EMG CONTROLLED ARTIFICIAL HAND

NEU, 2015

EMG CONTROLLED ARTIFICIAL HAND

GRADUATION PROJECT SUBMITTED TO
THE BIOMEDICAL DEPARTMENT
OF
NEAR EAST UNIVERSITY

BY

JASSIM ALKAJEK
MHD TALAL ALBOUCHI
ABDALKARIM ALHOUSSENI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF SCIENCE IN BIOMEDICAL ENGINEERING

NICOSIA 2015

EMG CONTROLLED ARTIFICIAL HAND

GRADUATION PROJECT SUBMITTED TO
THE BIOMEDICAL DEPARTMENT
OF
NEAR EAST UNIVERSITY

BY

JASSIM ALKAJEK
MHD TALAL ALBOUCHI
ABDALKARIM ALHOUSSENI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF SCIENCE IN BIOMEDICAL ENGINEERING

NICOSIA 2015

We hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. We also declare that, as required by these rules and conduct, we have fully cited and referenced all material and results that are not original to this work.

Name, Last name:
JASSIM ALKAJEK - MHD TALAL ALBOUCHI - ABDALKARIM ALHOUSSENI

Date: 05/01/2015

ABSTRACT

This project aims to produce an artificial hand that fulfills the expectations of the amputees, that is, to perform and look as much as possible like a real human hand. In order to do that we used information carried on electromyogram signal and processed by ARDUINO environment. This signal operates a servo motor to manipulate the movement of the hand. The mechanical design of the hand needs to be approached from a bio mechatronics point of view that is, considering the integration of biological and medical issues and findings in a design that harmonizes the control and movement, the electric and electronic issues within the mechanical framework. Results show that a functional and trustworthy bionic hand can be made with cheap and available components.

Keywords: EMG, ARDUINO, artificial hand, grip hand, servo motor.

Dedication

I would like to dedicate this project to my beloved parents, grandparents and family for their encouragement and support. Also to my friends, I will always appreciate all they have done.

Last but not least, I hope that peace and love find their way back to my wounded countries, Syria and Palestine.

JASSIM ALKAJEK - MHD TALAL ALBOUCHI - ABDALKARIM ALHOUSSENI

ACKNOWLEDGEMENTS

First and foremost we would like to thank our supervisor Mr. Ali Işın who has shown plenty of encouragement, patience and support. He guided us through this project as graduate students.

We are also thankful for the contributions and comments of the teaching staff of the Department of Biomedical Engineering.

We are especially grateful to Assoc.Prof.Dr. Terin Adalı for being a constant source of encouragement. She helped us gain self-confidence. Here also we would like to thank our co-advisor Mr. Fatih Nurçin and friends at the Department of Biomedical Engineering who helped us one way or another.

This project was generously supported by the Department of Biomedical Engineering of the Near East University.

CONTENTS

LIST OF FIGURES

CHAPTER 1, INTRODUCTION

Much work has been done in the area of artificial hands. Previous theoretical work in the areas of kinematics, dynamics, grasping, sensing and actuation of artificial hands has been developed since the early 1980's; for one of the first studies of kinematics and force control issues for artificial hands, see (J. Kenneth Salisbury John J. Craig, 1982). The last five years have seen a big development in the practical implementation of these systems. There is a great amount of work done in identifying the motion of the human body and, in particular, of the human hand. The fields of interest are also diverse; much of the work has been done in the area of computer graphics, in order to create realistic virtual motion for avatar animation, for automatic hand language identification, for automatic sketching.

Recently, various prosthetic hands have been developed, but few are both attractive and functional. Considering human coexistence, prosthetic hands must be both safe and flexible. This project relates to the development of prosthetic myoelectric hand that performs many functions of real human hand like opening and closing of fingers. This movement of prosthetic hand is controlled by muscle contraction. Below-wrist amputee persons with missing limbs can append this prosthetic hand with the available stump and can do some of the hand operations with multiple degrees of freedom by voluntary activation of muscles using electromyogram (EMG) electrodes. The main design consideration includes degrees of freedom. Use of servo motor and microcontroller based on grip force generation based on EMG signals imparts a new function to the device. It will be useful for both robotic and prosthetic industry. This project aims to help amputees restore some of the capabilities of real hand. To rehabilitate such a person, training for facilities like opening, closing and grasping like natural hand must be done. This is done by using muscle signal which is converted to a control signal to drive a servo motor. This motor is used to control the movement of the prosthetic hand. Our project is split into three main categories which are signal acquisition (receiving and preprocessing) from biceps muscle, interpretation of the signal using microcontroller circuit (Arduino Duemilanove) and last but not least the hand design which mimic the mechanical aspects of the real hand.

In our project we are trying to answer certain questions such as:

- How to obtain the EMG signal?
- How to create a control signal from the EMG signal?
- How to mimic the real hand?

The answer to these questions helped us configure the design and approach suitable for our project.

In this project we tried to participate in adding features to the artificial hand by modifying hardware and software components of the hand.

In the hardware section we used multi joint fingers to imitate real finger movement. All joints cooperate with each other for grasping motion because they are connected together. According to the geometry of the object, motion of some joints is constrained by the contact between link and object and other free joints continue closing motion, then the shape of finger is changed for grasping resultantly. Also adding tendons and ligaments-like material to facilitate opening and closing and make more realistic.

In the software section we used an open-source electronics platform based on easy-to-use hardware and software (Arduino Duemilanove) which is a cheap and relatively simple to use microcontroller. The code was written using the Arduino development environment.

Chapter 2, PHYSIOLOGY

Electrical potentials exist across the membranes of virtually all cells of the body. In addition, some cells, such as nerve and muscle cells are capable of generating rapidly changing electrochemical impulses at their membranes, and these impulses are used to transmit signals along the nerve or muscle membranes. In other types of cells, such as glandular cells, macrophages, and ciliated cells, local changes in membrane potentials also activate many of the cells' functions. The project is only concerned with membrane potentials generated both at rest and during action by nerve and muscle cells.

2.1 Rest potential

For quiescent cells, the relatively-static membrane potential is known as the resting membrane potential. The resting membrane potential is at equilibrium since it relies on the constant expenditure of energy for its maintenance. It is dominated by the ionic species in the system that has the greatest conductance across the membrane [**Figure 1**]. For most cells, this is potassium. As potassium is also the ion with the most-negative equilibrium potential, usually the resting potential can be no more negative than the potassium equilibrium potential.

A neuron at rest is negatively charged because the inside of a cell is approximately 70 millivolts more negative than the outside (−70 mV); this number varies by neuron type and by species. This voltage is called the resting membrane potential and is caused by differences in the concentrations of ions inside and outside the cell. If the membrane were equally permeable to all ions, each type of ion would flow across the membrane and the system would reach equilibrium. Because ions cannot simply cross the membrane at will, there are different concentrations of several ions inside and outside the cell. The difference in the number of positively-charged potassium ions (K+) inside and outside the cell dominates the resting membrane potential. When the membrane is at rest, K+ ions accumulate inside the cell due to a net movement with the concentration gradient. The negative resting membrane potential is created and maintained by increasing the concentration of cations outside the cell (in the extracellular fluid) relative to inside the cell (in the cytoplasm). The negative charge within the cell is created by the cell membrane being more permeable to K+ movement than Na+ movement.

In neurons, potassium ions (K+) are maintained at high concentrations within the cell, while sodium ions (Na+) are maintained at high concentrations outside of the cell.

The cell possesses potassium and sodium leakage channels that allow the two cations to diffuse down their concentration gradient. However, the neurons have far more potassium leakage channels than sodium leakage channels. Therefore, potassium diffuses out of the cell at a much faster rate than sodium leaks in. More cations leave the cell than entering it causing the interior of the cell to be negatively charged relative to the outside of the cell. The actions of the sodium-potassium pump help to maintain the resting potential, once it is established. Recall that sodium-potassium pumps bring two K+ ions into the cell while removing three Na+ ions per ATP consumed. As more cations are expelled from the cell than are taken in, the inside of the cell remains negatively charged relative to the extracellular fluid (John E. Hall, 2010).

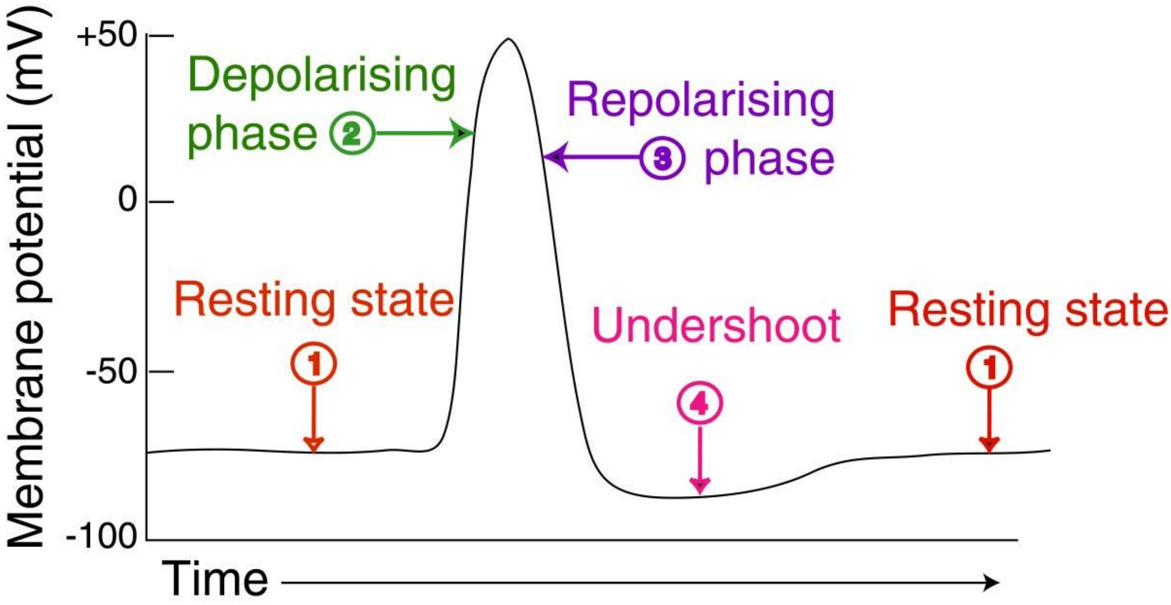

Figure 1 Rest potential

2.2 Action Potential

When the membrane potential of the axon hillock of a neuron reaches threshold, a rapid change in polarity occurs that moves along the axon in the form of an action potential.

This moving change in polarity has several stages:

- The depolarization, also called the rising phase, is caused when positively charged sodium ions (Na+) suddenly rush through open sodium channels into a neuron. The membrane potential of the stimulated cell undergoes localized change from-65 millivolts to 0 in a limited area. As additional sodium rushes in, the membrane potential actually reverses its polarity so that the outside of the membrane is negative relative to the inside. During this change of polarity the membrane actually develops a positive value for a moment (+40 millivolts). The change in voltage stimulates the opening of additional sodium channels, which are called voltage-gated ion channels.
- The repolarization, or falling phase, is caused by the closing of sodium ion channels and the opening of potassium ion channels releasing positively charged potassium ions (K+) from the neuron when potassium gates open. Again, these are opened in response to the positive voltage--they are voltage-gated. This expulsion acts to restore the localized negative membrane potential of the cell; a level of about -65 or -70 mV is typical for nerves.
- Many more potassium channels have been opened than are required and not all close when the membrane potential returns to normal, causing an undershoot or hyperpolarization. This will persist until the membrane permeability to potassium returns to normal [Figure 2].
- The refractory phase which can be divided into an absolute refractory period during which it is impossible to evoke another action potential, and then a relative refractory period, during which a stronger-than-usual stimulus is required. After the sodium channels close, they become inactive and cannot be opened again, regardless of the membrane potential (absolute refractory), until they transition to an active state. As more sodium channels return to active states the cell may depolarize, but a fraction of potassium channels remain open hyperpolarizing the cell, making it harder to depolarize to threshold. The absolute refractory period is responsible for the unidirectional propagation of action potentials.
- The action potential generated at the axon hillock propagates as a wave along the axon. The currents flowing inwards at a point on the axon during an action potential spread out along the axon, and depolarize the adjacent sections of its membrane. The absolute refractory period keeps the direction of propagation

unidirectional. In order to enable fast and efficient transduction of electrical signals in the nervous system, certain neuronal axons are covered with myelin sheaths. Myelin is a multi-lamellar membrane that wraps the axon in segments separated by intervals known as nodes of Ranvier. Myelin is produced by Schwann cells--specialized cells found exclusively in the peripheral nervous system--and by oligodendrocytes found exclusively in the central nervous system. Myelin prevents ions from entering or leaving the axon along myelinated segments. However, the current is carried by the cytoplasm, which is sufficient to depolarize the first or second subsequent node of Ranvier. Instead, the ionic current from an action potential at one node of Ranvier provokes another action potential at the next node; this apparent "hopping" of the action potential from node to node is known as saltatory conduction. The myelin sheath and nodes of Ranvier in combination help in reducing energy expenditure at the area of depolarization. Thus, the amount of sodium/potassium ions that need to be pumped to bring the concentration back to normal, or repolarize, is decreased. The conduction in myelinated fibers is hundreds of times faster since the action potentials only occur at the nodes of Ranvier. The myelinated fibers allow for transmission of signals quickly and efficiently (Khurana, 2009).

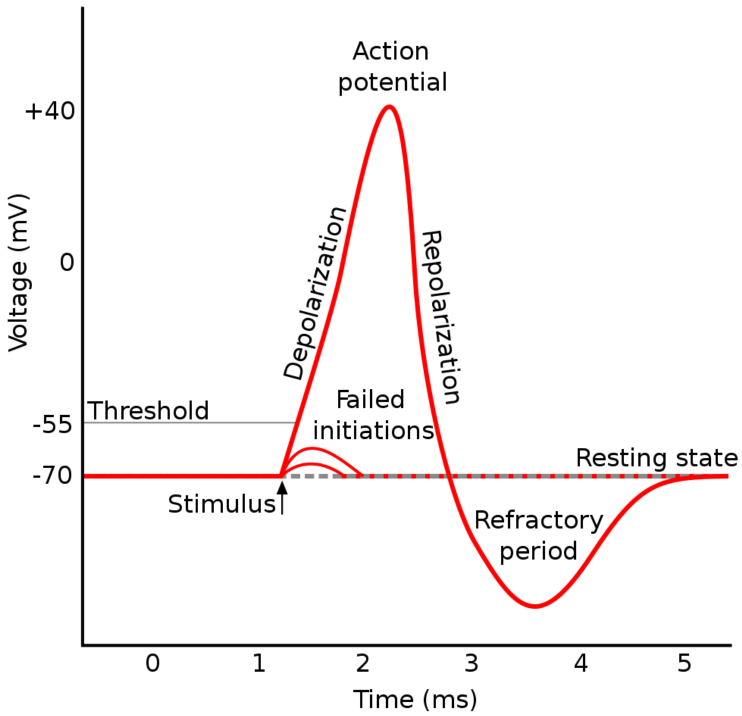

Figure 2 Action potential

CHAPTER 3, SIGNAL OBTAINING AND PREPROCESSING

Most medical instruments are electronic devices and so must have an electrical signal for an input. When a bio potential must be acquired, some form of electrode is used between the patient and the instrument. In other cases, a transducer is used to convert some nonelectrical physical parameter or stimulus, such as force, pressure, or temperature, to an analogous electrical signal proportional to the value of the original stimulus parameter.

3.1 EMG Signal obtaining

3.1.1 EMG Signal

The EMG signal is the electrical manifestation of the neuromuscular activation associated with a contracting muscle. It is an exceedingly complicated signal which is affected by the anatomical and physiological properties of muscles, the control scheme of the peripheral nervous system, as well as the characteristics of the instrumentation that is used to detect and observe it. Most of the relationships between the EMG signal and the properties of a contracting muscle which are presently employed have evolved serendipitously [Figure 3]. EMG signal recorded from skeletal muscles has amplitude ranging from 50 µV and up to 20 to 30 mV (Merletti, 2004).

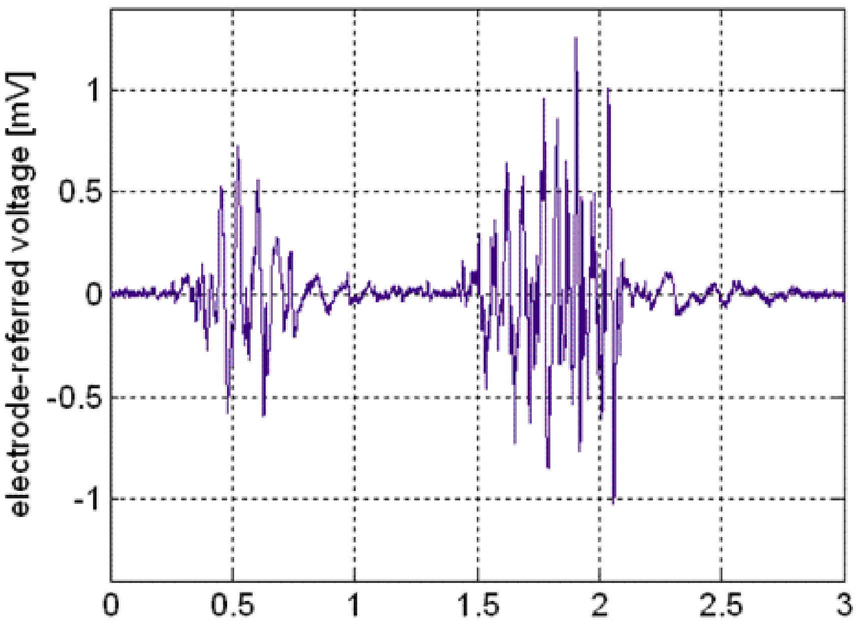

Figure 3 EMG signal

3.1.2 Surface Electrodes

Surface EMG electrodes are placed on the skin overlying a muscle. It is typical for surface EMG signals to be detected using a bipolar electrode configuration consisting of two electrodes with approximately 1cm spacing and a third electrode that works as reference. The reference electrode is attached to boney area such as elbow or wrist [Figure 5]. Using surface electrodes is appropriate for the purpose of the project because this method is easy to use and these electrodes are disposable.

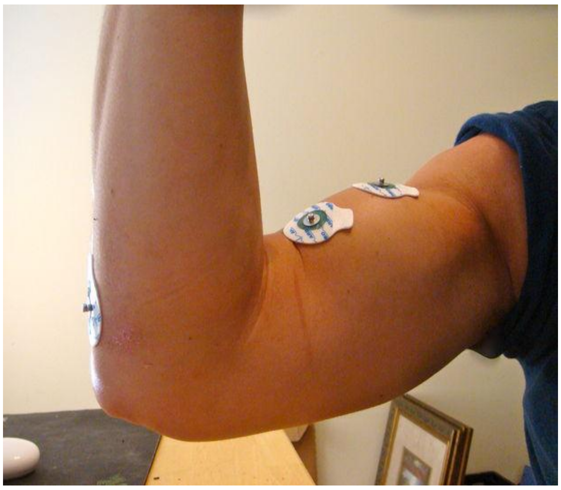

Figure 5 Electrode position

Figure 4 Surface electrodes

3.2 Preprocessing

The aim of preprocessing steps is to improve the general quality of the EMG for more accurate analysis and measurement. Noises may disturb the EMG to such an extent that measurements from the original signals are unreliable. The main categories of noise are: low frequency noise caused by body movements, high frequency random noises caused by mains interference (50 or 60Hz) and muscular activity and random shifts of the EMG signal amplitude caused by poor electrode contact and body movements. A number of linear and non-linear techniques have been developed to eliminate these artifacts. The preprocessing comprises of three steps: removal of low frequency noise, removal of high frequency noise and rectification (Carr, Joseph J. Brown, John M., 2001).

3.2.1 Differential Amplification

Differential amplifiers take two input signals and amplify the differences (good signal) while rejecting their common levels noise. It receives inputs from the two electrodes attached to the subject's skin. The two electrodes are connected to different parts of the bicep and will receive impulses of 13-15ms in duration and of voltages between 20-20000uv. The instrumentation amp has very high input impedance and doesn't require impedance matching which makes the design simple and efficient. The instrumentation amplifier is essentially a difference amplifier which means that it only amplifies the difference between the two electrodes attached to the bicep which should cancel out noise which would be equally affecting both inputs and therefore will not be amplified. This implies that the placement of the electrodes on the bicep must be far enough apart to have dissimilar signals in order to get a coherent output from the instrumentation amplifier. The output of the instrumentation amplifier will be a signal consisting of the signal we are interested in between 50-500Hz and noise which is spread over the entire spectrum of frequencies.

For the purpose of this project, INA118P differential amplifier circuit was used [Figure 6]. The INA118 is a low power, general purpose instrumentation amplifier offering excellent accuracy. Its versatile 3-op amp design and small size make it ideal for a wide range of applications. Current-feedback input circuitry provides wide bandwidth even at high gain (70 kHz at G = 100).

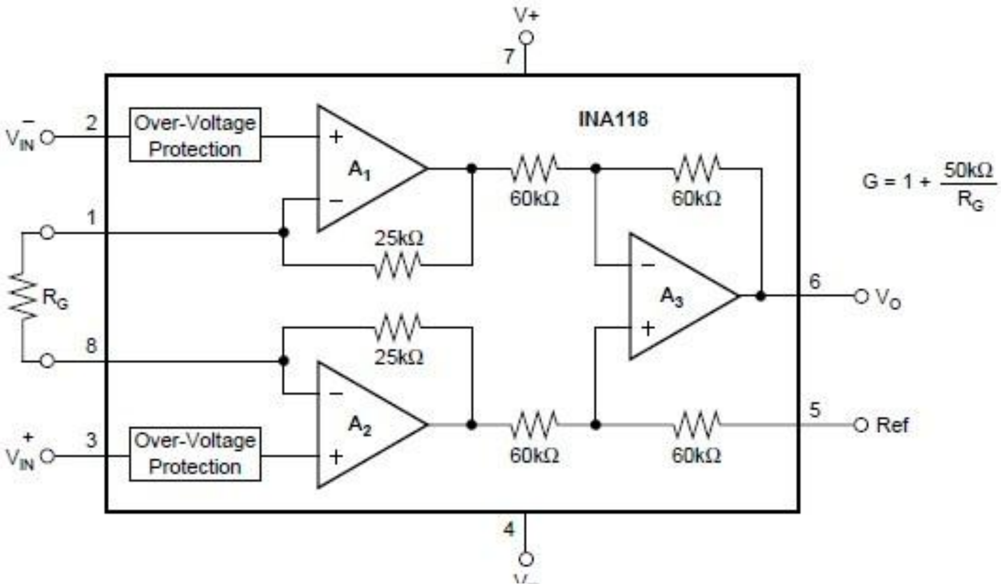

Figure 6 Differential amplifier

3.2.2 Band Pass filters

A band pass filter is composed of a low pass filter and a high pass filter with cutting frequencies of 50 and 500 Hz. These frequencies are chosen because most of valuable physiological information is carried on these frequencies.

A simple LF351N op-amp is used with different configuration to produce the low and high filter. The LF351 is JFET input operational amplifier with an internally compensated input offset voltage. The JFET input device provides wide bandwidth, low input bias currents and offset currents.

The low pass filter [Figure 7] is connected to the output of the instrumentation amplifier and is designed to remove frequencies that are above 500Hz. The low pass filter removes the frequencies above 500Hz because that is above the maximum signaling rate of the nerves in human muscles. Therefore any energy in frequencies above 500Hz is noise and will degrade the overall performance of the system if it is not removed.

The high pass filter [Figure 8] is connected to the output of the low pass filter and filters out frequencies that are below 50Hz. For the same reasons as with the low pass filter we filter out any frequencies that are below 50Hz because we know that they are noise. Putting a low pass filter is series with the high pass filter effectively creates a band pass filter that only allows frequencies between 50-500Hz to pass through without attenuation; which is the range of frequencies human nerves can transmit signals.

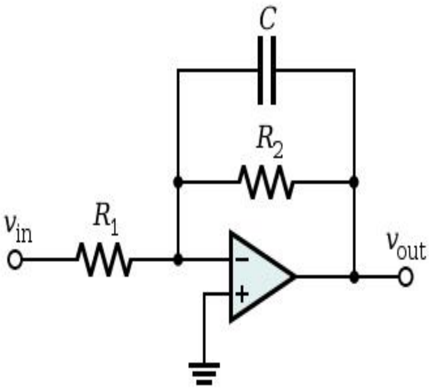

Figure 7 Low Pass Filter

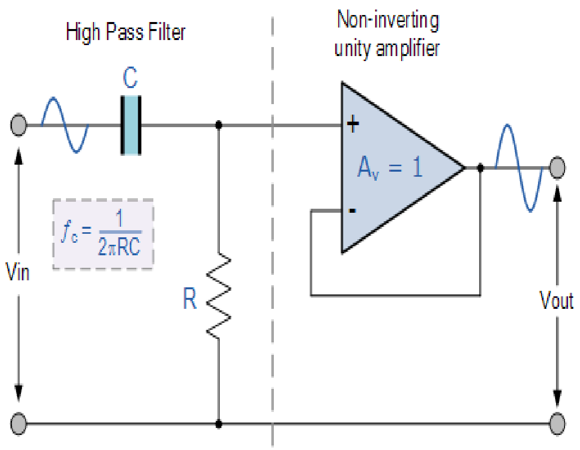

Figure 8 High Pass Filter

3.2.3 Rectification

The precision rectifier is a half wave rectifier that is configured so that the op-amp never goes into saturation due to the diode in parallel with the resister. The output of the filter must be rectified so that it can be an input to a comparator. The output of the precision rectifier is a signal that is always positive and consists of a series of impulses when the bicep is flexed and relatively close to zero when the muscle is relaxed.

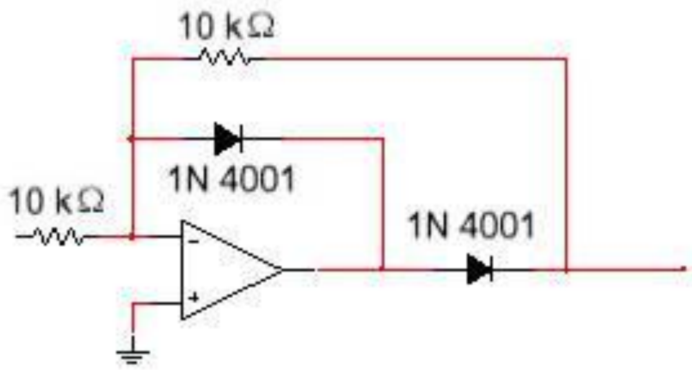

Figure 9 Rectifier

CHAPTER 4, EMG SIGNALS INTERPRETATION

4.1 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. We used ARDUINO as microcontroller that receives the output of the rectifier stage and interprets the impulses in order to convert them into modulated width pulse signal to control the servomotor.

Starting clockwise from the top center:

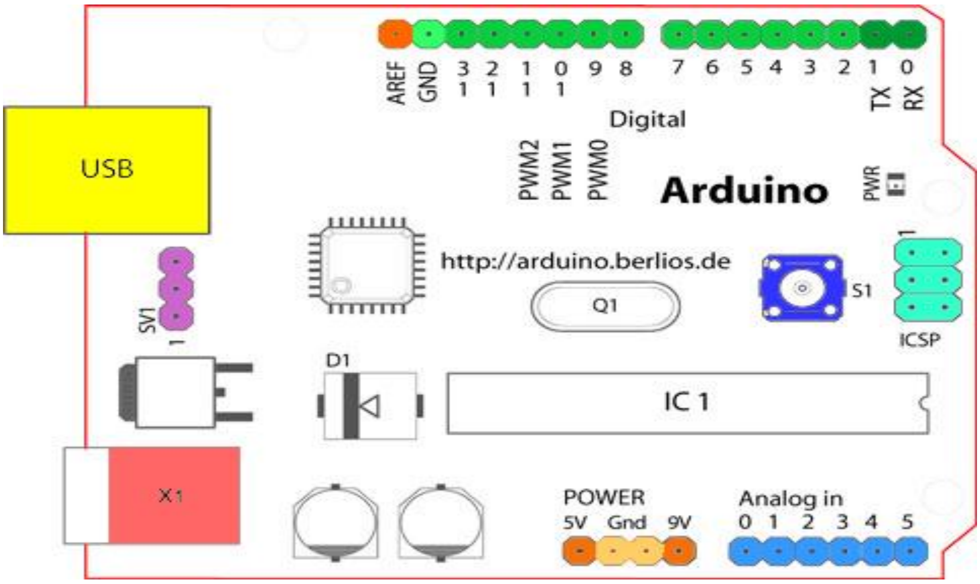

Figure 10 Arduino schematic **Ref, (ARDUINO)**

Analog Reference pin (orange).

Digital Ground pin (light green).

Digital Pins 2-13 (green).

Reset Button - S1 (dark blue).

Analog in Pins 0-5 (light blue).

Power and Ground Pins (power: orange, grounds: light orange).

External Power Supply In (9-12VDC) - X1 (pink).

USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow).

What we used for our project:

1. Arduino Duemilanove Board.
2. USB programming cable.
3. 9V battery or external power supply (for stand-alone operation). The board is powered by a battery rather than through the USB connection to the computer.
4. Breadboard for external circuits, and solid wire for connections.
5. PC running the Arduino development environment. One of the important features of Arduino environment is that you can easily create any code to achieve your own purpose, just download it from Arduino website and it will run automatically. After uploading the board it can be disconnected from the PC, and the program will still run from the top each time you push the reset button.

4.1.1 Board (Duemilanove)

The Arduino Duemilanove [Figure 12] is a microcontroller board based on ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, USB connection, a power jack, and a reset button. And the power source is supplied by connecting it to a computer with a USB cable or powers it with an AC-to-DC adapter or battery. We chose this type of Arduino board [Figure 11] because it is cheap, easy to use and achieve project's purpose (McRoberts, 2013).

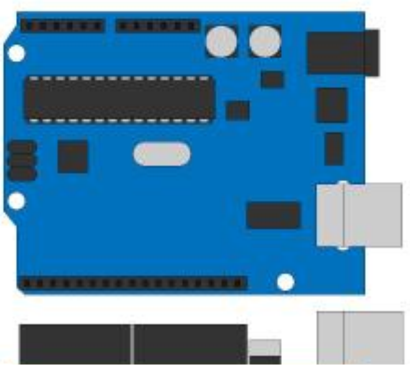

Figure 11 Arduino Board

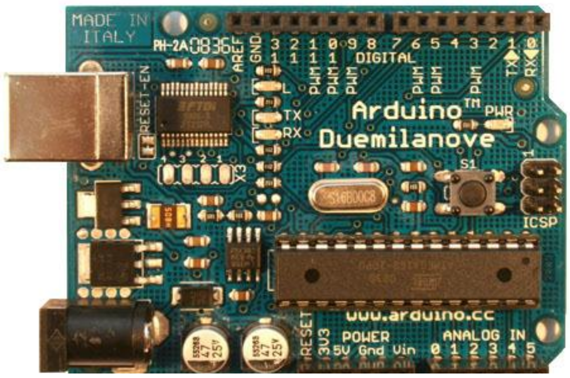

Figure 12 Arduino Duemilanove

4.1.2 Software

Arduino programs divided into three main parts: structure, values (variables and constants) and functions. After writing the code, the Arduino programming environment [**Figure 13**] compiles it (Purdum, 2012).

Problems:

First time that we uploaded the code some problems appeared in the bottom of the program window as syntax error in the program caused by probably a mistake in typing. Generally, staring at the error line will reveal the problem by the following steps such as:

- Run the Arduino program again.
- Check that the USB cable is secure at both ends.
- Reboot your PC because sometimes the serial port can lock up.
- If a “Serial port…already in use” error appears when uploading.

Second is how to choose proper threshold which determines the status of bionic hand (open\close). The solution was by trial and error; which is a fundamental method of solving problems. It is characterized by repeated, varied attempts which are continued until success.

Two required functions / methods / routines:

```
Void setup ()
{
        // runs once
}
Void loop ()
{
        // repeats
}
```

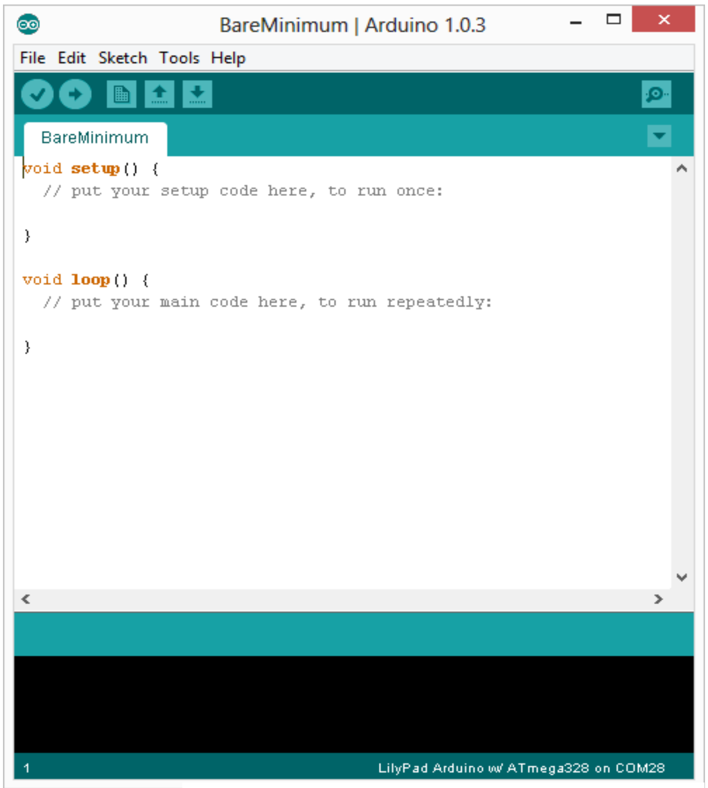

Figure 13 Programming environment

The code implemented is provided below:

```
//these constants will not change. they are used to give names to the pins used:

const int analogIn=0;  //analog input pin that the potentiometer is attached to
const int analogOut=13; //analog output pin that the LED is attached to
int sensorValue=0;           //value read from pot
int outputValue=0;           //value output to the PWM

void setup()                 //initialize serial communication at 9600 bps:
{

  Serial.begin(9600);
  servol.attach(9);          //attach the servo on pin 9 to the servo object
}
void loop(){

  sensorValue=analogRead(analogIn);
  if(sensorValue>50)
  {
    servol.write(170);        //tell servo to go to position in variable 'pos'
    delay(1000);               //waits 1s for the servo to reach the position
  }
  if(sensorValue<=50)
  {
    servol.write(10);         //tell servo to go position in variable 'pos'
    delay(100);
  }

  outputValue=map(sensorValue,0,1023,0,255);   //map it to the range of the analog out
  analogWrite(analogOut, outputValue);        //change the analog out value
```


Figure 14 Project code

4.2 Algorithm

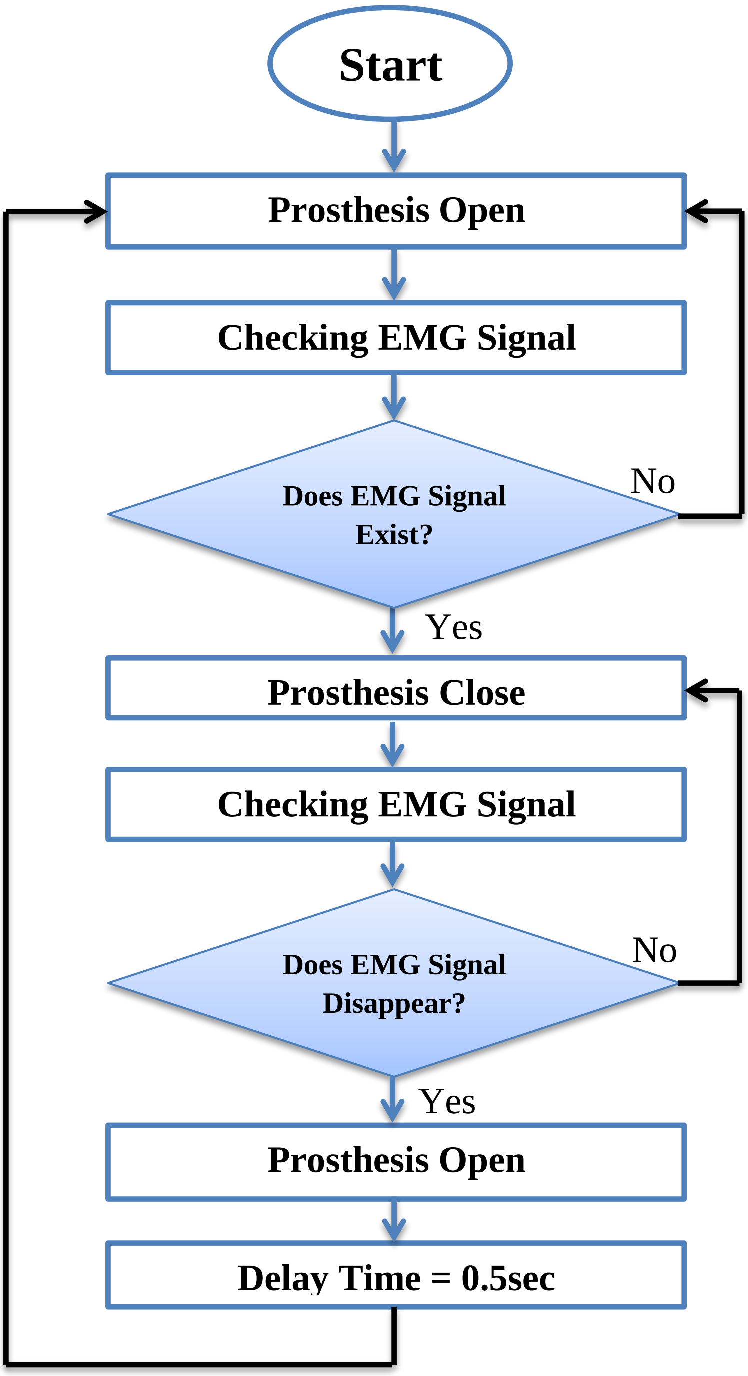

Figure 15 Algorithm

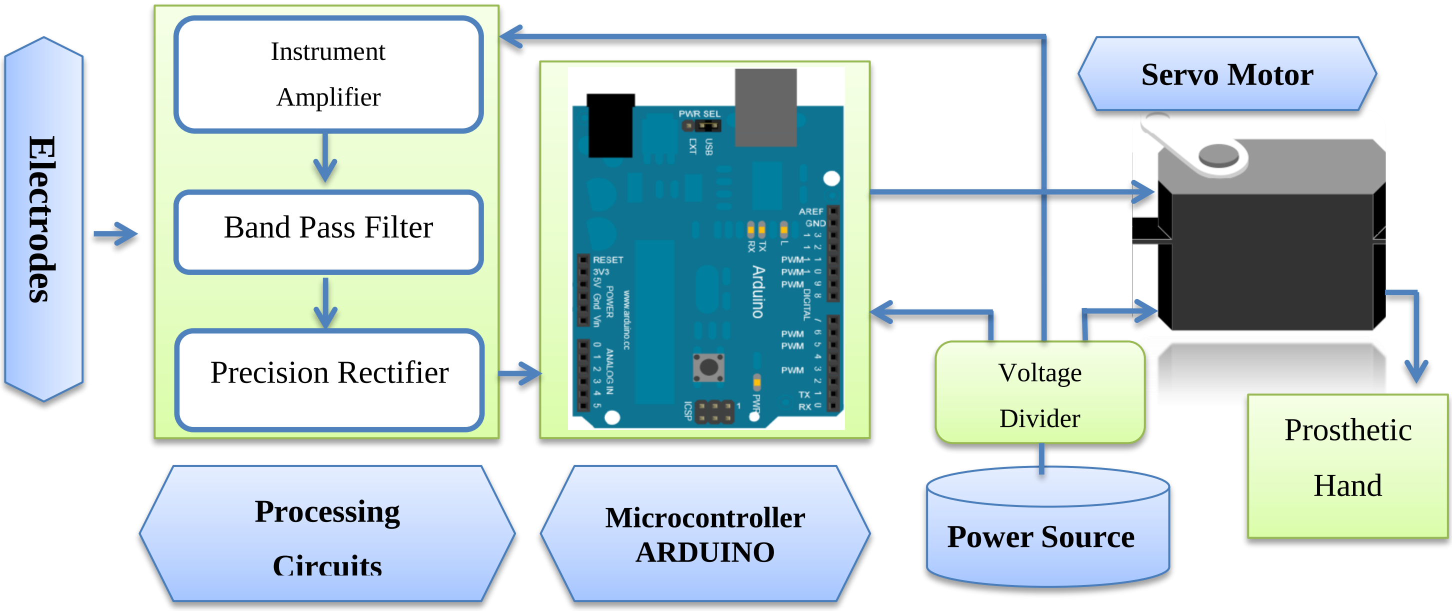

Figure 16 Diagram of main stages

The diagram above describes the three main different stages in the project which are split into:

1) Processing circuits (instrument amplifier – Band pass filter – Precision rectifier).
2) Microcontroller (Arduino) and servo motor.
3) Prosthetic hand.

CHAPTER 5, MECHANICAL DESIGN OF THE HAND

5.1 Servo motor

A few pins on the Arduino allow us to modify the output to mimic a digital signal. This is done by a technique called pulse width modulation (PWM), which is used everywhere such as in Lamp dimmers, motor speed control and power supplies. Three characteristics of PWM signals are pulse width range (min/max), Pulse period (1/pulses per second) and voltage levels (0-5V, for instance).

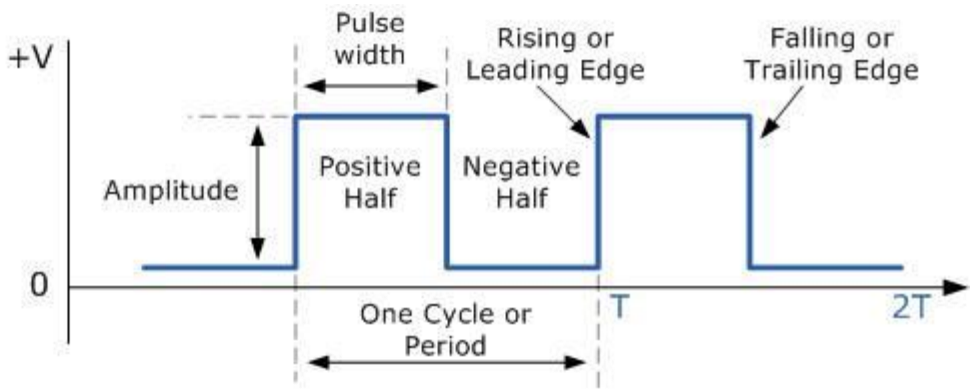

Figure 17 Pulse Width Modulation Ref, (Sen M. Kuo, Bob H. Lee, Wenshun Tian, 2006)

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes. The output shaft of a servo does not rotate freely, but rather is made to seek a particular angular position under electronic control. Servo motors are typically rated by torque and speed.

The potentiometer inside the servo will allow the motor to rotate until the programed position is reached and the motor will stop rotation when reached the position.

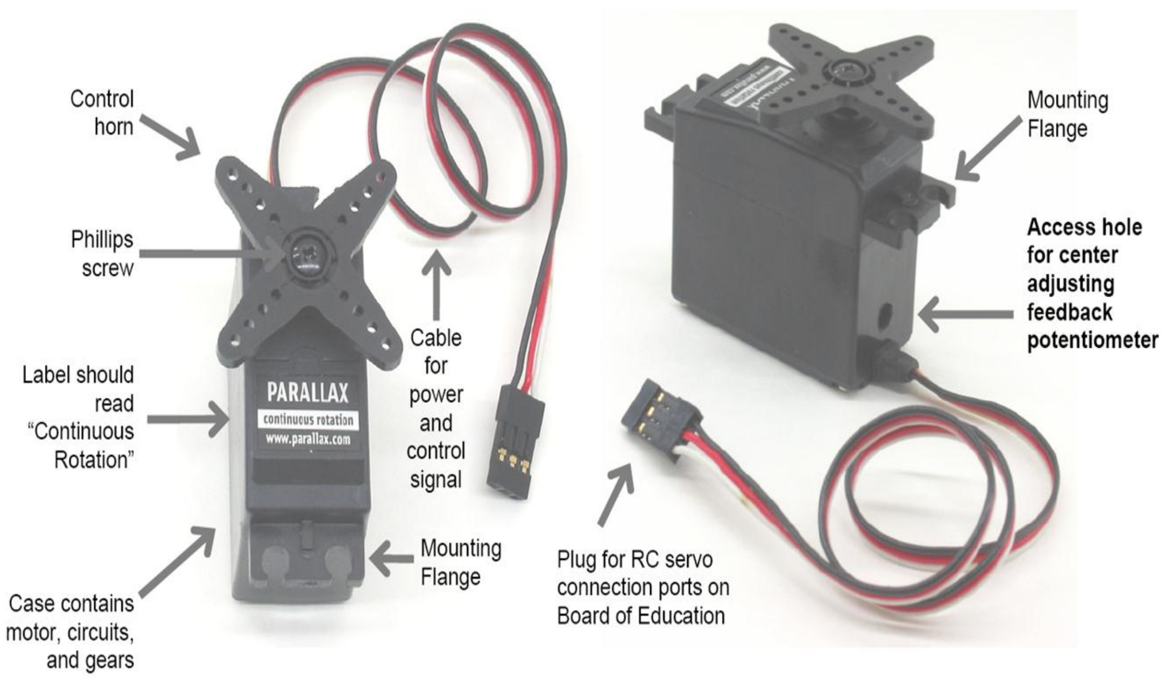

Figure 18 Servo motor components Ref, (Lindsay, 2012)

Servo motor system:

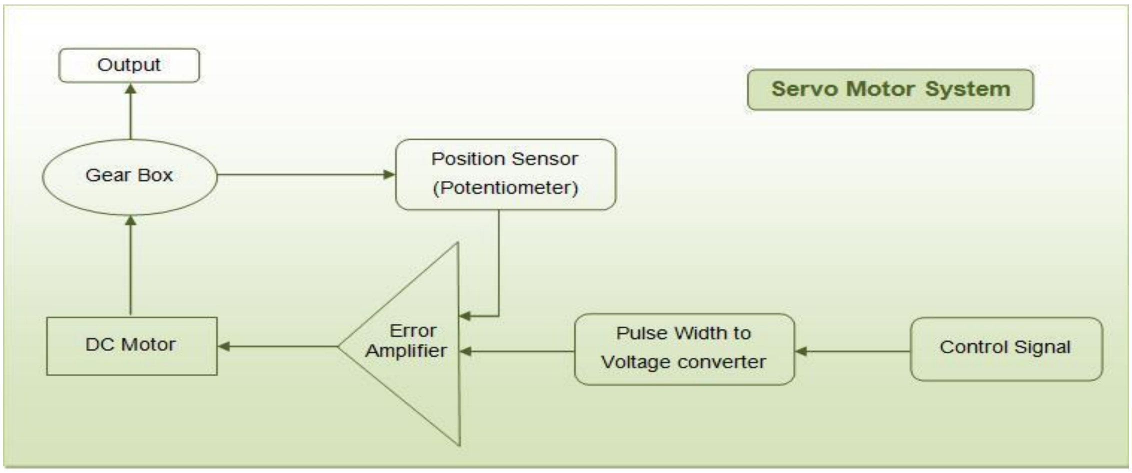

Figure 19 Servo motor circuit Ref, (Masatoshi Nakamura Satoru Goto Nobuhiro Kyura Tao Zhang, 2004)

When the positive peaks (analog input) are delivered to the microcontroller they are converted to a pulse width modulated signal (PWM), then the servo motor receives control signal from the output signal of Arduino.

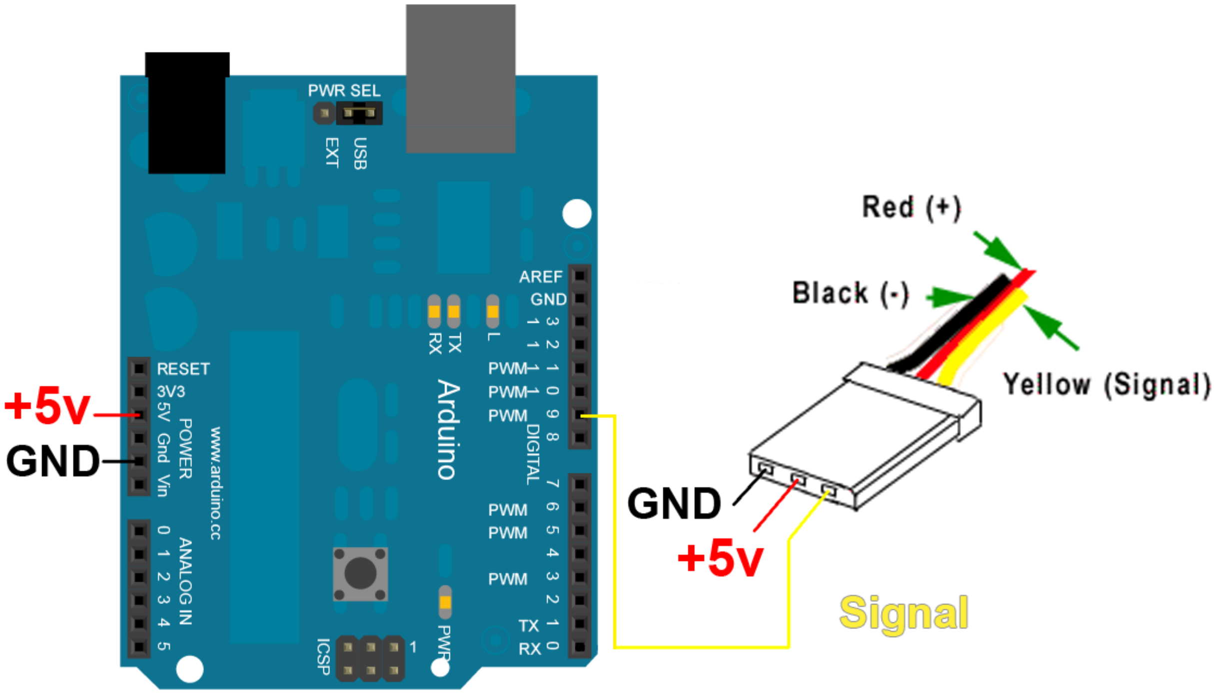

Figure 20 Servo connection to Arduino

The figure above shows that the signal wire (yellow) is connected to the pin number 9 which is specific for PWM, the red\black wires are connected to the power supply (ARDUINO).

5.2 Hand design

The design of the prosthetic hand is a simple design that imitates functionality of real hand. It uses information carried on electromyogram to perform as close as possible to a real hand. In our design we used wood because it's easy to find and not expensive. And also we used other cheap materials to do this design like (thread, twine, and screws). As seen in [Figure 21], our design is simple. That was the best to achieve with limited available sources.

Materials:

- 3/4" hardwood dowel.
- 4" x 1" Pine plank.
- Bags of #216-1/2 small screw eyes (eyelets).
- Roll of thread.
- Roll of twine.
- Bag of 1/4" wide rubber bands.
- Cup hook (open eyelet).
- 3/4" sheet rock screws.

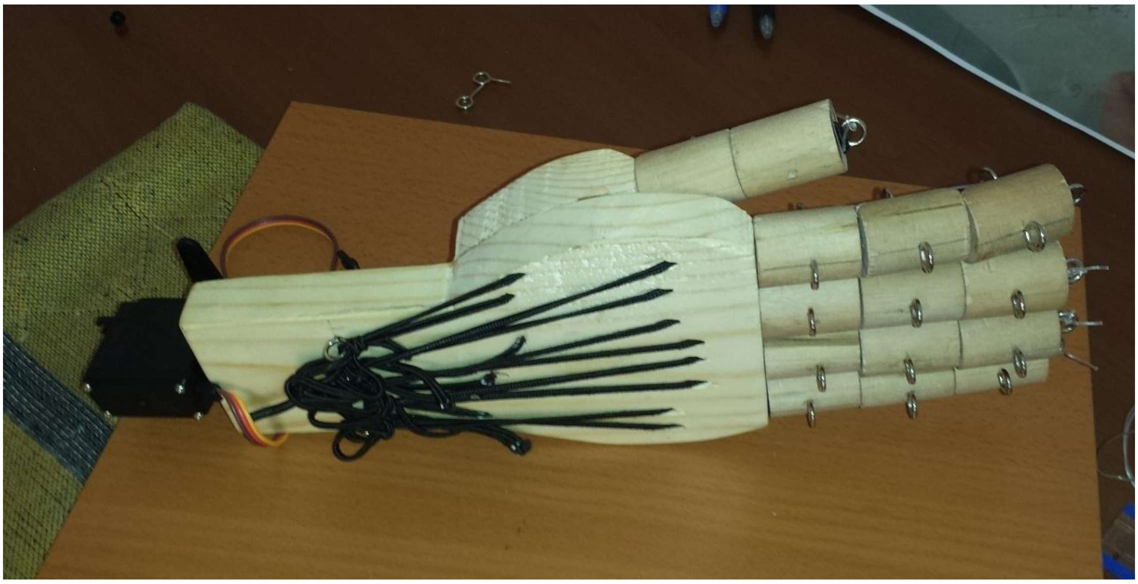

Figure 21 Hand design

Tools:

We used various hand tools, a small drill press, 4" side grinder and a chop saw with a trim blade. These are time savers, but you could do it all with hand tools if you can spend the time.

1-The design template:

We started by drawing the outline of the hand on a piece of wood [Figure 22]. Notice the thumb carpal bone (1st from the wrist) is cut off too. The 1st thumb segment will be shaped from a scrap of the pine 1x (one by) and screwed to the bottom of the hand. Cut out your hand template and transfer it to the pine one by. Extend the wrist lines long enough to make it to the elbow. Cut out this shape with the jig saw and put it aside.

Figure 22 Design template

2-Cut the wooden fingers:

Rather than trying to work out specific lengths for the various finger bones, we just decided to make them all the same. We setup the chop saw with a stop block and C clamp to lock in the length.


Figure 23 Cut the wooden fingers

3-Drilling ligament holes:

Figure 24 Drilling ligament holes

We used the chop saw to cut a 90 degree notch in a scrap. Then we used this and a speed square to mark a center line on each finger bone segment. Once all the parts were marked we lined up the first one with the drill bit and c-clamped the 90 degree notch block to the drill press table to make it easy to drill all the holes accurately and quickly.

4-Using the Grinder

Figure 25 45° slope

We used a 4" side grinder to cut a 45 degree angle on both ends of all but five of the finger bone segments. Five of them should have the 45 degree angle on only one end to make the finger tips. We used a 100 grit sandpaper disk to make the cut. You could also use a bench grinder but be careful that it doesn't grab.

5-The palm and forearm

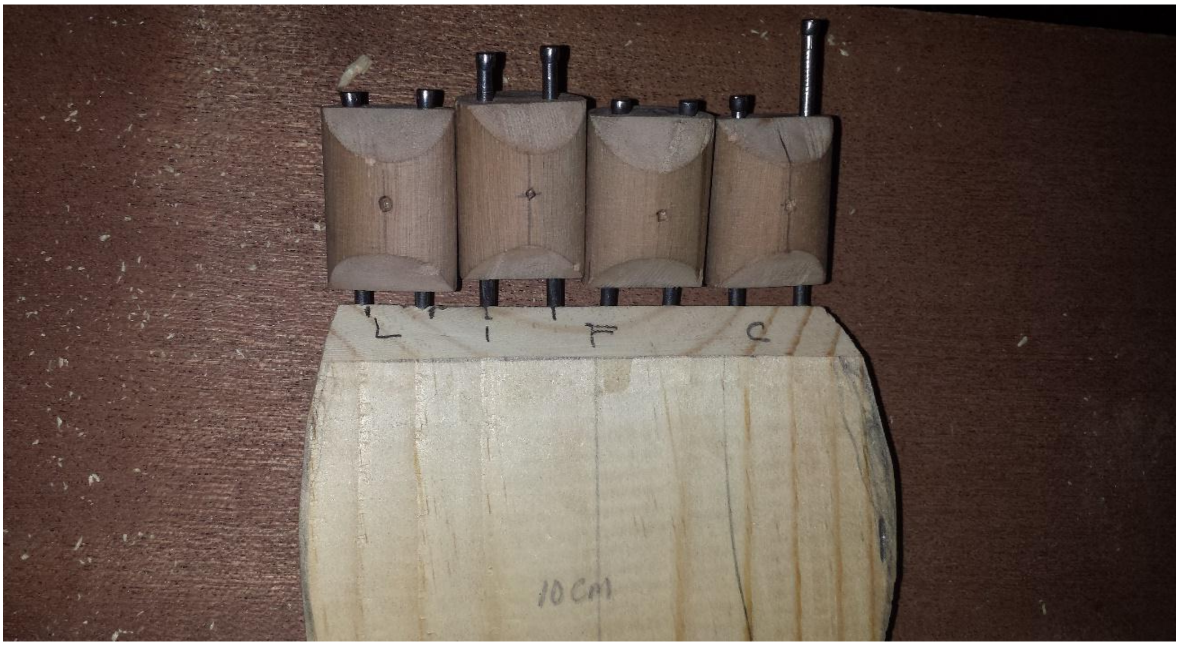

Figure 26 The palm and forearm

We cut a triangular piece of the 1x4 scrap, pre-drilled holes and used sheet rock screws to attach it to the bottom of the hand in the thumb 1st joint spot. We screwed this piece to the hand by using the 4" side grinder to shape and smooth the entire piece. We need to cut a 45 degree flat on the underside of where the fingers attach. This slope will mate up with the slope on the first bone segment of the fingers. We also need to grind off a slope on the inside of the thumb attachment point. The sloped parts of the joints must rest against the sloped part of the hand when the finger is folded.

Figure 27 Hand assembly

6-Ligaments and Tendons

Drill a pilot hole on the inside near far end of each joint piece (except the finger tips) and screw in the small eyelets. The finger tips pieces get an eyelet closer to the inner slope just inside of center. Drill a pilot hole in the center of the end of the fingertip and add the eyelets [Figure 28]. We decided to add eyelets & tendons as a twine to the back of fingers [Figure 29], this makes it more complex, but realistic, then take each twine to servo motor. For ligaments we used rubber thread to make the fingers go back easily after tighten.

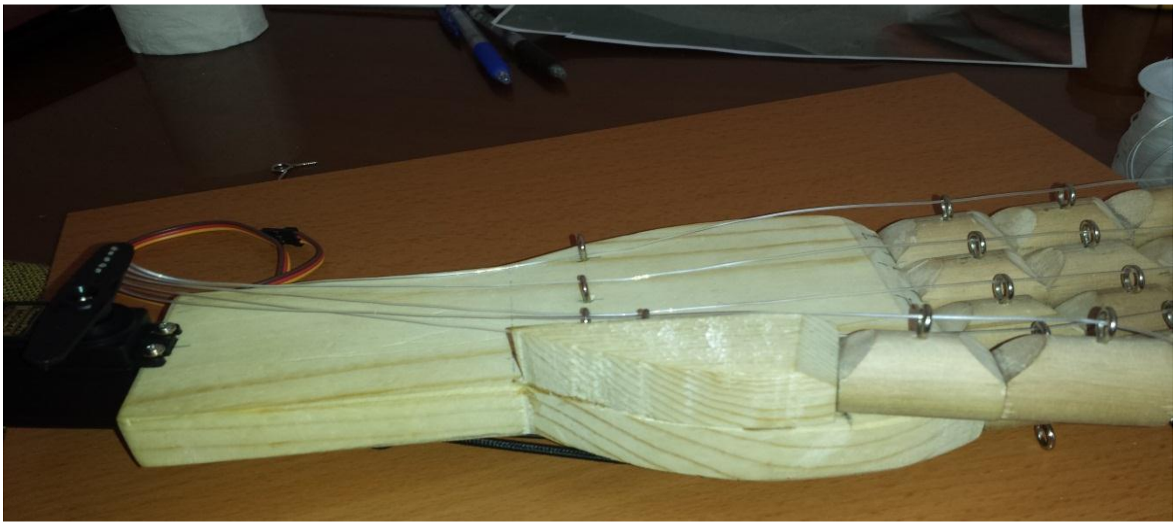

Figure 28 Hand tendons

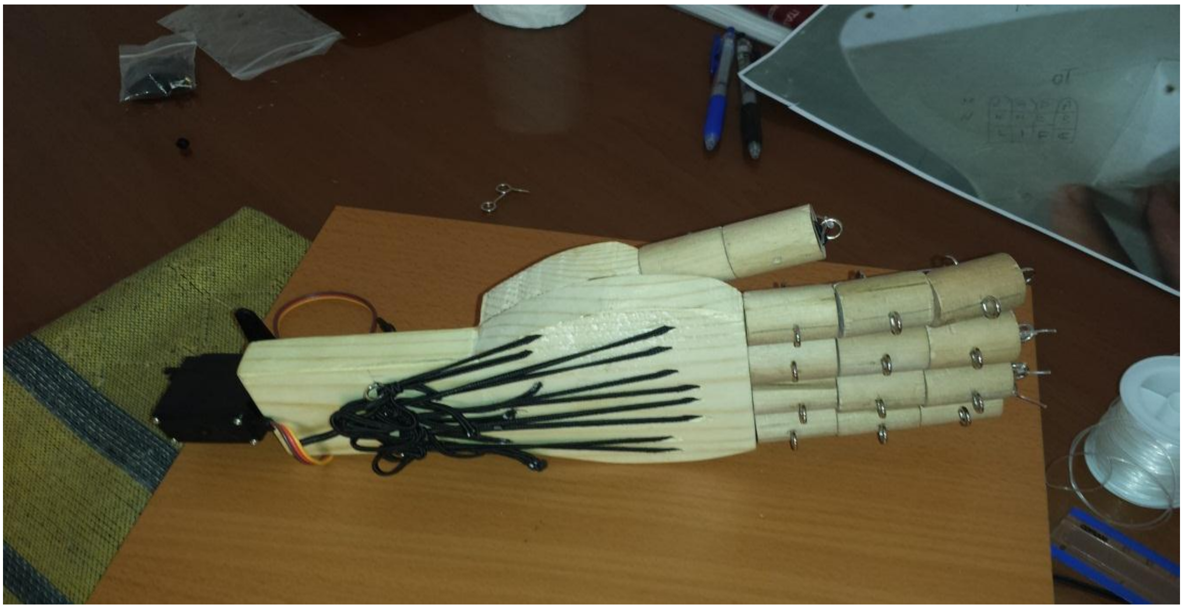

Figure 29 Hand ligaments

Chapter 6, RESULTS AND DISCUSSION

After we finished assembling mechanical components and electronic elements to take the final form, we had to test the hand on different subjects and the results were as below:

During the assembly stage a few problems showed up:

- The differential amplifier INA118P that we bought turned out to be faulty. As a result we had to use the EMG kit that is available at the biomedical lab at our university to detect the muscle signal. The EMG kit was unstable and hard to move around due to its big size.
- EMG leads are difficult to find. This problem was also solved by using the university kit.
- We used standard servo motor (180°) instead of full rotation one which usually have a short range of movement. We extended that range of contraction by diverting all tendon wires through specific lanes on front of the hand which lengthened the motion span.
- The hand was not able to return to the opening position after being contracted by the servo. That is why we added ligament like rubber ties to flex the hand to initial position.

During the operating stage we also faced a couple of problems:

- The most important parameter in our design is the threshold value which determines the status of the prosthesis. Selecting this value was done through trial and error method.
- The closing shift of the cycle is usually opposed by some force due to the load applied to it and the inertia caused by wooden material of the hand. The microprocessor does not recognize any commends while executing another. We added a delay time to closing shift cycle to give the microcontroller enough time to execute previous orders.
- After using the hand for several times, the muscles usually suffered from fatigue. This led to erroneous and random contractions. This issue can be solved by using high-sensitivity electronics and by raising the threshold value.

Overall the project accomplished our expectation and fulfilled our purpose of the hand.

CHAPTER 7, CONCLUSIONS AND FUTURE ADVANCES

Conclusion:

Prosthetic hand gives hope for amputees around the world to recapture their ability to perform complicated physical movement. We used EMG signal, because this signal is much more involved in the grasp movement, as control signal which is processed by microcontroller to obtain a pulse width modulated signal as a simple and practical approach to accomplish our purpose in order to increase the effectiveness of the hand movement. The electrical activity of the biceps muscle allows us to know whether the patient is trying to grip his hand.

Future advances:

The project can undergo many development and advances in order to become more suitable for clinical use. These advances can be in the materials, design or level of complexity to be more functional and easier to be worn by the patient. Multiple thresholds for more precise movement, 3D printed hand from composite material that allow light weight and close to reality hand and more compact design are among many advances that can be achieved through more researches. An implanted electrode with wireless transmission capability is an important feature that can minimize the use of wires for more mobile and compact design. Furthermore, artificial skin to sense pressure and heat can be used as feedback sensory to avoid damage or harm for the person and the prosthetic itself. Nevertheless it may be necessary to use smaller servo motors, because the one we used is too cumbersome and too powerful for our specific purpose. Moreover, it is also to develop more advanced technics for the EMG processing, that taking into account the natural variability of these signals. For this purpose the recording of the myoelectric activity from other sites can turn out useful.

Bibliography

Masatoshi Nakamura Satoru Goto Nobuhiro Kyura Tao Zhang. (2004). Mechatronic Servo System Control. Springer.

ARDUINO. (n.d.). ComponentLib/Servo. Retrieved 10 14, 2014, from arduino: http://playground.arduino.cc/ComponentLib/Servo

Carr, Joseph J. Brown, John M. (2001). Introduction to Biomedical Equipment Technology (4th Edition). New Jersey: Prentice Hall.

D. Reinkensmeyer, H. Hogan, H. Krebs, S. Lehma, and P. Lum. (2000). Biomechanics and Neural Control of Posture and Movement. Eds Winters J. Springer.

D. Reinkensmyer, L. K.-C. (2000). Understandning and Treating Arm Movement Impairment After Chronic Brain Injury:Progress with The Arm Guide. Journal of Rehabilitation Research and Developement.

J. Kenneth Salisbury John J. Craig. (1982). Articulated Hands Force Control and Kinematic Issues. International Journal of Robotic Research.

J. Kenneth Salisbury John J. Craig. (n.d.). Articulated Hands Force Control and Kinematic Issues. The International Journal of Robotics Research.

John E. Hall, P. (2010). Guyton and Hall Textbook of Medical Physiology, 12th Edition. London: Grune and stratton Inc.

Khurana, I. (2009). Textbook Of Medical Physiology. India: Elsevier.

Light, C.M. and Chappell, P.H. . (2006). Development of a lightweight and adaptable multiple-axis hand prosthesis.

Lindsay, A. (2012, 2 29). Activity 4: Connect Servo Motors and Batteries. Retrieved 9 23, 2014, from PARALLAX Inc: http://learn.parallax.com/node/180

McRoberts, M. (2013). Begining Arduino, 2nd Edition. New York: Apress.

Merletti, R. (2004). Electromyography: Physiology, Engineering, and Non-Invasive Applications. New Jersey: John Wiley and Sons Inc.

Purdum, J. (2012). Begining C for Arduino. New York: Apress.

Sen M. Kuo, Bob H. Lee, Wenshun Tian. (2006). Real-Time Digital Signal Processing, 2nd Edition. Wiley.

Appendices

Appendix 1: source code

```
#include <Servo.h>

Servo myservo;     //create servo object to control a servo
//these constants will not change. they are used to give names to the pins used:

const int analogInPin=0;  //analog input pin that the potentiometer is attached to
const int analogOutPin=13; //analog output pin that thr LED is attached to
int sensorValue=0;          //value read from pot
int outputValue=0;          //value output to PWM

void setup()                //initialize serial communication at 9600 bps:
{

  Serial.begin(9600);
  myservo.attach(9);         //attach the servo on pin 9 to the servo object
}
void loop(){

  sensorValue=analogRead(analogInPin);
  if(sensorValue>70)
  {
    myservo.write(170);       //tell servo to go to position in variable 'pos'
    delay(1000);              //waits 1s for the servo to reach the position
  }
  if(sensorValue<=70)
  {
    myservo.write(10);        //tell servo to go position in variable 'pos'
    delay(100);
  }

  outputValue=map(sensorValue,0,1023,0,255);   //map it to the range of the analog ou:
  analogWrite(analogOutPin, outputValue);       //change the analog out value:

}
```

Appendix 2: datasheets and schematic

http://www.ti.com/lit/ds/symlink/ina118.pdf

http://pdf.datasheetcatalog.com/datasheet2/b/0dditqczh92y425gj9ehipgzzgyy.pdf

http://brittonkerin.com/annotateduino/annotatable_duemilanove.html