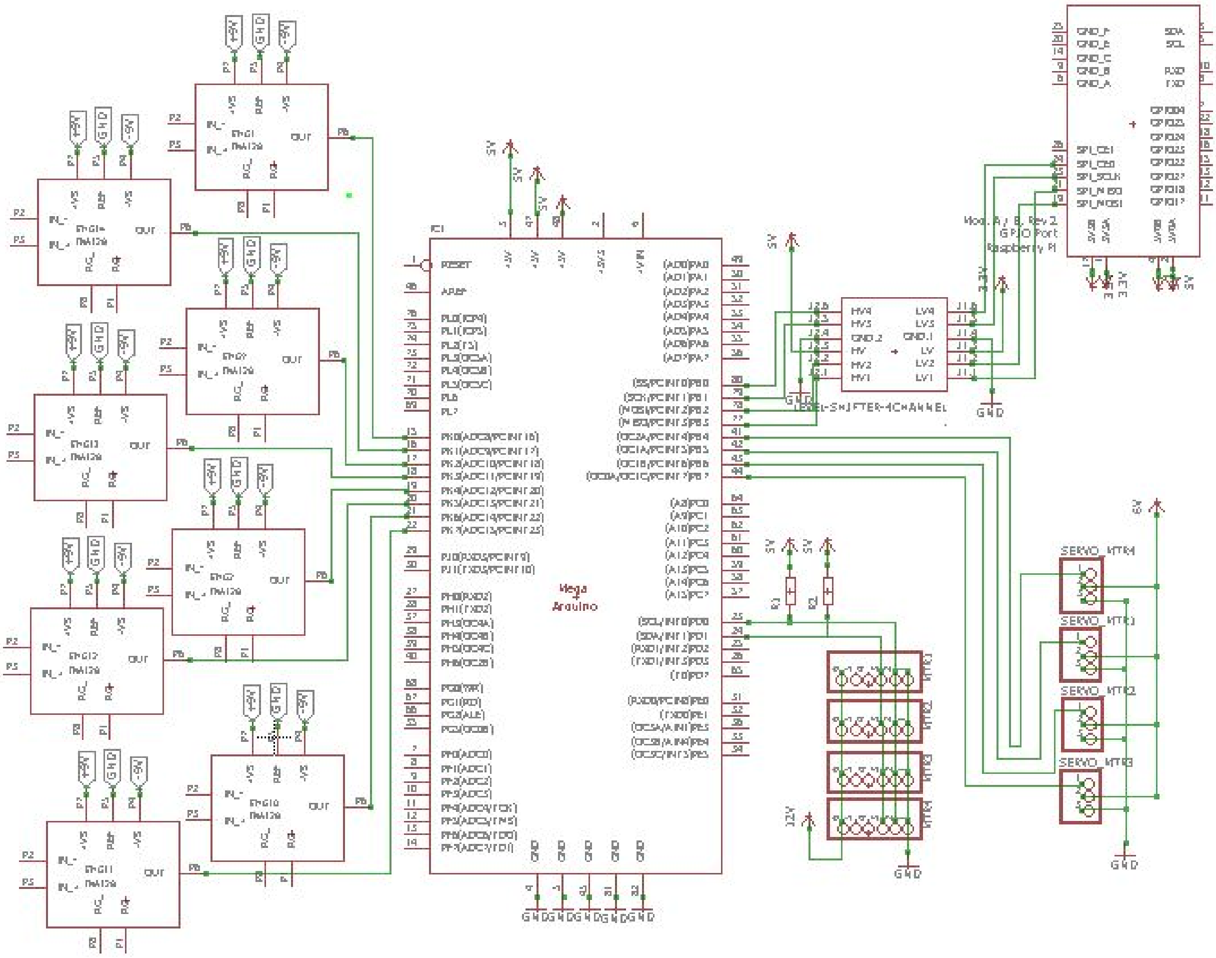
3.3.1 HIGH TORQUE SERVO MOTOR

This Encoder DC Servo motor solution integrates a 0.2 degree resolution optical encoder and a high power electronic servo drive on an Industrial grade high torque dc motor. It supports UART/I2C/PPM/Analog signals directly for absolute speed and absolute position control. This solution works extremely well for our mechanism which requires slow speed operation and it achieves this by providing high correction torque through a closed PI control loop.

3.3.2 LOW TORQUE SERVO MOTOR

Low Torque Standard Servo Motor with Metal Gears is used for the movement of fingers supported by the brackets which are 3D printed so that it enables the disabled with the ability to grab any object.

3.4 SCHEMATIC AND WORK-FLOW



Final Circuit

4. DEEP LEARNING MODEL

4.1 DATA ACQUISITION AND ITS PREPROCESSING

EMG data is acquired using a muscle sensor module using Arduino Mega. The sensor is non-intrusive in nature and reads the potential difference between two nearby points on a particular muscle on the human body. When a human being uses a certain body part, a set of muscles are either flexed or relaxed thus resulting in a potential difference building up along these muscles. Hence, we aim to capture these potential differences along a set of muscles in order to determine the action that a person desires to perform. The extent to which a muscle is involved while performing an action depends on the type of action and the intensity with which the action is performed. Hence, we aim to exploit this property in order to develop a human activity classifier which learns to predict the desired action from the time series data of the potential differences obtained from a set of muscles. An alternative to this method would be to use EEG signals from the brain to detect what action a human is willing to perform. However, since all actions involving arm and elbow movements are controlled by the neural cortex, the EEG signals corresponding to these movements would have a similar signature and hence, would be difficult to classify. Hence, we have considered using EMG data obtained from a set of muscles to achieve variety in the data, thus enabling us to develop a more robust classification paradigm capable of detecting small changes in the desired action.

The data, which is in a time-sampled form, is split into windows, each window containing a fixed number of samples. Then, each window is labelled according to the most prevalent action occurring within that window. After that, all the windows are split into three categories - training windows, validation windows and testing windows. As the names suggest, the training windows are used for training the model, the validation windows are used for validating the model in order to prevent the model from overfitting and the testing windows are used for testing whether the network works fine on the remaining part of the data, i.e., test data.

In order to make the data normalised for training, the data is normalized using Z-normalization, using the mean window and the standard deviation window of the training windows. If $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ be the mean window and standard deviation window of all the training windows respectively. Then, for a given window X, the normalised window X_n is :-

$$X_n = (X - \mu)/\sigma$$

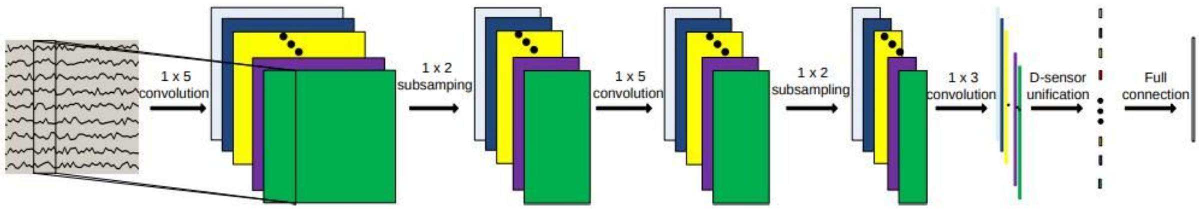
4.2 CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

Inspired from the paper on the use of Deep Convolutional Neural Networks for multichannel time series data, the network being used here is a Deep Convolutional Neural Network, with fully connected layers at the end.

For each arm, the data is obtained from four sensors placed at four distinct locations. The sensors are placed such that the signature of the EMG signal obtained from each sensor is minimally correlated with that obtained from other signals. Hence, a four-channel data is obtained from the sensors which are passed to a CNN based classification model in order to predict the desired action.

The input window of the network is such that, each row is the part of a time series belonging to an electrode, while the columns represent sampled data acquired from the EMG sensors. In order to learn the basic data representation firstly, the network's initial framework is such that the variation of data over time is learnt by using one-dimensional filters that convolve along the time axis. Each Convolutional filter is followed by a max pooling kernel, which aids in `spatial invariance', considering each window as an image. Also, in between, there are some Dropout layers which provide a form of regularization and help in reducing overfitting. After having learnt the vector representation of each of the electrodes, in order to learn the correlation between the electrodes, the individual electrode representations are concatenated. Finally, the last layer, which has as many neurons as there are classes, consists of a softmax[2] activation function and is fully connected to the previous layer. Each neuron's value in the final layer gives the probability of the input window belonging to that particular class. The flowchart of the network is depicted in the following figure :-

GIVE THE SAME FIGURE AS LAST TIME



Flowchart of the Network

$$\boxed{\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}}$$

(Here, σ represents the softmax function, xj being the neuron on which the activation function is being applied, and xi being any general neuron belonging to the layer on which the function is applied.)

The following table depicts the sizes and the number of trainable parameters of the network -

```
Net(
  (features): Sequential(
    (0): Conv2d(1, 50, kernel_size=(1, 5), stride=(1, 1))
    (1): BatchNorm2d(50, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(50, 40, kernel_size=(1, 5), stride=(1, 1))
    (5): BatchNorm2d(40, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

```
    (6): ReLU(inplace)
    (7): MaxPool2d(kernel_size=(1, 2), stride=(1, 2), padding=0, dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Linear(in_features=320, out_features=400, bias=True)
    (1): BatchNorm1d(400, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): Dropout(p=0.2)
    (4): Linear(in_features=400, out_features=8, bias=True)
    (5): Softmax()
  )
)
----------------------------------------------------------------------------------------------------------
Total params: 142,928
Trainable params: 142,928
Non-trainable params: 0
```

Thus, the network learns both the temporal aspect of the data, as well as, the extent to which the sensors under consideration are interdependent on each other, thus giving a very powerful learning model.

4.3 TRAINING PARAMETERS AND HYPERPARAMETERS USED

Training parameters, such as the number of training epochs, batch size etc. and Hyperparameters, such as learning rate, dropout etc. for a model are the most crucial parameters for the network to perform at its best potential. For the present model, the various training parameters and hyperparameters being used are as follows -

- Window Length = 30 samples
- Learning rate = 0.001
- Dropout = 0.2
- Number of epochs = 200
- Batch size = 1

4.4 RESULTS

Labelled data (divided into 9 classes - one of them corresponding to no movement, and other eight classes being the eight kinds of actions) of a certain duration is used for training, validation and testing. The data, consisting of 259 windows, is split between train, validation and test data in the ratio of 7:2:1 in a stratified fashion. The network trained, validated and tested using this data and the above-mentioned parameters and hyperparameters give a training accuracy of 100%, a validation accuracy of 95% and a test accuracy of about 90%.

Future Aspects :

Conductive Fabric Electrodes

Conductive fabric electrodes are a low cost and simple way to make reusable electrodes for sensing muscle activity. They can be sewn into any type of garment or used with a strap. When dampened, these electrodes allow you to sense the tiny electrical signals of your muscles just like traditional medical electrodes. For this tutorial, we're going to use them in a sleeve to detect the muscle activity of the forearm muscles.

Benefits of the conductive fabric electrodes over traditional EMG electrodes:

- Reusable - traditional EMG electrodes are meant to be used only once and then thrown out. Conductive fabric electrodes can be used over and over again simply by applying some water before use.

- No adhesive - traditional EMG electrodes use adhesives to stick to your skin. This adhesive can be somewhat of a pain to remove after use and can cause skin irritation to some people.

Multiplexing The Signals :

Using low noise 16:1 multiplexer, a single sensor will suffice to collect the EMG signal data from 16 different muscle points. Thus using 2x 16:1 multiplexer, the arms require only 1 amplifier/sensor and can switch between 16 channels. Thus there will be a significant cost reduction of the model.

From using dedicated sensors for each channel to using multiplexer, the reduction in cost will be around = ((Sensor Cost) x (16n-1) - 2n.(cost of multiplexer)) (for 16:1 channel analog multiplexer, where n is the number of such multiplexers are used). In our case, this value will be Rs. 29,000.
We experimented with the same during our testing, but due to noise in the hand fabricated circuit the results were not appreciable. This problem can be overcome by Industrial fabrication and using low noise multiplexer.

6. IMPACT

6.1 BACKGROUND

In 2010, 8.2 million people in Europe were affected by a stroke, with a total cost of about 64 billion euro per year. The number of patients paralysed due to stroke, spinal cord injury (SCI), post-polio, or other related diseases in orthopaedics is increasing. The induced paralysis puts these people at an increased risk of secondary complications, such as osteoporosis, muscle atrophy, diabetes, insulin resistance, and pressure ulcers.
Stroke is a serious condition associated with a high mortality rate and severe disability. Social functions, such as self-management and communication, are generally reduced in stroke patients, and many reports showed results of poor quality of life (QOL). Several studies have found that poor daily function decreases QOL and satisfaction levels in stroke patients; and it is noted that improving QOL was the primary objective of rehabilitation in stroke patients. Impairments in reaching movements occur in about two-thirds of stroke survivors:

Upper limb functions are altered in the 73–88 % of first-time stroke survivors, and in the 55–75 % of chronic post-stroke patients. Indeed, in most of the cases post-stroke subjects remain:

- Unable to use their paretic limb to execute even basic actions
- Losing their independence in carrying out the everyday activities. All those have a heavy and long-term financial burden imposed on both families and health care systems. Thus, medical devices like our **ArMyo** can help them hold and move which are highly desirable to improve their quality of life.

6.2 CURRENT ISSUES

The traditional mechanical Exo arms are abandoned by most of the paralyzed patients due to the unnatural and metabolically expensive movements required during the use of heavy actuators, such as lateral sway of the upper body, hip elevation in the swing phase. In addition, for some paralyzed patients without sufficient body strength, these arms cannot provide enough assistance and are not suitable. Any additional gear one puts on to compensate for the lack of movement, have to deal with these general problems:-

1. Intact Limb Pain
2. Back Pain
3. Poor Balance, Instability
4. General Fatigue and Reduced Mobility
5. Irritation and Skin Issues
6. Socket Issues or Discomfort

To tackle the above-mentioned problems which a normal person have to go through, we have certain features in our **ArMyo** which are explained below.

6.3 PROPOSED ADVANTAGES

Our **ArMyo** has multiple mounting points with which we attach the exo-arm to the user's body. This ensures that the overall load is not concentrated in a specific region which might be detrimental to the user in future. Additionally, just enough padding is provided on necessary regions of mounting so that the user does not have any comfort issues and irritation while using the **ArMyo**. We have used lightweight materials (like Aluminum, Wood, Acrylic and etc) so that the weight of **ArMyo** is kept minimum while still maintaining the minimum required structural rigidity. This is done to ensure that, the user can use our **ArMyo** for the longest possible time and does not have to deal with unnecessary fatigue and reduced mobility. Our LCD current similar products are very work specific, which means they cannot be used for multiple purposes which is because of less number of degree of freedom they provide to the user. On the other hand, our **ArMyo** has almost all the degrees of freedom which a normal human being's arm has. Thus it can be used for the multiple purposes giving the user more freedom.

Various uses for such a device which uses signals from a person's muscles to support and assist in the motion of limbs are possible:

1. This device is excellent for people suffering from muscle fatigue and is in therapy for regaining muscle strength. This device can artificially induce certain movement in muscles with little to no effort on the side of the user.
2. This device can also be used by people in the early stages of ALS by maintaining muscle movement even when their nerves undergo deterioration.

ANNEXURE

SPECIFICATIONS

- **Low Torque Servo Motor**
  - Operating Voltage: 4.8-7.2 Volts (Peak to Peak Square Wave)
  - Operating Temperature Range: -10 to +60 Degree C
  - Operating Speed (4.8V): 0.18sec/60 degrees at no load
  - Stall Torque (4.8V): 11 kg/cm
  - Dimensions: 1.6" x 0.8"x 1.4" (41 x 20 x 36mm)
  - Weight: 56gm
- **High Torque Servo Motor**
  - 10RPM 12V DC Servo motors with Metal Gearbox and Gears
  - 18000 RPM base motor
  - Diameter: Gearbox - 37mm , Motor- 28.5 mm
  - Length: Motor and Shaft 63mm + 15mm
  - Weight: 350gm
  - 120 kg-cm torque
  - No-load current = 800mA, Load current = up to 7.5 A(Max)
- **Encoder and Driver**
  - 0.2deg resolution optical encoder integrated on the motor output shaft
  - Absolute (32bit) Motor position control interface via UART, I2C, PPM signal or analog input
  - Speed and position can be controlled using a terminal or MCU via simple UART commands applied.
- **Battery**
  - Battery: Internal Lithium Polymer battery 480mAh
  - Battery life: up to 12 hours using proprietary wireless, up to 6 hours using Bluetooth® Smart

COST ANALYSIS

Component	Price
Materials cost	3000
Manufacturing cost	8000
EMG Sensors	8 x 2000
High Torque Servo Motors	4 x 2500
Low torque servo motor	4 x 300
Raspberry Pi 2	2500
Arduino Mega	600
LiPo Battery	1500
Total	**39800**

*for both the hands
**for single hand cost will be below 25000 INR

REFERENCES

1. [1] Bridle, J.S., 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Neurocomputing (pp. 227- 236). Springer, Berlin, Heidelberg.

2.https://www.christopherreeve.org/living-with-paralysis/stats-about-paralysis?fbclid=IwAR0pLd89pSmSJhU1RBEOzK7TOBaUQ0hJsKXT6TcqQanewlRLi3MMcjsN5tc

3. http://www.jsts.org/html/journal/journal_files/2018/12/Year2018Volume18_06_03.pdf