UNIVERSITY OF SOUTHREN QUEENSLAND

FUCULTY OF ENGINEERING AND SURVERYING

The Design and Construction of a Large Scale- Force Feedback Joystick for the use in Physiotherapy

Dissertation submitted by

Pako Maruping

In fulfillment of the requirement of

Courses ENG4111 and ENG4112 Research project

Towards the degree of

Bachelor of Engineering (Mechatronic)

Submitted:

ABSTRACT

The design and construction of the joystick in this project is aimed at helping stroke patients in rehabilitation, the joystick will help exercise the wrist muscles. The joystick will have assistive behavior. This assistive behavior will work such that a patient will follow a target and if they go off course the joystick will act as to assist them to go back on course.

A cube steel structure was designed as the structure for the joystick, the joystick was built using 25 x 25 x 3 steel beams this steel structure is to house the motors as well. A pulley system was used to transmit the motion of the motor to the stick and to transmit the motion of the stick to the motors. With a potentiometer mounted on the motor shaft as to determine the motor position. An Arduino Deicimila development board used to control the motors hence the joystick. A servo motor was used in this project.

The motors were be programmed to move back and forth in a straight line and the force feedback will be applied so as the motors to help the patient move in the right direction a PID was be incorporated in to the control system so as to correct the errors in the shortest time possible. If time permits the joystick will be programmed to move in a rotational motion so as to get the total wrist exercise. The design of the casing for the joystick, the hardware and software were completed, testing was completed as well.

University of Southern Queensland

Faculty of Engineering and Surveying

ENG4111 Research Project Part 1 & ENG4112 Research Project Part 2

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course "Project and Dissertation" is to contribute to the overall education within the student's chosen degree programme. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen
Dean
Faculty of Engineering and Surveying

CERTIFICATION

I certify that the ideas, designs and experimental work, results, analyses and the conclusions set out in the dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Pako Maruping

Student number: 0050030132

____

Signature

____

Date

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Selvan Pather for his support, guidance, help and motivation throughout the conduct of this project without which completion of this project would have been a difficult task.

I would like to extend my appreciation to my family for their support.

Table of Contents

CHAPTER 1: INTRODUCTION 1

1.1 NEED FOR THE PROJECT 1

1.2 PROJECT SPECIFICATION 1

1.3 METHODOLOGY 3

1.4 REVIEW OF DISSERTATION 4

1.5 CONCLUSION 4

CHAPTER 2 : BACKGROUND ON STROKE AND STROKE THERAPIES 5

2.1 INTRODUCTION 5

2.2 STROKE 5

2.3 TYPES OF STROKE 5

2.3.1 Ischaemic stroke 5

2.3.2 Haemorrhagic stroke 7

2.4 EFFECTS OF STROKE 9

2.4.1 General effects of stroke 9

2.4.2 Effects of Right Hemisphere Strokes 10

2.4.3 Effects of Left Hemisphere Strokes 10

2.4.4 Effects of Cerebellum Strokes 11

2.4.5 Effects of Brain Stem Strokes 11

2.4.6 Other Effects of Stroke 11

2.5 PHYSIOTHERAPHY 12

2.6 NEW EXPERIMENTAL THERAPY 13

2.6.1 Robot-Assisted Rehabilitation with Contraint-Induced Movement Therapy (CIMT)13

2.7 CONCLUSION 17

CHAPTER 3 : JOYSTICKS 18

3.1 INTRODUCTION 18

3.2 TYPES OF JOYSTICKS AND OPERATIONS 18

3.2.1 The Simplest Systems 18

3.2.2 Conventional Analog joystick 20

3.2.3 Analog to digital conversion 21

3.2.4 The force feedback joystick 22

3.3 CONCLUSION 23

CHAPTER 4 : CONCEPT DESIGNS 24

4.1 INTRODUCTION 24

4.2 CONCEPT DESIGNS AND ANALYSIS 24

4.2.1 Initial idea 1 24

4.2.2 Initial idea 2 26

4.2.3 Initial idea 3 26

4.3 FINAL DESIGN 27

4.4 CONCLUSION 29

CHAPTER 5 : FINAL DESIGN AND PROTOTYPE 30

5.1 INTRODUCTION 30

5.2 STRUCTURE 30

5.2.1 Pulley positions and length of cord 31

5.3 PROTOTYPE 32

5.4 ELECTRONICS 33

5.4.1 MOTOR SELECTION 33

5.4.2 Servo motor VS DC motor 33

5.4.3 Servo motor VS stepper motor 34

5.4.4 Servo motor 36

5.4.5 HARDWARE 41

5.4.6 The design and construction of the hardware 41

5.4.7 The olimex AVR - MT - 128 development board for ATMEGA128 microcontroller. 42

5.4.8 The Arduino Diecimila development board for ATMEGE168 microcontroller .43

5.4.9 Arduino Diecimila 45

5.5 SOFTWARE 49

5.5.1 Arduino Environment 50

5.6 CONCLUSION 50

CHAPTER 6 : SYSTEM EVALUATION AND ANALYSIS OF RESULTS 51

6.1 INTRODUCTION 51

6.2 TESTING AND ANALYSIS 51

6.3 CONCLUSION 56

CHAPTER 7 : CONCLUSION AND FUTURE WORK 57

7.1 INTRODUCTION 57

7.2 SUMMARY OF PROJECT 57

7.3 FUTURE WORK 58

REFERENCES 59

APPENDIX A: PROJECT SPECIFICATION 63

APPENDIX B: ARDUINO SCHEMATIC 64

APPENDIX C: DESIGN DRAWINGS 65

APPENDIX D: MODEL PICTURES 68

APPENDIX E THE PROGRAM 72

APPENDIX F: Calculating the length of the cord around the pulleys 77

APPENDIX G: NOTES ON PID 82

TABLE OF FIGURES:

Figure 2-1: Embolic stroke (source: GHI- Health and wellness 2005) 6

Figure 2-2: Thrombotic stroke (source: strokecenter 2008) 7

Figure 2-3: Intracerebral haemorrhage (source: strokecenter 2008)... 8

Figure 2-4 : Subarachnoid haemorrhage (source: strokecenter 2008) 9

Figure 2-5: Haptic joystick (source: Mechatronics and Haptic Interfaces Lab 2008)........... 14

Figure 2-6: experiment on Haptic joystick (source: Mechatronics and Haptic Interfaces Lab 2008) 14

Figure 2-7: modified robot arm (PUMA) (source: Mechatronics and Haptic Interfaces Lab 2008) 15

Figure 2-8:robotic interface (source: Mechatronics and Haptic Interfaces Lab 2008)........... 16

Figure 2-9(source: Mechatronics and Haptic Interfaces Lab 2008) 17

Figure 3-1: contact buttons (source: howstuffworks 1998) 19

Figure 3-2: contact pin (source: howstuffworks 1998).. 20

Figure 3-3: analog joystick (source: howstuffworks 1998) 21

Figure 3-4: potentiometer (source: howstuffworks 1998) 21

Figure 3-5: force feedback joystick **(**source: howstuffworks 1998**)** 23

Figure 4-1: initial idea 1... 25

Figure 4-2: cable on motor shaft and cam pin 25

Figure 4-3**:** axel movement of the joystick 25

Figure 4-4: steward platform (source: Wikipedia, the free encyclopedia) 26

Figure 4-5: spring – cam joystick 27

Figure 4-6: final design.. 28

Figure 4-7: top view of the final design.. 28

Figure 5-1: final design.. 30

Figure 5-2: PROTOTYPE.. 32

Figure 5-3: DC motor (source: Fasco 2008) 34

Figure 5-4: stepper motor (source: Futurlec 2008) 35

Figure 5-5: servo motor (source: servocity 2008) 37

Figure 5-6: time vs. angle 38

Figure 5-7: servo metal gears 39

Figure 5-8: AC servo motor (source: baldor) 41

Figure 5-9: AVR - MT – 128 (source: olimex 1999) 42

Figure 5-10**:** Arduino Diecimila (source: Arduino 2008) 43

Figure 5-11: Arduino Diecimila (source: Arduino 2008) 45

Figure 5-12: servo motor connection to arduino (source: electronics-lab 2002) 49

Figure 5-13: Arduino environment 50

6-1: potentiometer position 52

6-2: hyper terminal showing the results from serial communication 52

Figure 6-3: decoupling with a capacitor (source: Code, circuits, & construction, 2008) 53

Figure 6-4: results 54

Figure 6-5: results 55

Figure 6-6: results 56

CHAPTER 1: INTRODUCTION

In this chapter an introduction to the project is made. Starting the reason the project was undertaken. The project specification will be set as target goals for the project. A review of the whole dissertation will be under taken, and finally discussing the methodology in which the project will ran.

1.1 NEED FOR THE PROJECT

In Australia someone has a stroke about every 11 minutes. Stroke affects around 48 000 Australians each year. 2008 (Striking back at stroke). It's important to know that people do recover from stroke. The patients have to go though physiotherapy to help them recover. This project serves as to try and improve the mode of therapy.

1.2 PROJECT SPECIFICATION

This project aims are to design and construct a large scale force feedback joystick which will be used in Physiotherapy for arm rehabilitation. The joystick will initially assist weak patients as the motions of the motor will be manipulated as to assist the patient towards a target position. A series of steps or specifications need to be satisfied in order to achieve this task.

The following steps are the project specification:

- Research information on rehabilitations of stroke patients.

- Research on current technology used in Physiotherapy.

- Design a large scale force feedback joystick which will have north-south and east-west motion.

- Construction of the joystick.

- Design the software for the motors which will be used in the motions of the joystick

- Interface the joystick with software and test

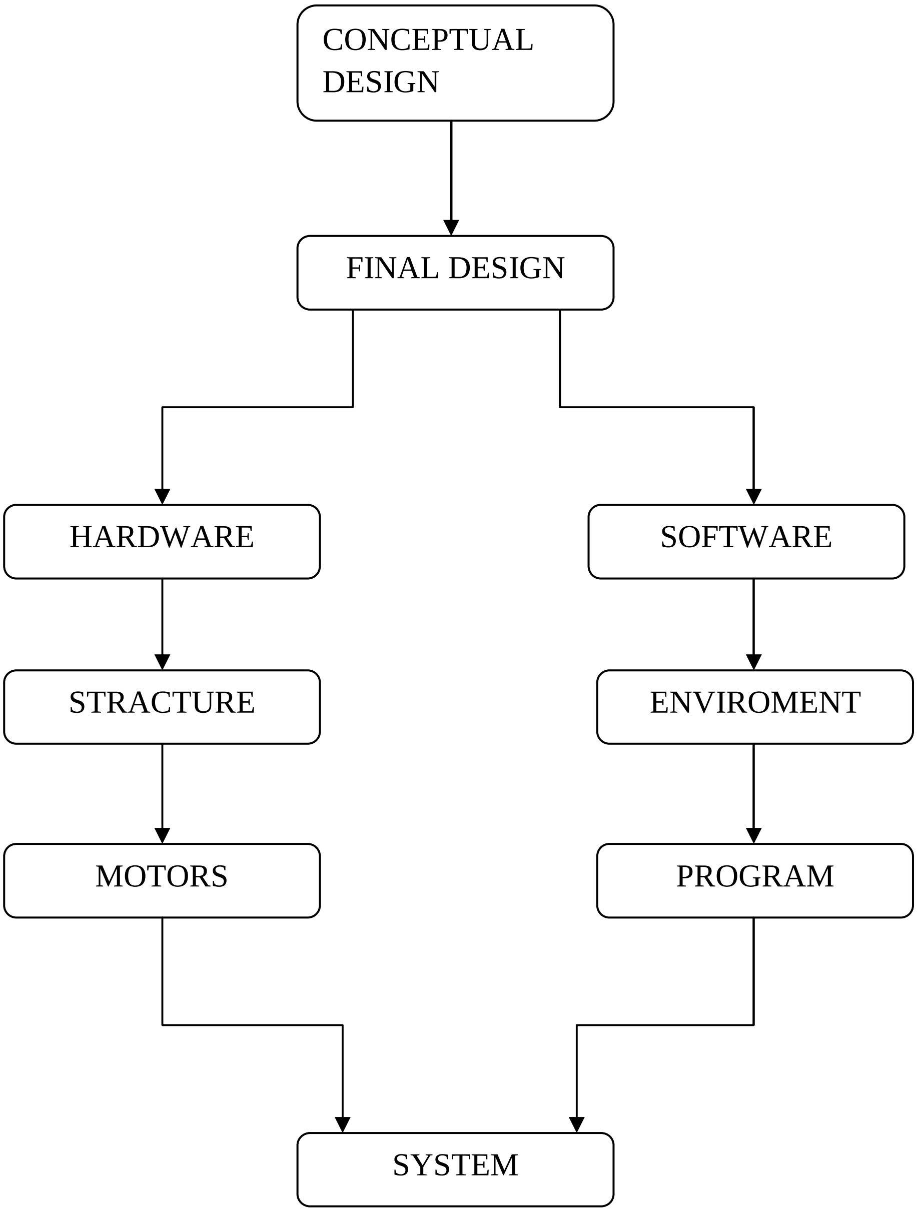
As time permits

- Include rotational motion in the motions of the joystick

- Human trials

1.3 METHODOLOGY

Several steps were taken to achieve the goals of the project. The concept design was the first part of the project following the concept design, a final design was done then the design had to be split in two parts namely: hardware and software, for more detailed analysis. With the hardware the design was about the structure of the system and motors to be used. With the software the environment of the software was discussed as well as building the program.



1.4 REVIEW OF DISSERTATION

This dissertation is made of seven chapters. Chapter 1 introduces the project and gives a brief outline of the project specification. Chapter 2 deals with the background information on stroke and stroke therapies, which give an understanding of conditions of people who have suffered a stroke. Chapter 3 gives a background on joysticks, types of joysticks and how they work.

In Chapter 4 the development of concept designs are discussed and the final design. Chapter 5 provides a detailed analysis of the final design. Chapter 5 it also provides the hardware and software design as well as the motor selection. Chapter 6 deals with system evaluation and analysis of the results. Chapter 7 gives a conclusion and discusses future work on the project.

1.5 CONCLUSION

The project has been introduced with the specification discussed, and a review of the structure of the dissertation was taken. A methodology diagram was created to guide on which way to conduct this research. Having introduced the project, some research has to be done on stroke, to have an understanding of the illness as well as understand what the stroke patients have to go though and current methods of rehabilitation.

CHAPTER 2 : BACKGROUND ON STROKE AND STROKE THERAPIES

2.1 INTRODUCTION

This chapter gives a background on stroke. Types of stroke and their effects on people will be discussed. Information on stroke was fairly easy to find, I went to the Toowoomba base hospital for some pamphlets on stroke, went through the library books and the internet was the major source of information.

2.2 STROKE

Cerebral neurons (brain cells) can function normally only with a constant supply of oxygen and glucose, this requires an adequate cerebral blood flow. The most common form of a stroke is the result of a sudden disruption in that cerebral blood flow, cerebral neurons thrive in oxygen so the disruption in blood flow will result in staving the brain cell in which case they will die (infarct). In other cases the blood vessels bursts causing serve bleeding in the brain, the blood will spread to other parts of the brain which will result in a stroke. Functions of the body that are normally controlled by these brain cells will be affected, in most cases unconsciousness and partial paralysis will occur these are the early signs of stroke. (Strokefoundation 2008)

2.3 TYPES OF STROKE

There are two main types of stroke namely Ischaemic and Haemorrhagic stroke.

2.3.1 Ischaemic stroke

Normally blood clotting is very useful. When a person has an injury and they are bleeding, blood clots works to slow and eventually stop the bleeding. However in case of Ischaemic stroke blood clotting is the main cause of stroke as the blood clot blocks the blood flow to the some parts of brain. This can happen in two ways:

2.3.1.1 Embolic stroke

This is when a blood clot forms somewhere in the body and travels though the blood vessels to the brain where the blood vessel's diameter decreases trapping the clot and hence disrupting the blood flow by blocking the blood vessel. This blockage is not always caused by a blood clot, sometimes it can be a fat particle, or bubble of air. Most emboli arise from the heart or large arteries and lodge in a cerebral vessel. The origins of the carotid (neck arteries) from the aorta are a major site of the atherosclerotic (arterial disease with cholesterol deposits plaques forming on the inner surface), parts of these deposits can be detached and flow with the blood to the brain.

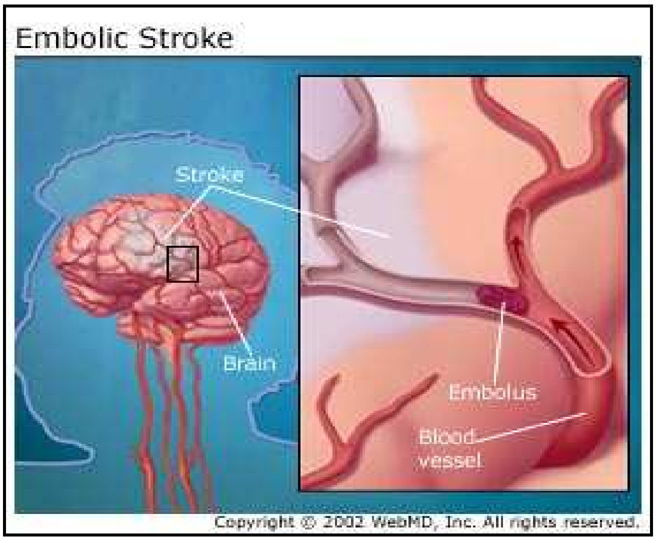


Figure 2-1: Embolic stroke (source: GHI- Health and wellness 2005)

2.3.1.2 Thrombotic

This occurs when a blood vessel narrow as a result of blood fat, cholesterol or calcium which grow to completely block the blood flow in that vessel as we can see in figure 2.2. Some cholesterol deposits are common on the carotid artery, middle cerebral artery, at the junction of vertebral anterior cerebral arteries and vertebral arteries. This blockage will cause tiny infarcts that eventually become a cystic (spherical swelling) and are called ‘lacunae’. In most cases cerebral infarction occurs as a result of occlusion of a single feeding cerebral artery. (P.E Kaplan, MD,L.J Cerullo, MD 1940)

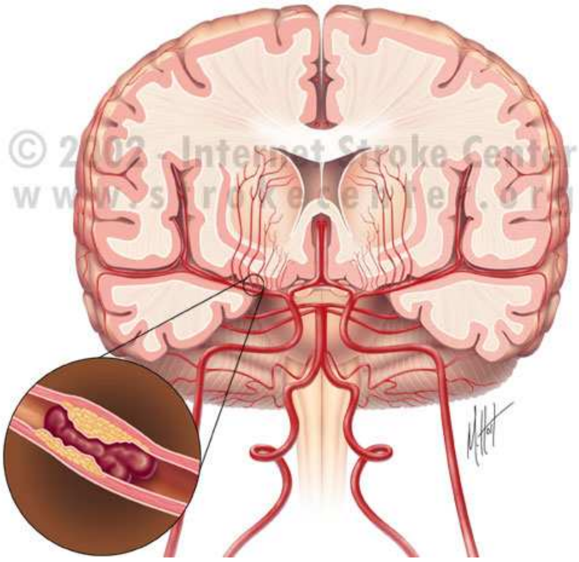


Figure 2-2: Thrombotic stroke (source: strokecenter 2008)

2.3.2 Haemorrhagic stroke

A haemorrhage stroke is caused when a blood vessel in the brain ruptures and there is bleeding in the brain. There are mainly two types of haemorrhage strokes which are intracerebral haemorrhage and subarachnoid haemorrhage.

2.3.2.1 Intracerebral haemorrhage

This is the most common form of haemorrhage stroke, this is when the bleeding occurs within the brain tissue itself which is shown in figure 2.3. This usually occurs with hypertension, which causes the direct damage to the artery wall. Sometimes there are multiple micro-infarcts and haemorrhages in the subcortical areas (the parts of the brain that lie immediately beneath the cerebral cortex).

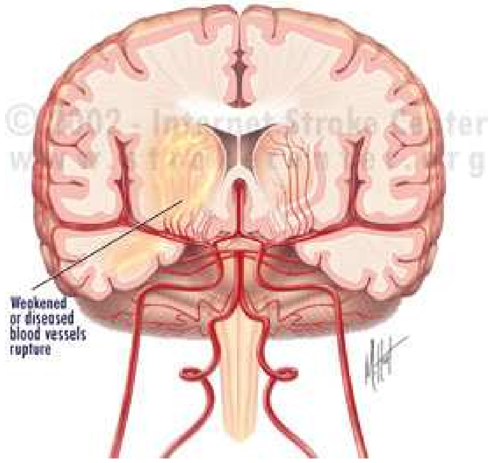

© 2002 - Internet Stroke Center
www.strokecenter.org
Weakened or diseased blood vessels rupture

Figure 2-3: Intracerebral haemorrhage (source: strokecenter 2008)

2.3.2.2 Subarachnoid haemorrhage

In this case bleeding occurs in the space around the brain which is often due to an aneurysm (a fluid-filled sac in the wall of an artery that can weaken the wall), which is shown in figure 2.4. 90 percent of these aneurysms are caused by congenital weakness of the arterial wall in the subarachnoid space (situated beneath the middle of the three membranes arachnoids that cover the brain and spinal cord). When rupture occurs blood pours out the subarachnoid space and some of it leaks into the brain substance. (P.E Kaplan, MD & L.J Cerullo, MD 1940)

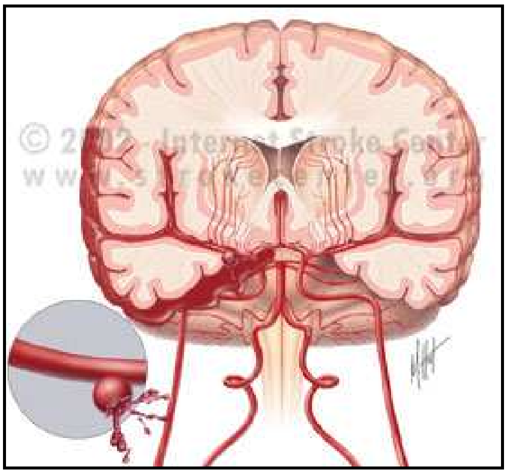


Figure 2-4 : Subarachnoid haemorrhage (source: strokecenter 2008)

2.4 EFFECTS OF STROKE

As having a stroke kills some of the brain cell the patient will lose control of these parts of the body controlled by these cells. Different types of stroke affect the body in many different ways. Some of the effects are discussed in this section.

2.4.1 General effects of stroke

Some general effects of stroke are shown below:

- Weakness or lack of movement (paralysis) in legs and/or arms
- Shoulder pain
- Trouble swallowing
- Changes to way things are seen or felt (perceptual problems)
- Changes to the way things are felt when touched (sensory problems)
- Problems thinking or remembering (cognitive problems)

- Trouble speaking, reading or writing
- Incontinence
- Feeling depressed
- Problems controlling feelings
- Tiredness

(Americanheart 2008)

2.4.2 Effects of Right Hemisphere Strokes

The right hemisphere of the brain controls the movement of the left side of the body so stroke in the right hemisphere often causes paralysis in the left side of the body. This is known as left hemiplegia. Survivors of right-hemisphere strokes may also have problems with their spatial and perceptual abilities. This may cause them to misjudge distances (leading to a fall) or be unable to guide their hands to pick up an object, button a shirt or tie their shoes. They may even be unable to tell right side up from upside-down when trying to read.

Along with these physical effects, survivors of right-hemisphere strokes often have judgment difficulties that show up in their behaviour. They often act impulsively, unaware of their impairments and certain of their ability to perform the same tasks as before the stroke. This can be extremely dangerous. It may lead them to try to walk without aid or to try to drive a car. Survivors of right-hemisphere strokes may also experience left-sided neglect. This is a result of visual difficulties that cause them to "forget" or "ignore" objects or people on their left side. Some survivors of right-hemisphere strokes will experience problems with short-term memory. Although they may be able to recall a visit to the seashore that took place 30 years ago, they may be unable to remember what they ate for breakfast that morning.

2.4.3 Effects of Left Hemisphere Strokes

The left hemisphere of the brain controls the movement of the right side of the body. It also controls speech and language abilities for most people. A left-hemisphere stroke often causes paralysis of the right side of the body. This is known as right hemiplegia. Someone who has had a left-hemisphere stroke may also develop aphasia. Aphasia is a catch all term used to describe a wide range of speech and language problems. These problems can be highly specific, affecting only one part of the patient's ability to communicate, such as the ability to move their speech-related muscles to talk properly. The same patient may be completely unimpaired when it comes to writing, reading or understanding speech.

In contrast to survivors of right-hemisphere stroke, patients who have had a left-hemisphere stroke often develop a slow and cautious behaviour. They may need frequent instruction and feedback to finish tasks. Patients with left-hemisphere stroke may develop memory problems similar to those of right-hemisphere stroke survivors. These problems can include shortened retention spans, difficulty in learning new information and problems in conceptualising and generalising. (Stroke - the after effects 2007)

2.4.4 Effects of Cerebellum Strokes

The cerebellum controls many of our reflexes and much of our balance and coordination. A stroke that takes place in the cerebellum can cause abnormal reflexes of the head and torso, coordination and balance problems, dizziness, nausea and vomiting.

2.4.5 Effects of Brain Stem Strokes

Strokes that occur in the brain stem are especially devastating. The brain stem is the area of the brain that controls all of our involuntary functions, such as breathing rate, blood pressure and heart beat. The brain stem also controls abilities such as eye movements, hearing, speech and swallowing. Since impulses generated in the brain's hemispheres must travel through the brain stem on their way to the arms and legs, patients with a brain stem stroke may also develop paralysis in one or both sides of the body.

2.4.6 Other Effects of Stroke

Depression is very common amongst people who have had a stroke. It can be quite severe, affecting both the survivor and his/her family. A depressed person may refuse or neglect to take medications, may not be motivated to take part in physical rehabilitation or may be irritable with others. This in turn makes it difficult for those who wish to help, and tends to deprive the survivor of valuable social contacts that could help dispel the depression. In time the depression may lift gradually, but counselling and appropriate medication may be necessary. In the past, researchers speculated that some of the older anti-depressant drugs might interfere with a person's mental performance but recent studies suggest that anything that can effectively treat post-stroke depression, whether an old or new treatment, may also improve mental ability and enhance rehabilitation.

The way in which the person affected by stroke reacts to these changes will affect their personality, and may cause changes in control of emotions and behaviour. People affected by stroke may become confused, self-centred, uncooperative and irritable, and may have rapid changes in mood. Sudden laughing or crying for no apparent reason and difficulty controlling emotional responses also affects many stroke survivors. There may be little happiness or sadness involved, and the excessive emotional display will end as quickly as it started. They may not be able to adjust easily to anything new and may become anxious, annoyed or tearful over seemingly small matters.

2.5 PHYSIOTHERAPHY

Physiotherapy, or Physical Therapy, is a HealthCare profession concerned with the assessment, maintenance, and restoration of the physical function and performance of the body. Physiotherapy assists in regaining as much movement and function as possible. For stroke patient, treatment often focuses on sitting balance, standing balance, walking, using the affected arm / hand and managing any changes in muscle tone, pain or stiffness.

Neurological stroke physiotherapy can help:

- improve balance and walking
- increase ability to roll / move in bed / sit / stand
- reduce muscle spasms, pain and stiffness
- increase strength
- retrain normal patterns of movement
- increase affected arm and leg function

- increase energy levels
- increase independence and quality of life
- reduce the risk of falls

The traditional neurological stroke physiotherapy does help, the problem is that regular appointments with a physiotherapist have to be made for the program to work properly. This project as many other projects is to help stroke patients recover as soon as possible with less participation of the physiotherapist. (Physiotherapy Evidence Database 1999)

2.6 NEW EXPERIMENTAL THERAPY

There are currently other researches on improving the rehabilitation of stroke patients. As we will see some of this experimental therapy uses haptic technology (Haptic technology refers to technology which interfaces to the user via the sense of touch by applying forces, vibrations and/or motions to the user). Some of these of these researches and experimental therapies are discussed on this section.

2.6.1 Robot-Assisted Rehabilitation with Contraint-Induced Movement Therapy (CIMT)

For this research the performance of patients in a rehabilitation protocol that utilizes CIMT (Contraint-Induced Movement Therapy**)** methods where analyzed. The device is a modified commercial haptic joystick (figure 2.5), with the development of visual and haptic displays for the joystick.

The aim of the project was to compare robotics motor improvement measures with the clinical measures such as Fugl-Meyer (FM) and Log (MAL), measures such as trajectory and accuracy and smoothness of the movements. Another goal is to examine effects of different operating modes on patient's progress and to analyze changes in kinematic aspects of the movements during the recovery process. (http://mahilab.rice.edu/research)

The test was done such that in the figure 2.6 the patient's movements will be traced and shown as the green dot and they have to follow the blue as it goes around in the two red circles.

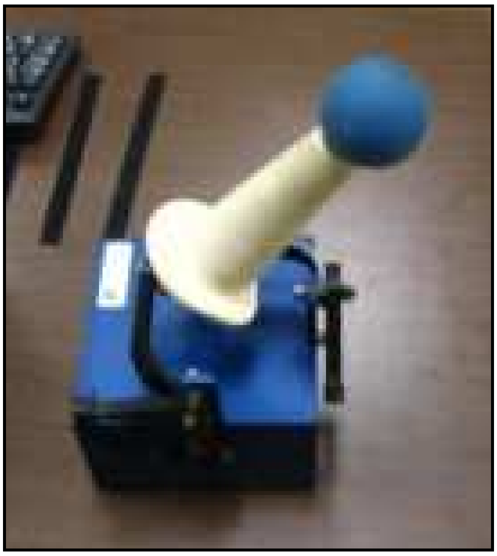


Figure 2-5: Haptic joystick (source: Mechatronics and Haptic Interfaces Lab 2008)

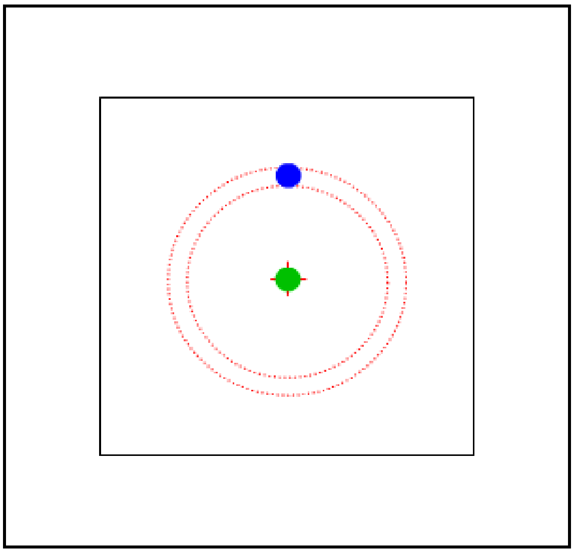


Figure 2-6: experiment on Haptic joystick (source: Mechatronics and Haptic Interfaces Lab 2008)

Robotic-Assisted Stroke Rehabilitation

This design of a robot was done for upper extremity stroke rehabilitation that is capable of applying assistive or resistive forces during reaching movements. Motions of the shoulder and elbow are controlled by the MIME (Mirror Image Movement Enabler) system, which utilize a Puma robot. Motions of the forearm and wrist are controlled by the RiceWrist, a parallel structure cable driven robot mounted on the end of the PUMA (seen in figure 2.7). Three operating modes are implemented on the combined MIME-RiceWrist system and a graphical user interface (GUI) is available that allows the number of repetitions, start and end points of the reaching movements, the mode and the mode parameters to be selected at each trial. Data are saved during the sessions for further analysis, such as patients' movement characteristics or evaluation of the motor recovery of the patient. (http://mahilab.rice.edu/)

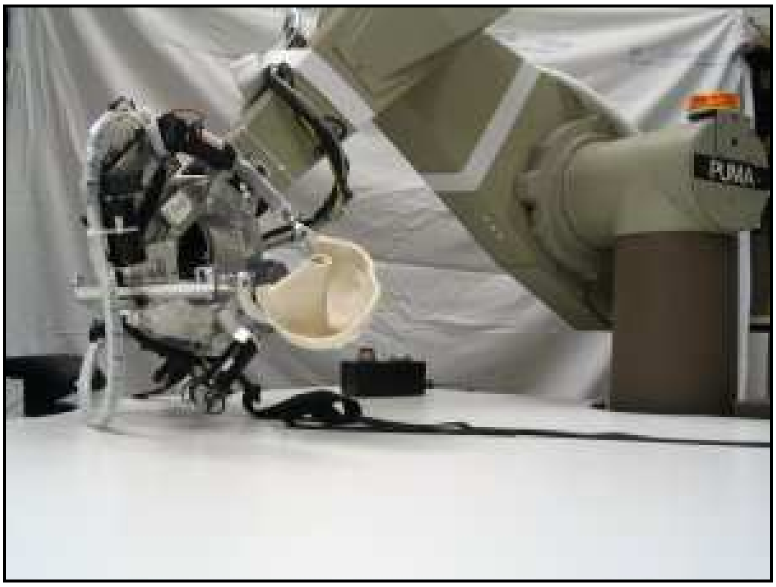


Figure 2-7: modified robot arm (PUMA) (source: Mechatronics and Haptic Interfaces Lab 2008)

Shared Control for Skill Transfer in Human-Robot Haptic Interactions

The primary goal of this research effort is to improve the effectiveness of skill transfer, rehabilitation, and collaboration via haptic devices. To do so, requirements will be formulated for shared control between humans and robots in haptic systems designed for training, rehabilitation, and collaboration.

In the first, the human act as the novice or patient, and the robot serves as the expert. Control schemes for the expert system (the haptic device seen in figure 2.8) will be designed and analyzed theoretically and experimentally. The second phase of the research effort will explore human-robot-human interfaces. At this stage, the focus is on expert-novice and therapist-patient teams, with a robotic system acting as the mediator between the two.

It was hypothesized that mediating robotic interfaces (either serving as the expert or placed between a human expert and the novice) can facilitate and improve the effectiveness of skill transfer and collaboration in expert-novice pairs as well as in therapist-patient rehabilitation interactions. (Mechatronics and Haptic Interfaces Lab 2008)

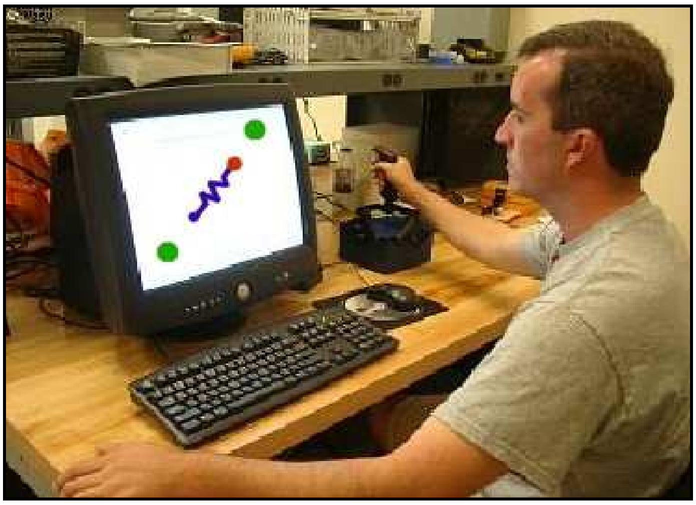


Figure 2-8:robotic interface (source: Mechatronics and Haptic Interfaces Lab 2008)

VRROOM

The Virtual Reality and Robotic Optical Operations Machine (VRROOM). VROOM is an integrated system combining virtual reality graphics environment, haptic robotic force feedback, and tracing of limb segments using a magnetic tracking system (Figure. 2.9). The system's primary component is the visual display system, the Personal Augmented Reality Immersive System (PARIS). (J. Patton, Y. Wei, & C. Scharver)

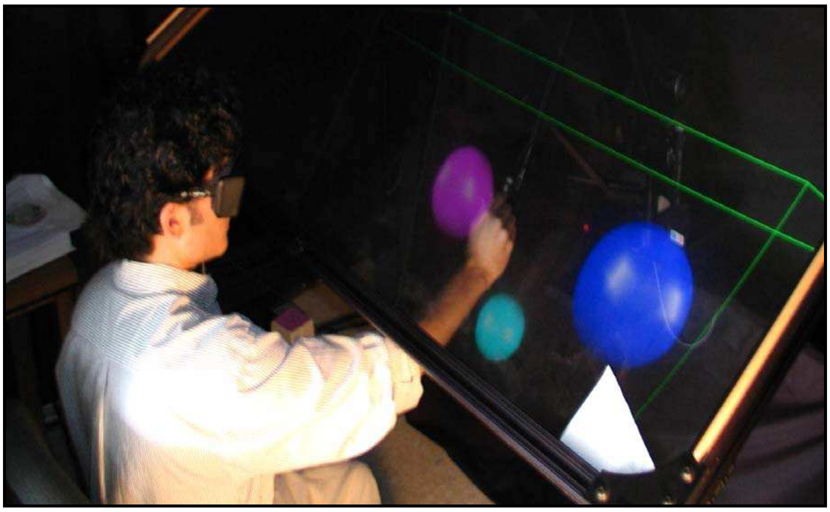


Figure 2-9(source: Mechatronics and Haptic Interfaces Lab 2008)

2.7 CONCLUSION

An extensive research on stroke, physiotherapy and stroke rehabilitation was successful. With an understanding of stroke and physiotherapy a more compatible design can be made. Having leant about stroke and its rehabilitation techniques, there is a need to learn more about joystick which is considered in the next chapter.

CHAPTER 3 : JOYSTICKS

3.1 INTRODUCTION

Joysticks are widely used in industries, in games and simulators. An extensive research needs to be done on joystick and how they work. With knowledge of how a joysticks works and an understanding in stroke a good design of the physiotherapeutic joystick can be made. In this chapter different types of joysticks will be analyzed.

3.2 TYPES OF JOYSTICKS AND OPERATIONS

In this section types of joysticks will be discussed, starting with simple “on and off” joysticks the kind being used in the arcade game (games such as Pac man). Then a discussion on the analog joystick and finally the force feedback joystick (howstuffworks 1998).

3.2.1 The Simplest Systems

The basic idea of a joystick is to translate the movement of a plastic stick into electronic information a computer can process. These joysticks do not provide incremental movement they can only turn on or off.

3.2.1.1 Contact buttons

The basic design of a joystick consist of a stick which has a rubber sheath base that is aligned to a circuit board which has printed wires making a simple electrical circuit as seen in figure 3.1. This circuit carry electricity from one point to the next. These circuits are connected to contact points taking the signal to the computer. When the joystick is in a neutral position all the buttons are not touched which means that their circuits are open, so moving in one direction will result in one of the buttons touching hence closing the circuit that signal will be sent to the computer and if the stick is moved in diagonal motion two buttons will be pressed.

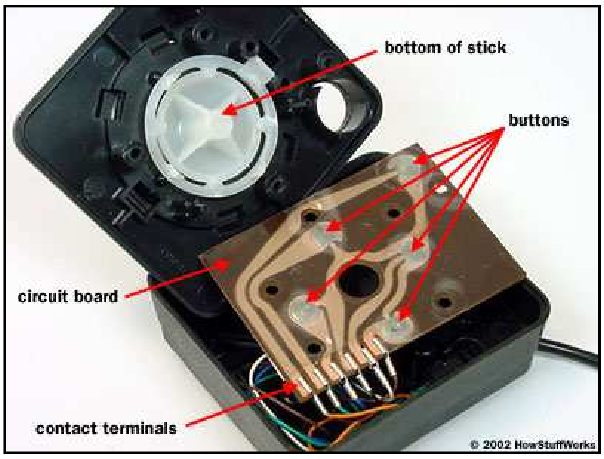


Figure 3-1: contact buttons (source: howstuffworks 1998)

3.2.1.2 Contact pins

This simple system also has contacts points connected to four switches (seen in figure 3.2) in such a way that when you push the stick forward the stick will push against the switch which will close and a signal sent to the computer, the computer will know the stick is pushed forward. These are fairly simple designs for a joystick even though the design might be ideal for some application and some games like Pac man or Tetris they are very limited in other applications as the joystick cannot tell if you are pushing on stick with a small or large force.

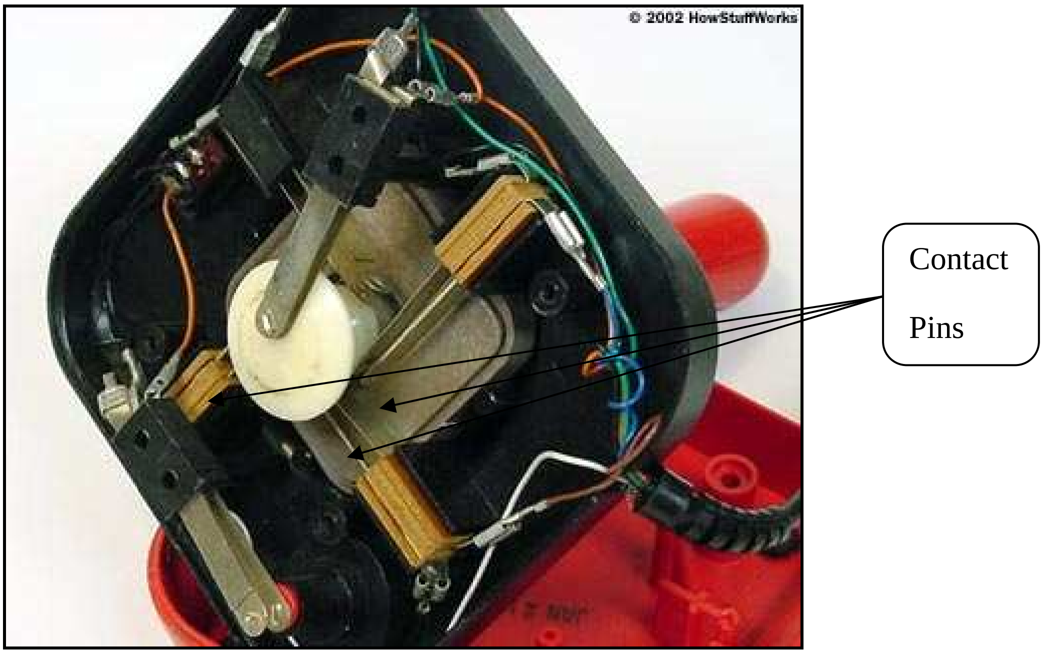


Figure 3-2: contact pin (source: howstuffworks 1998)

3.2.2 Conventional Analog joystick

In order to communicate a full range of motion the stick's position needs to be measured on two axes (X axis and Y axis). In this case the joystick uses potentiometers to measure its position and send the signal to the computer. Each potentiometer has a resister in a curved track and a movable contact arm. The computer will provide a current to the input terminal and through the contact arm. The computer will get the signal from the joystick, as the contact arm varies with the motions of the joystick. For example as the joystick is moved forward as far as it can go the contact arm on the potentiometer will move to the right which means the signal from the computer will have to move through the rest of the resistor track before going back to the computer though this the computer will be able to determine the position of the joystick and when the joystick is pulled back the signal from the computer will have to pass though a shorter section of the resistor before returning to the computer

With this joystick the signal is analog therefore a need to change it to a digital signal so that the computer can understand the signal. In a conventional system a printed circuit board within the computer handles the conversion from analog to digital. This uses the varying voltages from each potentiometer to charge a capacitor such that when the joystick is pushed as far forward as possible there will be more resistance hence the capacitor will take a long time to charge and when the stick is pulled back there will be less resistance hence the capacitor will take a short time to charge. By discharging the capacitor and timing how long it take to discharge, the converter can then determine the position of the potentiometer and therefore the joystick.

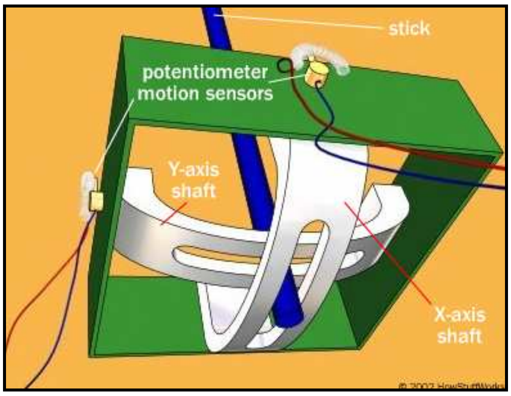


Figure 3-3: analog joystick (source: howstuffworks 1998)

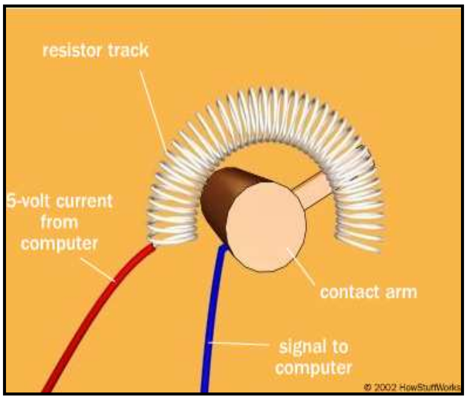


Figure 3-4: potentiometer (source: howstuffworks 1998)

3.2.3 Analog to digital conversion

The analog to digital conversion process is not accurate hence compromises the sensitivity of the joystick. The host computer has to dedicate a lot of power to determining the position of the stick this take a lot of power away from other operations. There are a few solutions to counteract these problems.

One of the solutions for these problems was to add a sensitive analog to digital converter chip in a specialized joystick adapter card in which the converter sends out digital information directly to the computer which improves the accuracy of the stick and reduces the workload on the host processor, these new joysticks models can use a USB (Universal Serial Bus) port which further improves the speed and reliability.

Another solution was skipping the potentiometer technology and optical sensors to read the stick movement digitally. In this technology two slotted wheels are connected to the shafts of the joystick each of these wheels is positioned between light emitting diodes and two photocells. When each light emitting diode shines though one of the slots it causes the photocell on the other side to produce a small amount of current, when the wheel moves slightly the slot moves and the light is blocked so the current is not been produced in this case so this repetitive shine and blockage of light though to the photocell causes the photo cell to produce a rapid pulses of current looking at the number of the pulses that the photocell has produced the computer can then know how far the stick has moved.

3.2.4 The force feedback joystick

The idea of a force feedback is to move in conjunction with on screen action to achieve this we have to add a microprocessor, two electrical motors and either a gear train or a belt system. Each of the two axes is connected to one motor through the belt system which will transmit the motion of the joystick to the motor. An electrical signal from the processor and the motion of the joystick will both move the motor, this way you can control the joystick to have assist-resist motions. The motor will have a position sensor which can either a potentiometer or an optical sensor, the sensor will detect any motion. Being able to move the motor with a signal form the microprocessor this gives us the opportunity to manipulate the signal in such a way that we can control the assist-resist. Using this we can manipulate the code such that the motors assist the patient follow the target and we manipulate the code to use the motor to resist the motion of the joystick which will be beneficial when the patient getting stronger. This will help in designing a device that can be used in almost all the stages of the rehabilitation.

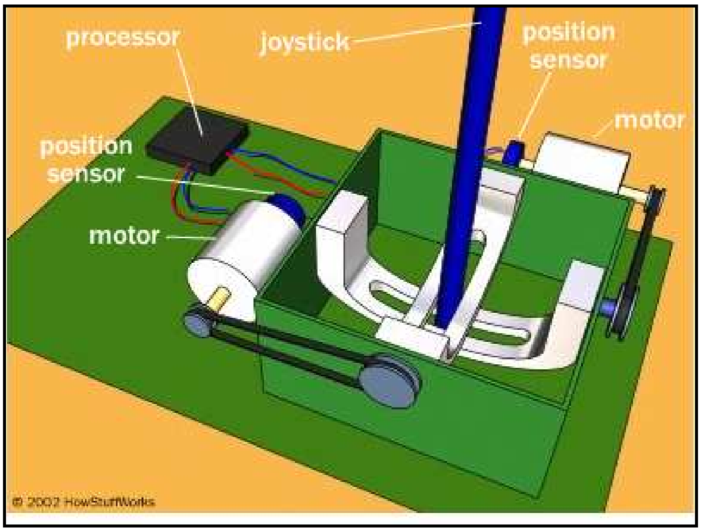


Figure 3-5: force feedback joystick **(**source: howstuffworks 1998**)**

3.3 CONCLUSION

In this chapter different types of joystick were analyzed. The simpler analog joysticks, the contact buttons and contact pins where discussed. The force feedback joystick was also analyzed. The information on how these joysticks work, gave me a few ideas on how I could design the therapeutic joystick. These ideas will be discussed in the next chapter.

CHAPTER 4 : CONCEPT DESIGNS

4.1 INTRODUCTION

With the knowledge on how joysticks work, a few initial ideas where generated using solid model software called ProEngineer (wild fire). Using software to model the joystick has advantages as I have the allowance of making mistakes without any waste of material and money, the measurements can be checked and proper calculations done before the actual product is built.

4.2 CONCEPT DESIGNS AND ANALYSIS

In this section some of the ideas that I had to start the development of the structure of the joystick are analyzed, with the final idea chosen.

4.2.1 Initial idea 1

The design is such that the joystick can move in two axes and can move diagonally between the axes, the motor will be inside the box with the shaft going through a hole so as to connect to the cam by a cable. The cam was to have a cable connected at both end as shown in figure 4.2. The cable was to be wrapped around the motor shaft as shown in figure 4.3

By pulling the stick back a chain of rotational movements is created which will move the motor in an anticlockwise direction, the angles at which the stick and arc rotates about the pivot would have been very important when it comes to designing the software for the joystick.

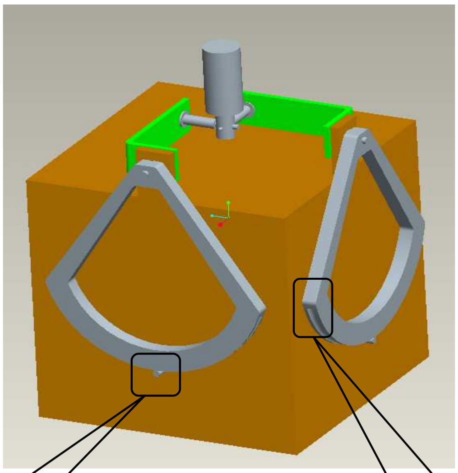


Figure 4-1: initial idea 1

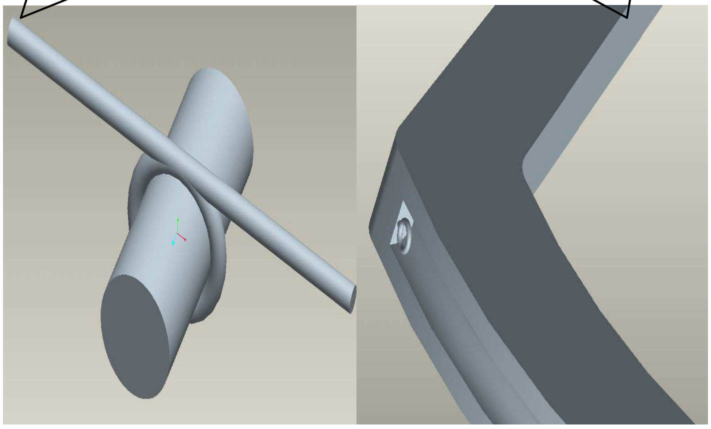


Figure 4-2: cable on motor shaft and cam pin

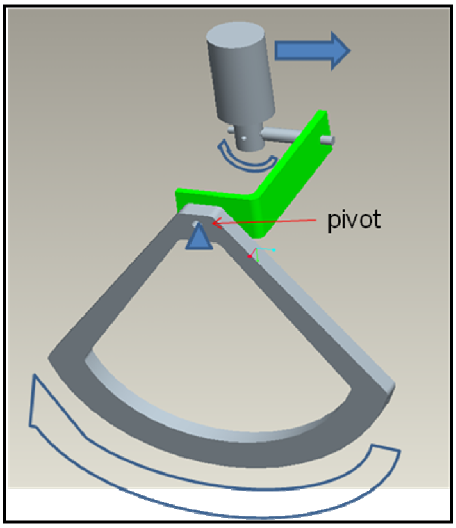


Figure 4-3**:** axel movement of the joystick

4.2.2 Initial idea 2

A steward platform was also considered in the initial ideas, a simplified version (on the right in figure 4.4) of the platform was designed using ProEng wildfire. Three actuators would have been used to control the position of the stick. This idea was considered to be too complex to be followed though, as the software development would have been harder using actuators.

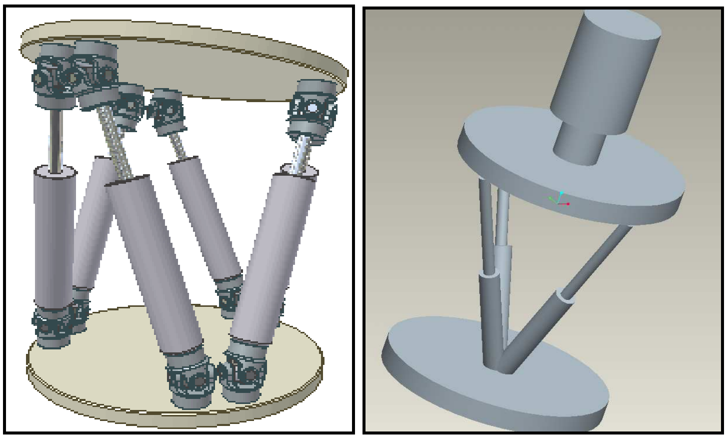


Figure 4-4: steward platform (source: Wikipedia, the free encyclopedia)

4.2.3 Initial idea 3

In the initial idea 3 the joystick was arranged such that the cam will be held between the two springs on the curved rod, as shown in figure 4.5. The plate has a rotating base which allows the joystick to move in all directions. The cam will be connected to the motor with a cable like in the first initial idea.

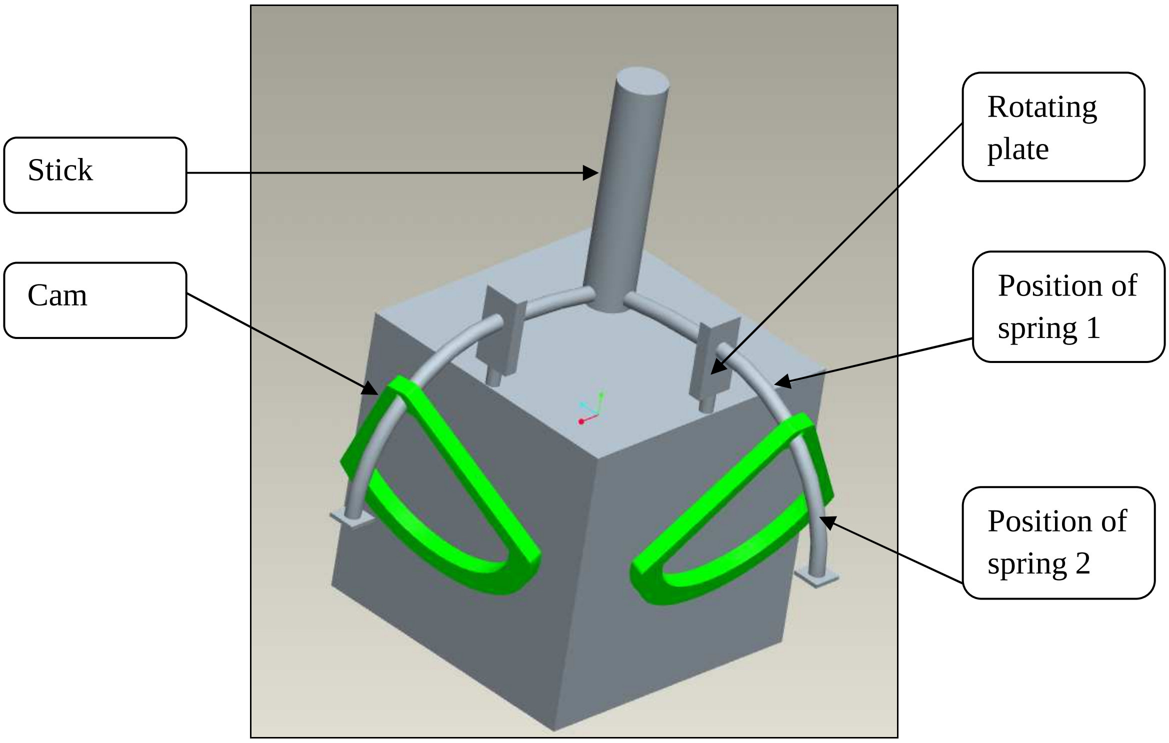


Figure 4-5: spring – cam joystick

4.3 FINAL DESIGN

From the initial ideas the cam idea was considered to be farfetched as the motor will have to rotate a many rotations just for the cam to move a few millimeters, so the cam was replaced by some pulley (pulleys seen in figure 4.6) to transmit the motion of the joystick to the motors and vice versa. The structure was changed from being just a box to a frame, so as to use standard steel beams and for easy machining. From these initial ideas a final design of the joystick was created. In the final design servo motors where used instead of DC motors.

A 25 x 25 x 3 steel section was used to create the joystick frame, 12 beams where welded together to create the frame. Three plates that hold the pulleys where welded to the box

The critical part of this design was to keep symmetry about the line A in figure 4.7, if there was a slight miscalculation the joystick will not be able to move freely diagonally. Another critical part of the design is to keep a 90 degree angle between the two pins holding the stick indicated on the diagram by the lines B and C

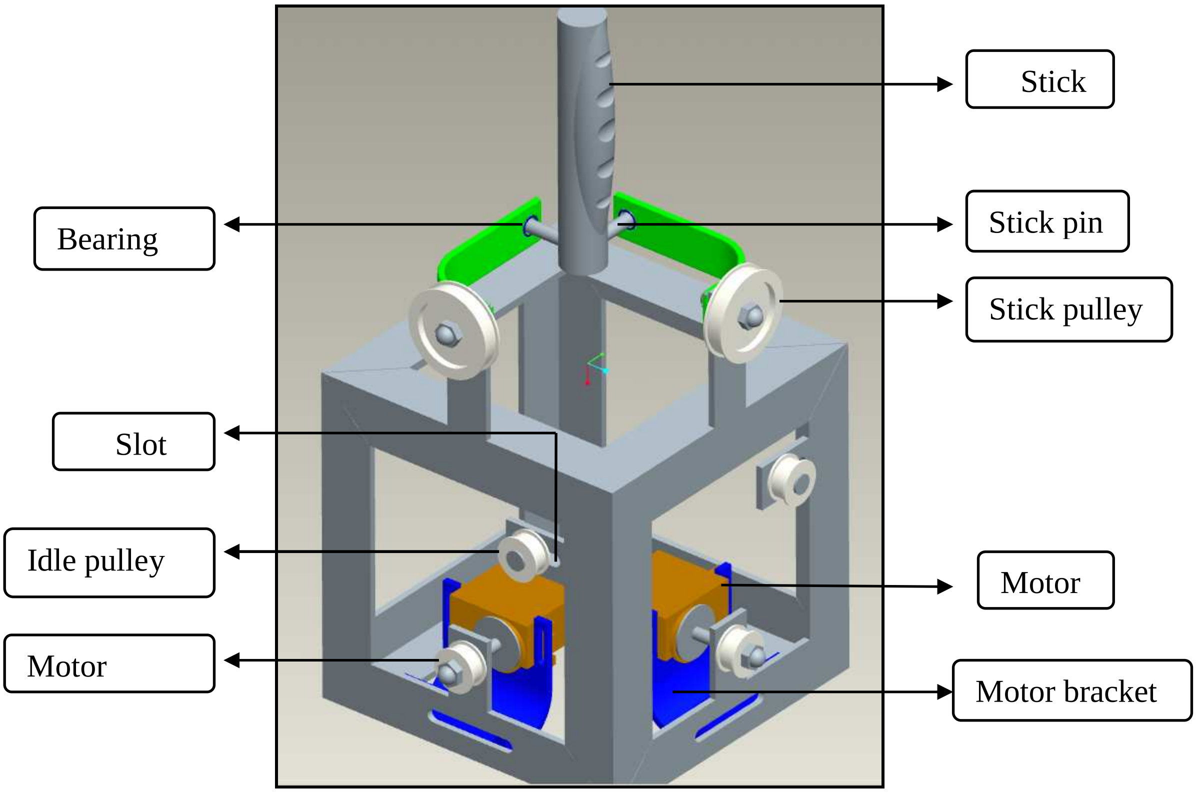


Figure 4-6: final design

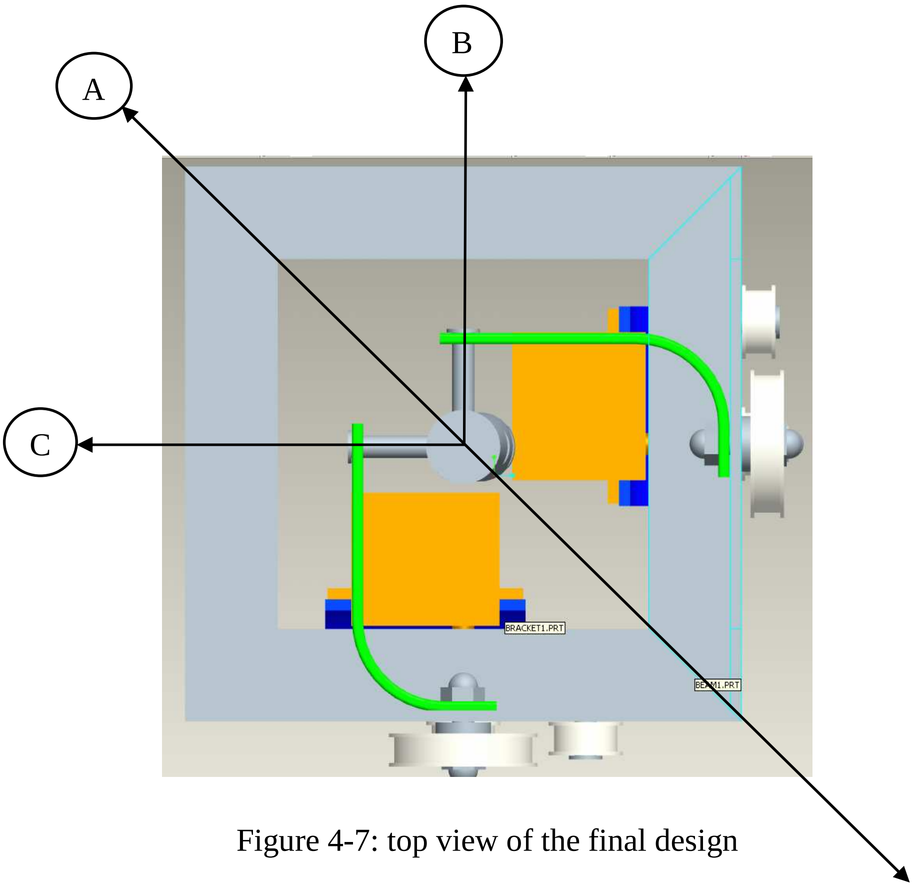


Figure 4-7: top view of the final design

4.4 CONCLUSION

Through process of elimination a suitable design was selected. The final design was created from all the initial ideas they were analyzed and some concepts from the ideas were used to make the final design.

CHAPTER 5 : FINAL DESIGN AND PROTOTYPE

5.1 INTRODUCTION

The final design is discussed in detail, in this chapter the structure, the hardware and the software are designed and analyzed. Calculations of length of the cord were determined in this chapter.

5.2 STRUCTURE

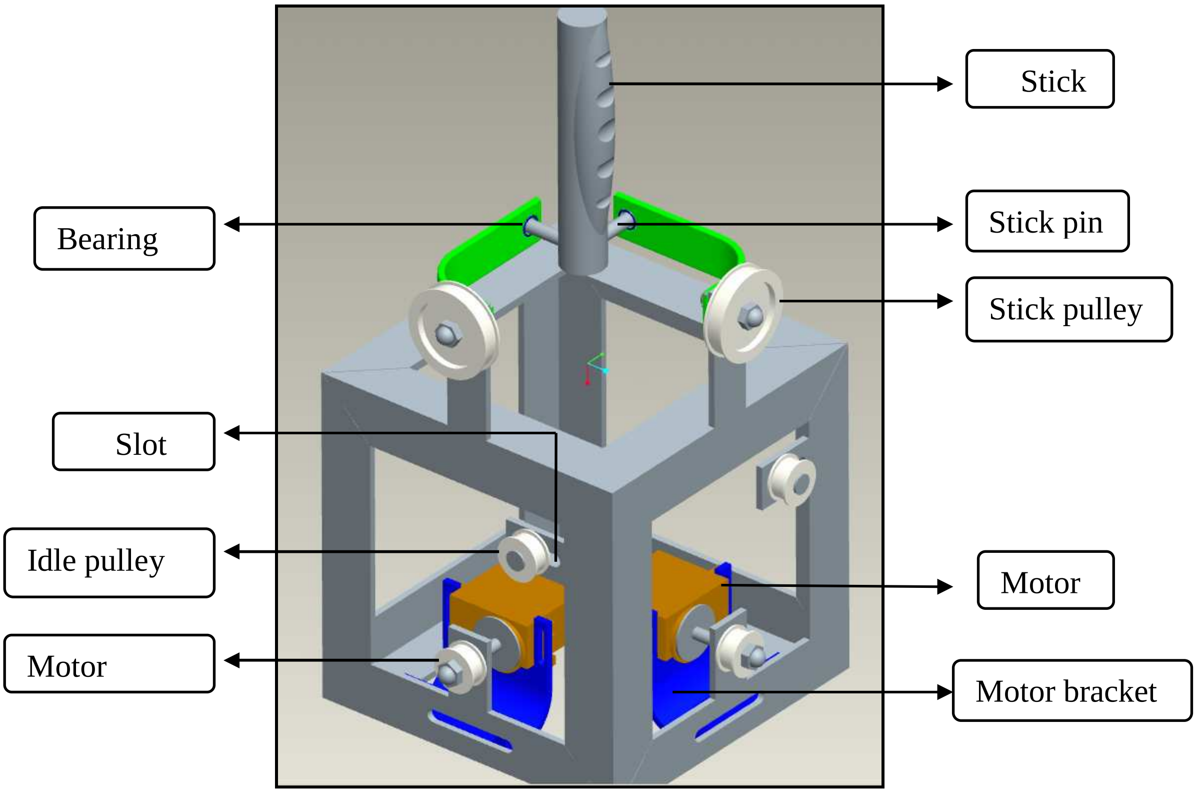
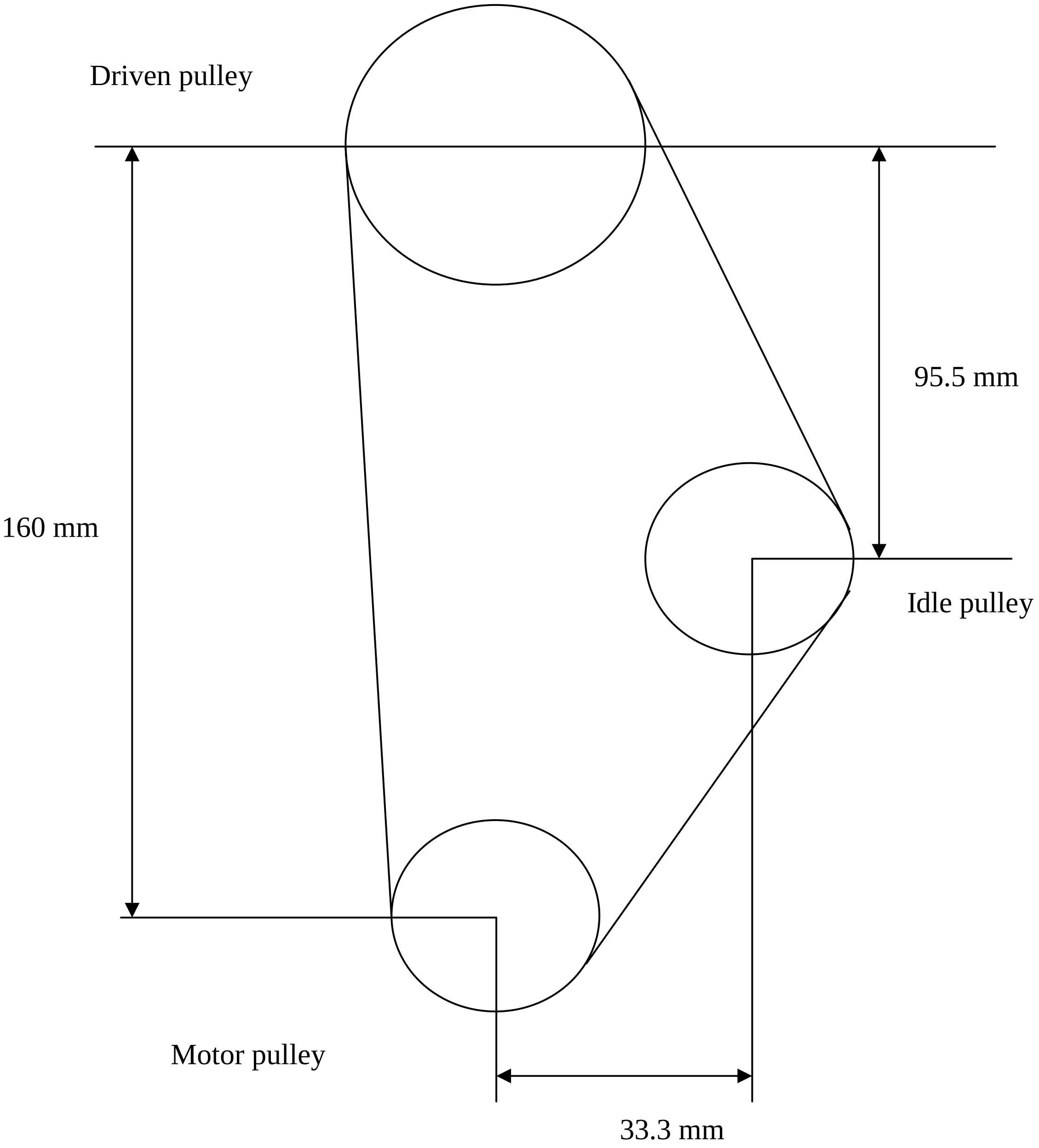


Figure 5-1: final design

5.2.1 Pulley positions and length of cord

This section shows the position of the pulleys on the final design



From the calculations found in the Appendices the shortest possible length of the cord was calculated to be 385mm

5.3 PROTOTYPE

Pictures of the prototype were taken. The prototype works well with minor machining error as the stick could not move the whole 360 degrees the design intended but it did had the overall achieved what the design. More pictures of the prototype are in appendices.

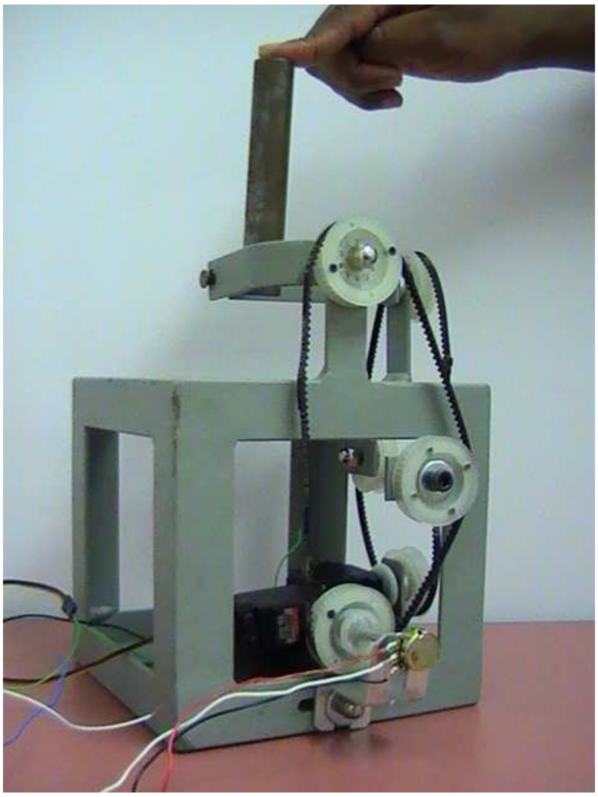


Figure 5-2: PROTOTYPE

5.4 ELECTRONICS

In this section the electronics that were used will be discussed. The motor selection and the development board used are analyzed.

5.4.1 MOTOR SELECTION

The servo motor selected for the joystick control because of the feedback, the servo motor was compare to two other types of motor to make sure it was the right motor for the system. The DC motor and the stepper motor are the two motors chosen to compare with servo motor.

5.4.2 Servo motor VS DC motor

A DC motor (seen in figure 5.3) has only two wires connections, these wires are positive and negative terminal and they are used to provide the motor with drive power. All controls for the motor is done through these terminals, as these terminals are provided with power the motor starts spinning around until the power is removed then the motor will stop. Most DC motors are very fast revolving about 32000 revolutions per minute. (http://www.batteryspace.com/index.asp?PageAction=VIEWCATS&Category=522).

The DC motor uses PWM (pulse width modulation) to control power level and hence its speed, this technique works in such a way that the motor power level is controlled by turning the power off and on. A good example of PWM is riding a bicycle as the rider paddle the bicycle it is given energy to move forward and as the rider stops paddling the momentum will carry the bicycle forward but it will slow down due to wind resistance, friction on the bicycle and friction against the road, then paddling again to keep the same speed. A duty cycle of 100% means that the rider is paddling all the time and a duty cycle of 50% means that the rider is only paddling half of the time. When using an 8 bit micro controller the duty cycle will be 0 – 255, pulsing with a 127 will give a duty cycle of 50%. Running the motor at a duty cycle of 50% the motor will be running at 50% its speed.

The servo motor is different it is actually an assembly of four things: a normal DC motor, a gear reduction unit, a position sensing device and a control unit. The servo works in such a way that it receives a signal representing a desired position of the motor shaft and apply power to the DC motor until the shaft turns to that position, with the position sensing device determining the position of the shaft it will know which way to turn the motor to go to the target position. The servo has three wires instead of two(DC motor), the positive terminal , negative terminal and the control signal, in case of the servo motor the power is supplied constantly and the control signal is the one which is pulse width modulated.

Pulse width modulation is used differently in case of the servo motor, the motor is pulsed from 0.6 ms to 2.4 ms (for Hitec hs 311) , the motor move 180 degrees only with 0.6 ms at 0 degrees, 0.9 ms at 90 degrees and 2.4 ms at 180 degrees

Figure 5-3: DC motor (source: Fasco 2008)

5.4.3 Servo motor VS stepper motor

Stepper motors (seen in figure 5.4) are permanent magnetic motors that step one increment each time the computer gives its control electronics one pulse. They do not have position feedback that is if they are run within their limits, and they inherently hold their position when stopped. Stepper motor systems are often open loop which means the controller only tells the motors how many steps to move and how fast to move but does not have a way of determining if the motor arrived at the correct position, this may lead to some error, if the motor is part of a bigger system even a very small error might lead to a cumulative error, which may in turn lead to huge accidents.

Servos are generally standard DC or brushless motors with an encoder feedback loop. The computer reads the position of the motor and applies the power in the right direction to move the motor shaft to the right position. (http://www.netmotion.com/)

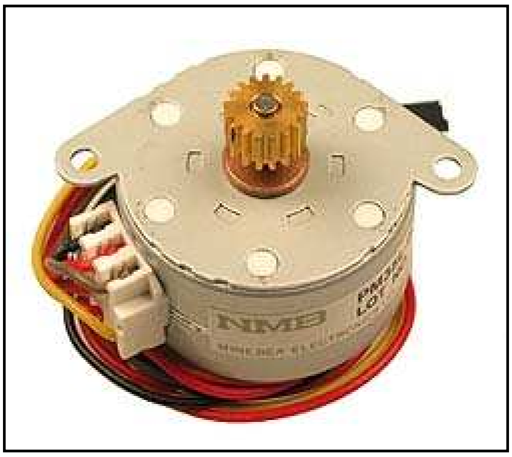


Figure 5-4: stepper motor (source: Futurlec 2008)

Table 2: Servo VS stepper table (source: NetMotion)

Motion characteristics	Servo motor	Stepper motor
High torque, low Speed	Servos are able to achieve high Torque at low speed.	Steppers can also do continuous duties at High torque and low speeds.
High toque and High speed	Continuous duty applications Requiring High torque and high speed. A DC servo motor can deliver greater Continuous shaft power at high speeds Compared to steppers. High speed up to 12000 rpm are possible.AC servos motors can handle Higher current surges compared to DC servo	Stepper motors Become too bulky at high torques And high speeds of More than 2000 rpm.

	Motors. There are a lots stronger servos as Compared to either DC servo motor and DC Stepper motor.	
Short, rapid Repetitive moves	Servo motors are better when dealing with High dynamic requirements.	Stepper would be a More economical Choice if the Equipments are modest
Positioning Applications	Servo motors can handle efficiently when load is Mostly inertia instead of friction. The ability to Overdrive servo motor in intermittent duty Allows a smaller motor to be used. A servo is better when positioning is critical in micron level.	Stepper motors can Be used when Positioning is not Very critical.
Applications in Hazardous Environments	A brushless servo motor is preferred in this application.	A step motor is preferred in this Application.
Low speed , high Smoothness	A DC servo motor is preferred at low speeds with high smoothness.	A micro stepping Motor is preferred In this application.
Control method	A servo motor is a closed loop controlled motor.	A stepper motor Is an open loop Controlled motor.

5.4.4 Servo motor

A standard HiTEC HS-311 (figure 5.5) servo motor was selected for this project due to its position feedback. Servos are extremely popular with robot, RC plane, and RC boat builders. Most servo motors can rotate about 90 to 180 degrees. Some rotate through a full 360 degrees or more. However, servos are unable to continually rotate, meaning they can't be used for driving wheels (unless modified) but their precision positioning makes them ideal for robot arms and legs, rack and pinion steering, and sensor scanners to name a few. Since servos are fully self contained, the velocity and angle control loops are very easy to implement, while prices remain very affordable. (http://www3.towerhobbies.com/cgi-bin/wti0001p?&I=LXDEL5)

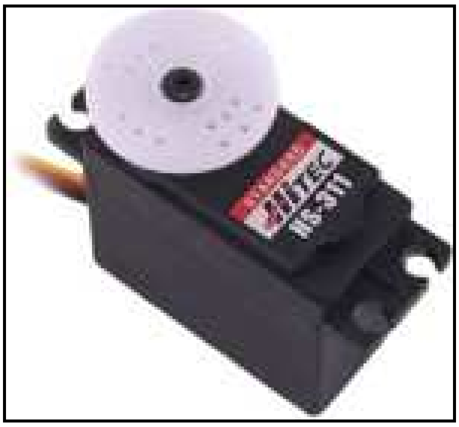


Figure 5-5: servo motor (source: servocity 2008)

5.4.4.1 Servo Wiring

All servos have three wires for the hitec hs 311 we have:
Black is for ground.
Red is for power (~4.8-6V).
Yellow is the signal wire (3-5V).

Servo Voltage (Red and Black)

Servos can operate under a range of voltages. Typical operation is from 4.8V to 6V. There are a few micro sized servos that can operate at less, and now a few Hitec servos that operate at much more. The reason for this standard range is because most microcontrollers and RC receivers operate near this voltage.

5.4.4.2 Signal Wire (Yellow)

While the black and red wires provide power to the motor, the signal wire is what you use to command the servo. The general concept is to simply send an ordinary logic square wave to your servo at a specific wave length, and your servo goes to a particular angle (or velocity if your servo is modified).

The standard **time vs. angle**:

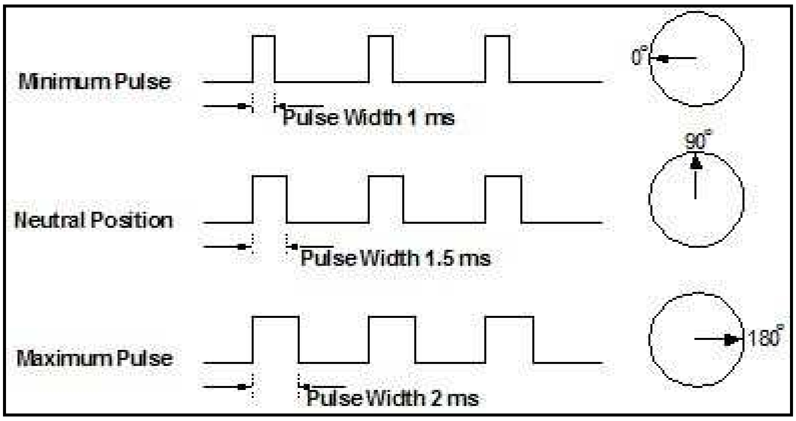


Figure 5-6: time vs. angle

The diagram shows the range of the servo motor pulse, the diagram show a general range it will be different for many different types of servo motors. The range of pulse for the Hitec Hs 311 is 0.6 ms to 2.4 ms, with 1.5 ms as the centre.

5.4.4.3 Gear Types

More expensive servos come with metal gears for higher torque and longer life, followed by karbonite and then nylon gears for the cheapest.

5.4.4.4 Nylon Gears

Nylon gears are most common in servos. They are extremely smooth with little or no wear factors. They are also very lightweight, but lack in durability and strength.

5.4.4.5 Karbonite Gears

Karbonite gears are relatively new to the market. They offer almost 5 times the strength of nylon gears and also better wear resistance. Cycle times of well over 300,000 have been observed with these gears with virtually no wear. Servos with these gears are more expensive but what you get in durability is more than equaled.

5.4.4.6 Metal Gears

Metal gears have been around for some time now. Although the heaviest and having the highest wear rate of all gear types, they offer unparalleled strength. With a metal output shaft, side-loads can be much greater. Unfortunately, due to wear, metal gears will eventually develop slight play in the gear-train. Accuracy will slowly be lost.

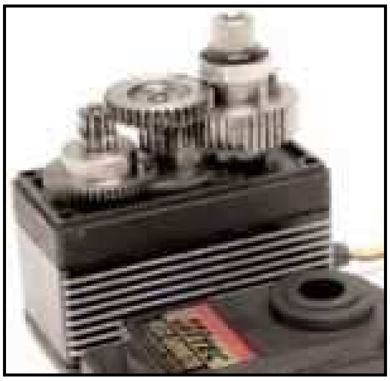


Figure 5-7: servo metal gears

5.4.4.7 Efficiency and Noise

Due to noise and control circuitry requirements, servos are less efficient than DC motors uncontrolled. To begin with, the control circuitry typically drains 5-8mA just on idle. Secondly, noise can more than triple current draw during a holding position (not moving), and almost double current during rotation.

Noise is often a major source of servo inefficiency and therefore should be avoided. Sometimes a servo will vibrate. This is because the servo is rapidly jumping between two different angles due to interference. The interference may be caused the signal wire which acts as a long antenna, capable of accepting unwanted foreign signals and sending them straight to your servo as a command. A common interference source is usually from other nearby servos and/or servo wiring. To prevent this problem the servo signal wire needs to be kept short. (http://www.rcuniverse.com/)

5.4.4.8 Digital Servos VS Analog Servos

The difference between an analog and digital servo

Digital servos, at the user end, are controlled no differently than analog servos. The difference is in how the servo motor is controlled via the circuit board (amplifier). The motor of an analog servo receives a signal from the amplifier 30 times a second or at 30Hz. This signal allows the amplifier to update the motor position. Digital servos use a high frequency amplifier that updates the servo motor position 300 times a second or at 300Hz. By updating the motor position more often, the digital servo can deliver full torque from the beginning of movement and increases the holding power of the servo. The quick refresh also allows the digital servo to have a tighter deadband.

With the exception of a higher cost, there are only advantages for digital servos over analog servos.

The digital micro processor is 10 times faster than an analog servo. This results in a much quicker response from the beginning with the servo developing all the rated torque 1 degree off of the centre point. This faster response results in higher starting currents.

Digital servos can be programmed for direction of rotation, centre and end points, failsafe option, speed, and dead bandwidth adjustment. This is great for matching sets of servos for deadband width, centre and end points in giant scale aircraft applications.

The standing torque of a digital servo is 3 times that of its analog counterpart. This means digital servos are typically smaller and have more torque. Digital servos are also modifiable.

Figure 5-8: AC servo motor (source: baldor)

5.4.5 HARDWARE

5.4.6 The design and construction of the hardware

An initial hardware design was to use an ATMEGA8535 microcontroller and build the circuit board to control the motor. To achieve this a few components where needed:

Table 1: Components

QTY	NAME AND DESCRIPTION
1	Breadboard
1	ATMEGA8535 microcontroller chip
1	Max232 (RS driver Tx/Rx chip)
4	10 uF capacitors (for the max232)
1	7805(5V output dc voltage regulator)
18	480 ohm Resistors (0.25 watt)
5	LEDs
2	Capacitors 33 pF for AVR crystal clock source
1	7.3728 MHz crystal oscillator

The development of the circuit was abandoned as using a development board would save time, a comparison was taken between two development boards, the olimex AVR - MT - 128 and the Arduino Diecimila.

5.4.7 The olimex AVR - MT - 128 development board for ATMEGA128 microcontroller.

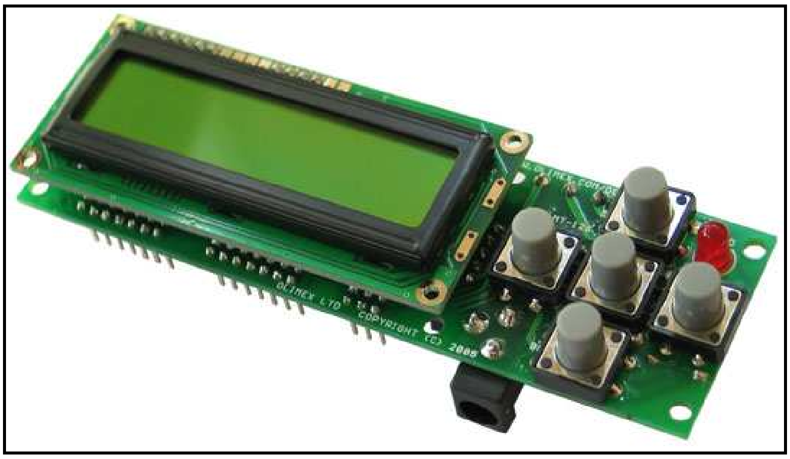


Figure 5-9: AVR - MT – 128 (source: olimex 1999)

Features:

- MCU: **ATMega128-16AI** with128K Bytes Program Flash, 4K Bytes data EEPROM, 4K Bytes RAM
- JTAG connector for in-circuit programming and debugging with AVR-JTAG
- ICSP 5x2 (10) pin STKxxx compatible connector for in-circuit programming with AVR-PG1B or AVR-PG2B
- RS232 connector with TTL levels
- RS232 interface circuit with Tx, Rx signals
- RS232 DB9 female connector
- Dallas touch button port
- Frequency input
- LCD 16x2 display

- LED status
- five buttons
- Buzzer
- power supply circuit +5V, 78L05 with plug-in power jack and diode bridge
- 32 768 Hz oscillator crystal
- 16 Mhz crystal oscillator
- power supply filtering capacitor
- RESET supervisor IC ZM33064
- RELAY with 10A/250VAC NO and NC contacts with screw terminals
- extension headers for unused in the schematic ports available for external connection
- PCB: FR-4, 1.5 mm (0,062"), green solder mask, white silkscreen component print
- four mounting holes 3.3 mm (0.13")
- Dimensions: 120x38 mm (4.7x1.5")

5.4.8 The Arduino Diecimila development board for ATMEGE168 microcontroller

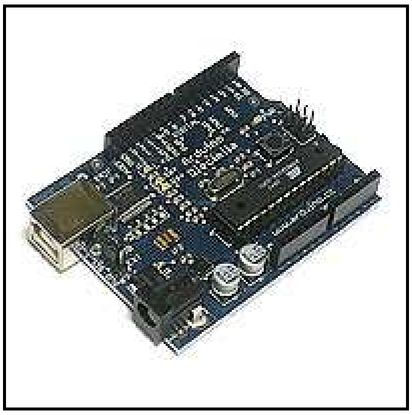


Figure 5-10**:** Arduino Diecimila (source: Arduino 2008)

The Arduino was selected over the olimex AVR-MT-128 for its simplicity and a fairly simpler software environment. The Arduino was chosen as it was fairly simpler to use than the olimex AVR - MT – 128, another helpful future was that the arduino uses the USB port for both serial communication and to transmit the program on to the microcontroller.

Futures:

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (of which 2 KB used by bootloader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz

5.4.9 Arduino Diecimila

5.4.9.1 Overview

The Arduino Diecimila is a microcontroller board based on the ATmega168. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

"Diecimila" means 10,000 in Italian and was named thusly to mark the facts that over 10,000 Arduino boards have been made. The Diecimila is the latest in a series of USB Arduino boards**.**

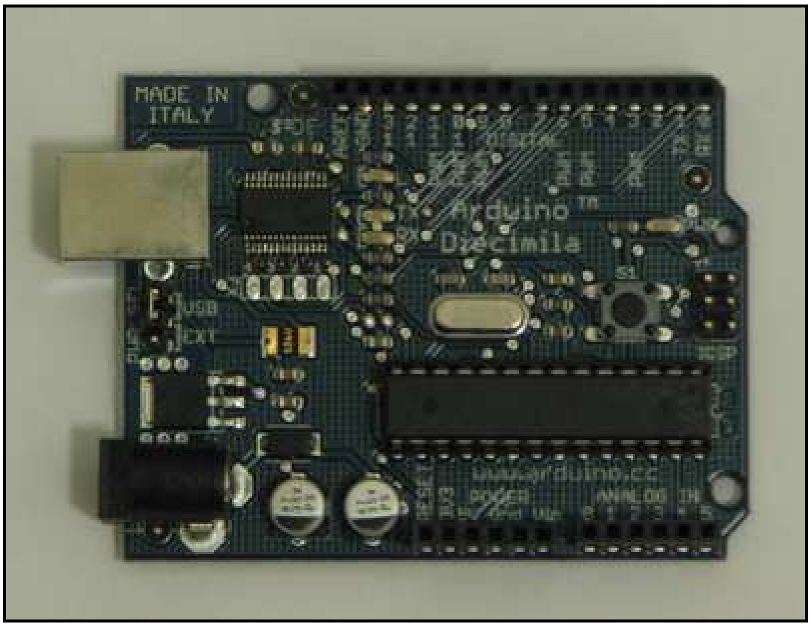


Figure 5-11: Arduino Diecimila (source: Arduino 2008)

5.4.9.2 Power

The Arduino Diecimila can be powered via the USB connection or with an external power supply. The power source is selected by the PWR_SEL jumper: to power the board from the USB connection, it has to be placed on the two pins closest to the USB connector, for an external power supply, the two pins closest to the external power jack.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. A low dropout regulator provides improved energy efficiency.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board FTDI chip. Maximum current draw is 50 mA.
- **GND.** Ground pins.

5.4.9.3 Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader). It has 1 KB of SRAM and 512 bytes of EEPROM.

5.4.9.4 Input and Output

Each of the 14 digital pins on the Diecimila can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Diecimila has 6 analog inputs, each of which provides 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and some low-level code. Additionally, some pins have specialized functionality:

- **I^{2}C: 4 (SDA) and 5 (SCL).** Support I^{2}C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference ().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

5.4.9.5 Communication

The Arduino Diecimila has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

A Software Serial library allows for serial communication on any of the Diecimila's digital pins. The ATmega168 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

5.4.9.6 Programming

The Arduino Diecimila can be programmed with the Arduino software which is based on C programming language. The ATmega168 on the Arduino Diecimila comes preburned with a bootloader that allows new code to be uploaded to it without the use of an external hardware programmer. It communicates using the original STK500 protocol.

5.4.9.7 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Diecimila is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. Version 0009 of the Arduino software uses this capability to allow the uploading of the code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Diecimila is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Diecimila. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

5.4.9.8 USB Over current Protection

The Arduino Diecimila has a resettable polyfuse that protects the computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

5.4.9.9 Physical Characteristics

The maximum length and width of the Diecimila PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. The Arduino is easily connected to the servo motor demonstrated in figure 5.12 (Arduino 2008).

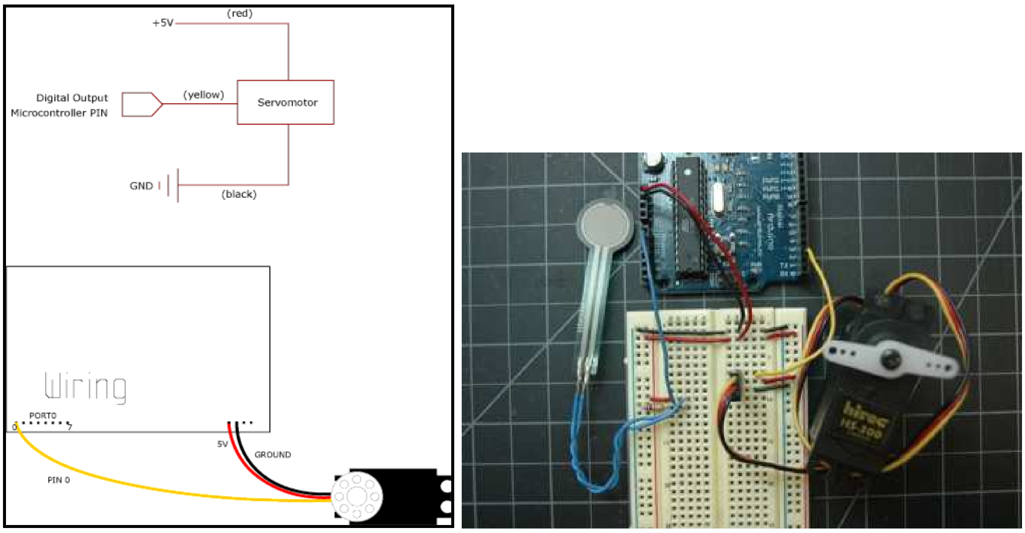


Figure 5-12: servo motor connection to arduino (source: electronics-lab 2002)

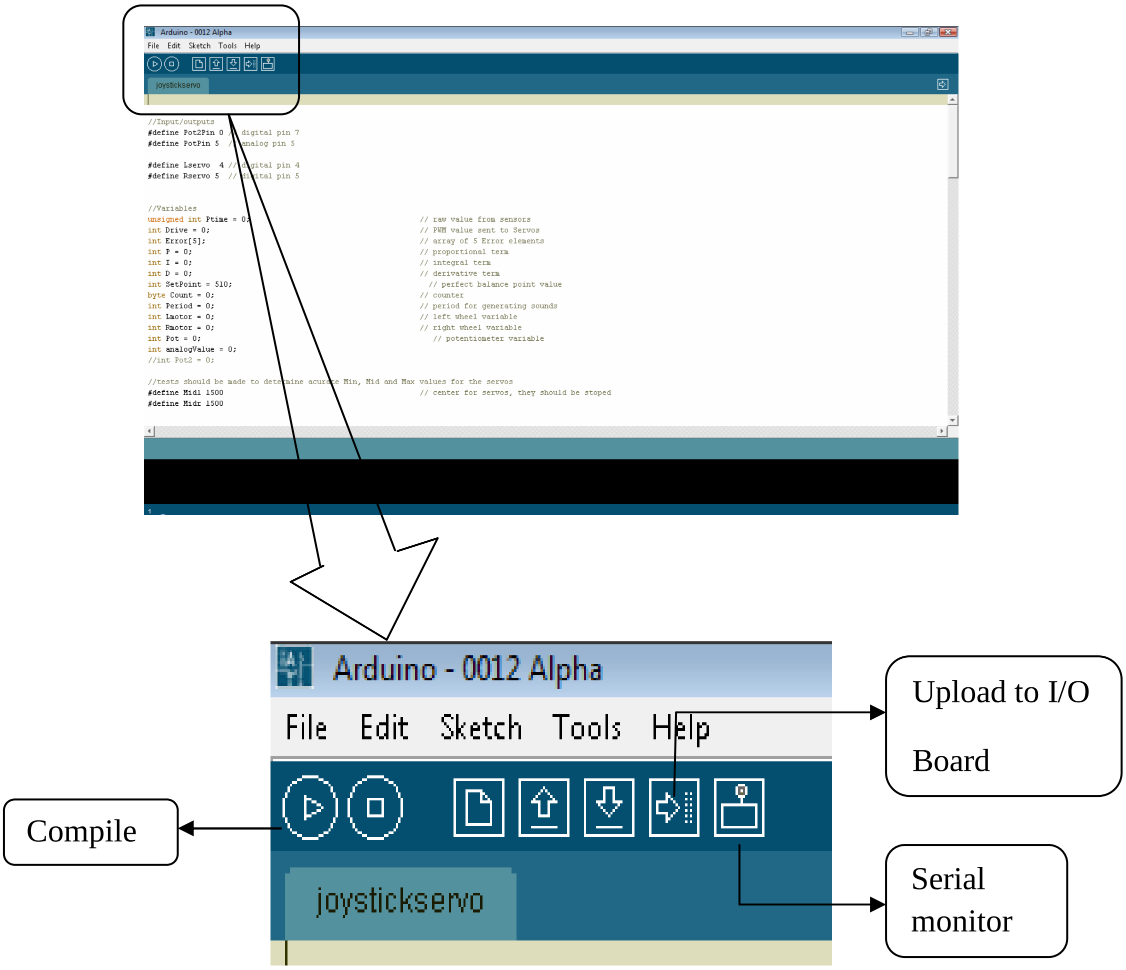
5.5 SOFTWARE

Initially Bascom was the choice for the software development, this language as well as C programming could have been used with the initial idea of the hardware design and with the olimex AVR - MT – 128, as the Arduino was the choice of the hardware C programming language for Arduino was selected. Arduino has its own environment, the open-source

Arduino environment makes it easy to write code and upload it to the I/O board. The environment is written in Java and based on Processing, avr - gcc, and other open source software.

5.5.1 Arduino Environment

The Arduino environment is fairly simple to use as seen from figure 5. Below



```
//Input/outputs
#define Pot2Pin 0 // digital pin 7
#define PotPin 5  // analog pin 5

#define Lservo  4 // digital pin 4
#define Rservo 5  // digital pin 5


//Variables
unsigned int Ptime = 0;                                  // raw value from sensors
int Drive = 0;                                           // PWM value sent to Servos
int Error[5];                                            // array of 5 Error elements
int P = 0;                                               // proportional term
int I = 0;                                               // integral term
int D = 0;                                               // derivative term
int SetPoint = 510;                                        // perfect balance point value
byte Count = 0;                                          // counter
int Period = 0;                                          // period for generating sounds
int Lmotor = 0;                                          // left wheel variable
int Rmotor = 0;                                          // right wheel variable
int Pot = 0;                                                // potentiometer variable
int analogValue = 0;
//int Pot2 = 0;

//tests should be made to determine acurate Min, Mid and Max values for the servos
#define Midl 1500                                        // center for servos, they should be stoped
#define Midr 1500
```

Figure 5-13: Arduino environment

5.6 CONCLUSION

In this chapter the final design was discussed in detail. A choice of motor was done with a discussion about the difference between the motors and a detailed discussing on the servo motor which was selected as the motor to be used in the project. Arduino board and software were chosen for the project.

CHAPTER 6 : SYSTEM EVALUATION AND ANALYSIS OF RESULTS

6.1 INTRODUCTION

In this chapter the final design was tested and results are recorded and then analyzed

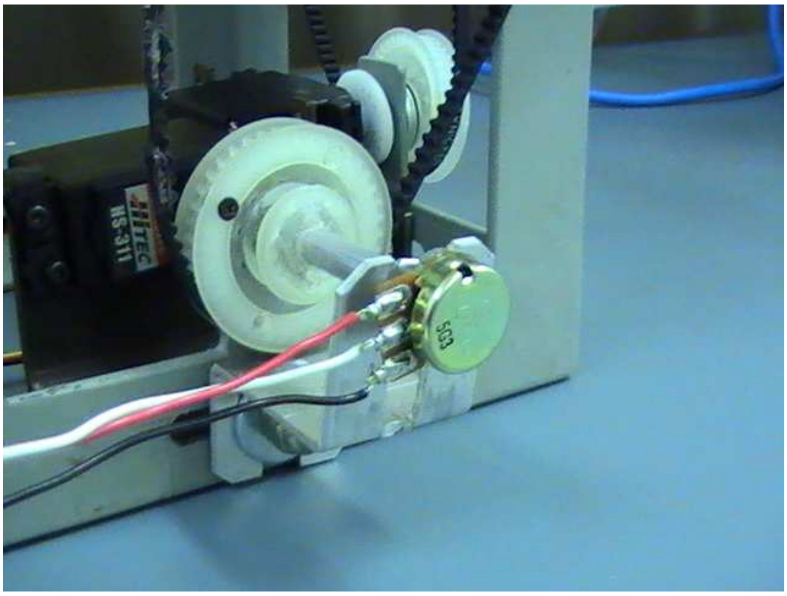
6.2 TESTING AND ANALYSIS

The testing was done by down loading the program in to the arduino board then running the program in the hyper terminal (figure 6.2) to view the results of the test. The results are then saved as a text file and then imported into matlab for plotting. One motor is left to run freely while the program is applied to the other motor. The target position is set and the joystick is pushed forward with some side movements on the way to observe whether the motor will act as to correct the joystick if it ventures off course.

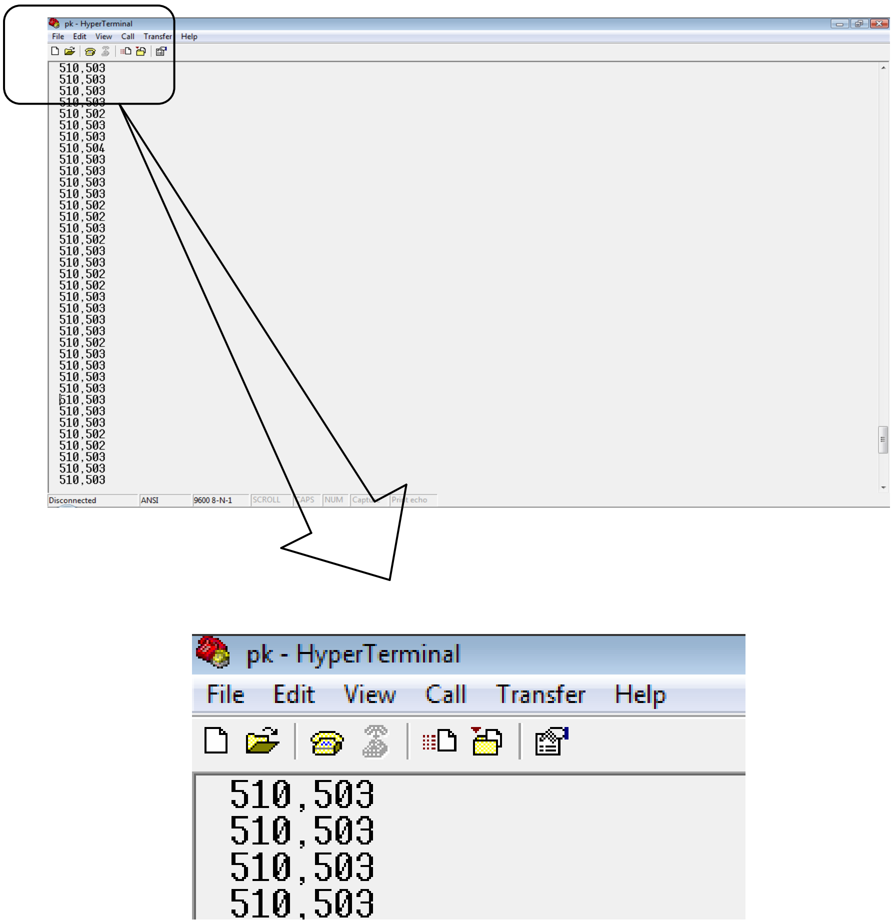
The potentiometer shaft is mounted on the motor shaft with a bracket holding it in place as we can see in figure 6.1, so if the motor moves the potentiometer shaft moves as well. The target position can be set using the potentiometer:

```
int SetPoint = 510;                    // target position
```

Then applying a PID loop to the motor setting the error as target position- potentiometer position. The motor drive towards the target position and stay at the target position until instructed otherwise. Should at any point the joystick be moved away from the target position the error on the PID will increase and as to correct the error the motor will move to the right position.



6-1: potentiometer position



```
510,503
510,503
510,503
510,503
510,502
510,503
510,503
510,504
510,503
510,503
510,503
510,503
510,502
510,502
510,503
510,502
510,503
510,503
510,502
510,502
510,503
510,503
510,503
510,503
510,502
510,503
510,503
510,503
510,503
510,503
510,503
510,503
510,502
510,502
510,503
510,503
510,503
```

Disconnected | ANSI | 9600 8-N-1

```
510,503
510,503
510,503
510,503
```

6-2: hyper terminal showing the results from serial communication

Decoupling Analog Inputs

Finding that the readings from your analog inputs are inconsistent, I tried to decouple your input circuit. Decoupling means smoothing out the dips and spikes going into the circuit from the rest of your microcontroller circuit. To do this, I placed a 0.1microfarad capacitor from voltage to ground shown in figure 6.3

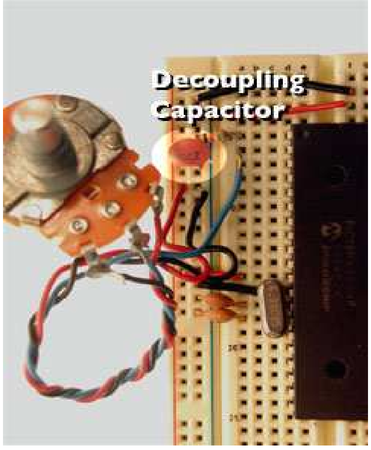


Figure 6-3: decoupling with a capacitor (source: Code, circuits, & construction, 2008)

INITIAL TEST RESULTS

Several tests were run using different PID parameters as shown below but most of the tests were inconsistent. With about seven out of ten test showing the expected behavior on the motor. There was a lot of noise on the signal, to try to avoid this I tried to decouple the analog inputs seen in figure 6.3.

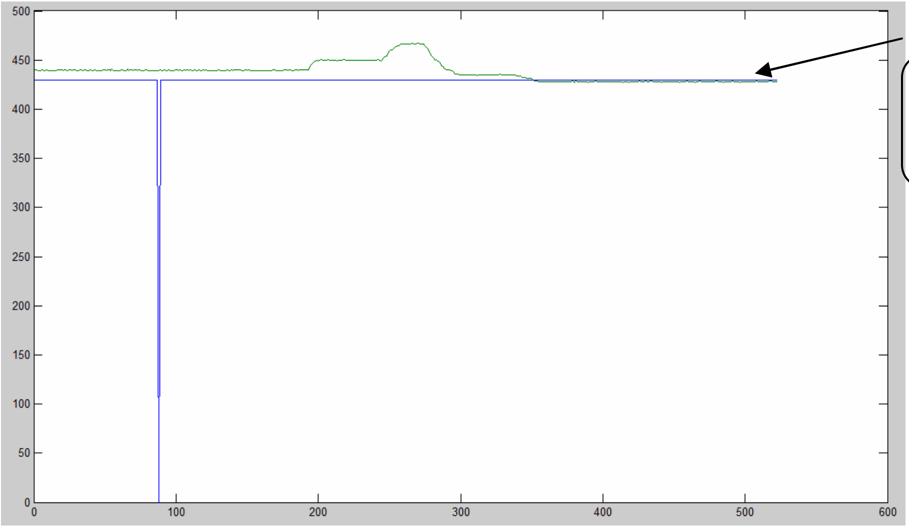
The test with different PID parameters of:

The system response shows some inconsistency as it is stable at about 440 for 2 seconds, but as I move the stick off target the system acts as not to bring the system to 440 but, it takes the stick to 430 which is the target position. This is seen in figure 6.4.

Kp = 15

Kd = 5

Ki = 10



TARGET POSITION

POSITION

TIME

Figure 6-4: results

The second test was better than the first test as the system showed some consistency, it starts at the target position as expected then with some outside disturbances the system acted as to correct the disturbance and returned to the target position. This is shown in figure 6.5

Kp = 25

Kd = 10

Ki = 5

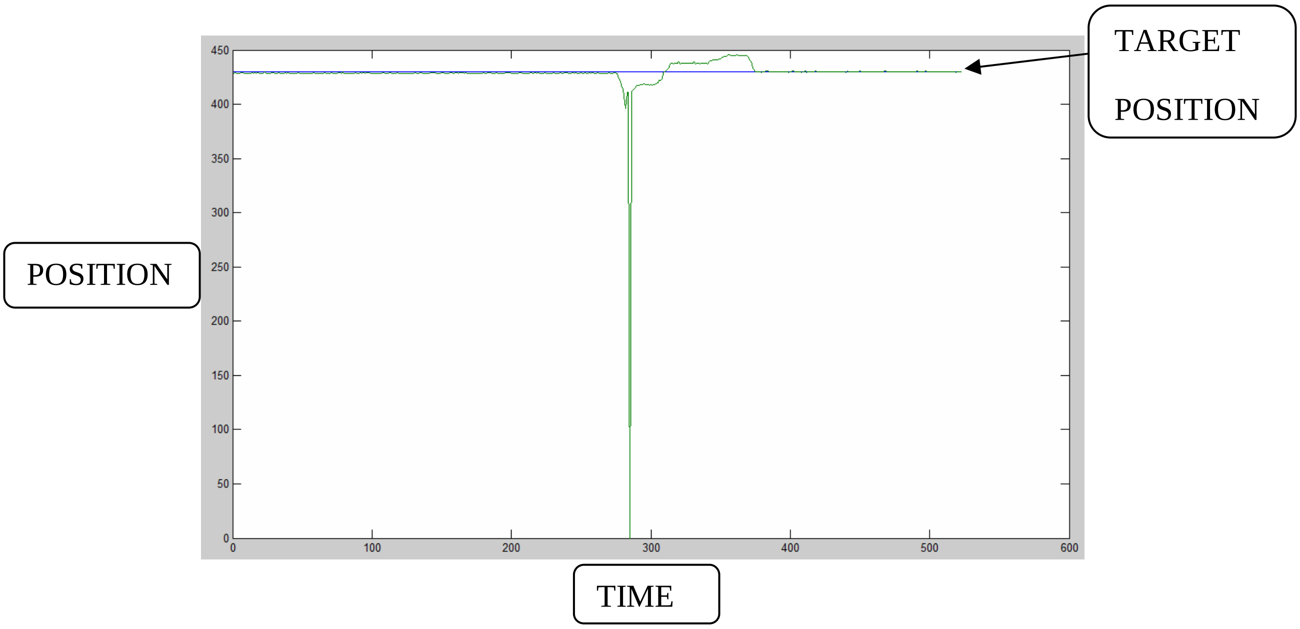


Figure 6-5: results

With the scale of the plot window reduce, it shows the inconsistencies that I mentioned. The system appears to stabilize at different positions. This is shown in figure 6.6

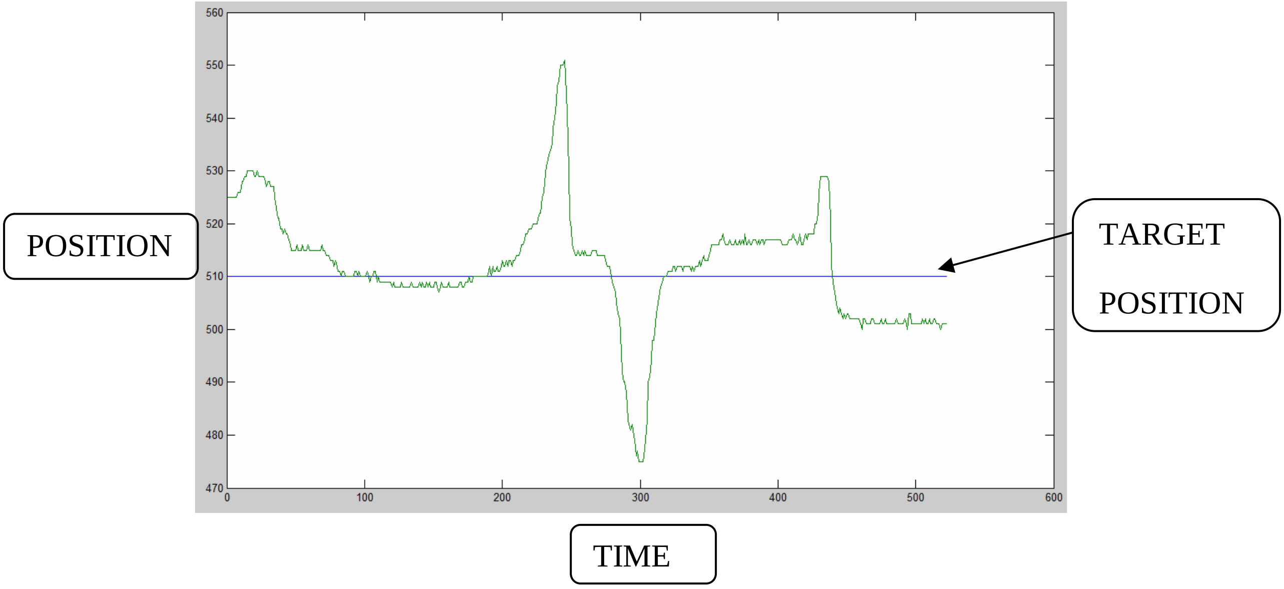


Figure 6-6: results

6.3 CONCLUSION

The results of this project where not as consistent as I hoped they will be, the problem of the results was traced to the program for the system. The system program was revised a few time as the and still the idea system response could not be achieved, due to time constraints the ideal solution for the problem could not be achieved, this was also due to limited knowledge of the C programming language.

CHAPTER 7 : CONCLUSION AND FUTURE WORK

7.1 INTRODUCTION

In this chapter the conclusion of the project is discussed and the project procedure is summarized with an analysis on the project specification and finally the future work to be done on this project.

7.2 SUMMARY OF PROJECT

A stroke is the sudden death of a portion of the brain cells due to a lack of oxygen. A stroke occurs when blood flow to the brain is damage resulting in abnormal function of brain. It causes by blockage or rupture of an artery to the brain. This illness can happen in many different ways, with the main two as: Ischaemic and Haemorrhagic stroke. Stroke can affect the body in many different was, for example when the left side of the brain is affected some parts of the right side of the body may lose their functionality, and when the right side of the brain is affected some parts of the left side of the body may lose functionality.

There are currently many researches on stroke therapy, the technology involved in these researches are haptic technology which is the technology that interfaces to the user via the sense of touch by applying forces, vibrations and/or motions to the user. This technology is often used with visual objects to stimulate rehabilitation. The use of joysticks as force feedback joysticks is common nowadays, though we haven’t seen more being used with haptic technology for the rehabilitation of stroke patient, the information was very helpful in this project.

Different joystick and how they work were studied, which enabled the creation of the concept design which lead to the final design of the force feedback joystick for the rehabilitation of stroke patients. The ProEngineer (wildfire) was the software used to create the joystick models. The modeling software has an advantage as the prototype could be built after the proper analysis of the model has been done. A servo motor was chosen for the project because it has feedback. The servo motor was compared with two other motor namely: DC motor and stepper motor.

Instead of developing the control circuitry a decision was made to use a development board to control the joystick. Two development boards the Arduino Deicimila and the olimex AVR - MT - 128 were compared, and the Arduino was chosen as it was fairly simpler to use than the olimex AVR - MT – 128, another helpful future was that the arduino uses the USB port for both serial communication and to transmit the program on to the microcontroller. A code in Arduino was created to run the system, and finally tests were made to determine the performance of the joystick.

7.3 FUTURE WORK

Future work to be done in this project:

- Make the system resistive as well as assistive.
- Modify the code so as to have the system moving diagonally as well as north-south and west-east.
- Create an interactive game to make the rehabilitation more efficient.

REFERENCES

Baldor 2001,servo motors, viewed July 2008, <http://www.baldor.com/products/servo_motors.asp>

Bergquist WH, Mclean B & Kobylinski BA,1994, *Stroke Survivors*, Jossey-Bass Inc.

Canada Physiotherapy Association, Canada, viewed March 2008 <http://www.physiotherapy.ca/PublicUploads/222460Strokeinfo.pdf>

Code, circuits, & construction, 2008 viewed august 2008

<http://www.tigoe.net/pcomp/code/input-output/analog-input>

David Cuartielles, David A. Mellis, 2008, viewed June 2008, <http://www.arduino.cc/>

Electronics-lab, 2002, viewed august 2008

<http://www.electronics-lab.com/blog/?cat=9>

Fasco 2008, DC motors, viewed June 2008,

<http://www.fasco.com/dcmotor1.asp>

Futurlec 2008, viewed July 2008

<http://www.futurlec.com.au/StepperMotors.jsp>

Margaret Johnstone, 1987, *The stroke patient,* 3rd edn, Longman Group UK Limited.

Mechatronics and Haptic Interfaces Lab 2008,

<http://mahilab.rice.edu/research>

Michael Aronson, MD, July 2005. What Is a Stroke, New York, NY, 2005 WebMD Inc, viewed march 2008 <http://www.ghi.com/topic.aspx?page=644>

MyDr 2001, *STROKE: SIGNS, SYMPTOMS AND TREATMENT*, Chandos Street, St Leonards, NSW, Australia, Reviewed: 25/6/2006, viewed March 2008, <http://www.mydr.com.au/default.asp?article=2440>

National Clinical Guidelines for Stroke (UK) Royal College of Physicians 2004, viewed march 2008, <http://www.rcplondon.ac.uk/pubs/books/stroke/>

National Stroke Foundation, 2008, stroke, Bourke Street, Melbourne, Victoria, Australia viewed April 2008, <http://www.strokefoundation.com.au/>

NetMotion 2008, viewed July 2008

<http://www.netmotion.com/>

Olimex, development boards and tools, 1999, Slavjanska St, Plovdiv, BULGARIA viewed April 2008, <http://www.olimex.com/dev/index.html>

Physiotherapy Evidence Database, 1999, viewed may 2008, <http://www.pedro.fhs.usyd.edu.au/>

Richard Robinson, stroke, answers.com, viewed March 2008, <http://www.answers.com/topic/stroke>

Robert G. Robinson,1998, *The Clinical Neuropsychiatry of stroke*, Cambride University Press.

Ro-Bot-X, 2008, Build a (simple) balancing robot, viewed April 2008 <http://www.societyofrobots.com/member_tutorials/node/185>

Rose FC & Capildeo R, 1981, *Stroke: the facts*, Oxford University Press.

Sarang Ghude, 2007, Design a PID Controller with Missing Packets

in a Networked Servo-System, University of Southern Queensland eprints.

Servocity, 2008, HS-311 Standard, viewed June 2008,

<http://www.servocity.com/html/hs-311_standard.html>

Stroke Association (TSA), *Stroke Association (TSA)*. Retrieved Oct 28, 2008, viewed March 2008, <http://www.stroke.org.uk>

Stroke center, 2008, St. Louis, Missouri, USA, viewed April 2008 <http://www.strokecenter.org/>

Stroke Effects, 2008, viewed April 2008, <http://www.americanheart.org/presenter.jhtml?identifier=4761>

Stroke - the after effects, 2007, State of Victoria, viewed may 2008, <http://www.betterhealth.vic.gov.au/bhcv2/bhcarticles.nsf/pages/Stroke_the_after_effects?OpenDocument>

Striking back at stroke, 2008, Toorak Road, Camberwell, Victoria, Australia viewed March 2008, <http://www.physiotherapy.asn.au/index.php/physiotherapy-a-you/older-people/striking-back-at-stroke>

Tom Harris, howstuffworks, 1998, viewed match 2008, <http://electronics.howstuffworks.com/joystick.htm>

Wikipedia, the free encyclopedia, Stewart platform, viewed match 2008, <http://en.wikipedia.org/wiki/Stewart_platform>

Yatsu FM, Crotta JC &Pettigrew LC, 1995, *stroke 100 MAXIMS,* volume 3, Edward Arnold, London.

APPENDIX A: PROJECT SPECIFICATION

University of Southern Queensland

FACULTY OF ENGINEERING AND SURVEYING

ENG 4111/4112 Research Project
PROJECT SPECIFICATION

FOR: **PAKO MARUPING**

TOPIC: THE DESIGN AND CONSTRUCTION OF A LARGE SCALE-FORCE FEEDBACK FOR USE IN PHYSIOTHERAPY

SUPERVISOR: **Selvan Pather**

ENROLMENT: ENG 4111- S1, D, 2008
ENG 4112- S2, D, 2008

PROJECT AIM: This project aims are to design and construct a large scale force feedback joystick which will be used for stroke patients in Physiotherapy arm rehabilitation. The joystick will initially assist weak patients and then resist as patients gain more control.

PROGRAMME: Issue A, 25th March 2008

1. Research information on rehabilitations of stroke patients.
2. Research on current technology used in Physiotherapy.
3. Design a large scale force feedback joystick which will have north-south and east-west motion.
4. Construction of the joystick.
5. Design the software for the motors which will be used in the motions of the joystick
6. Interface the joystick with software and test

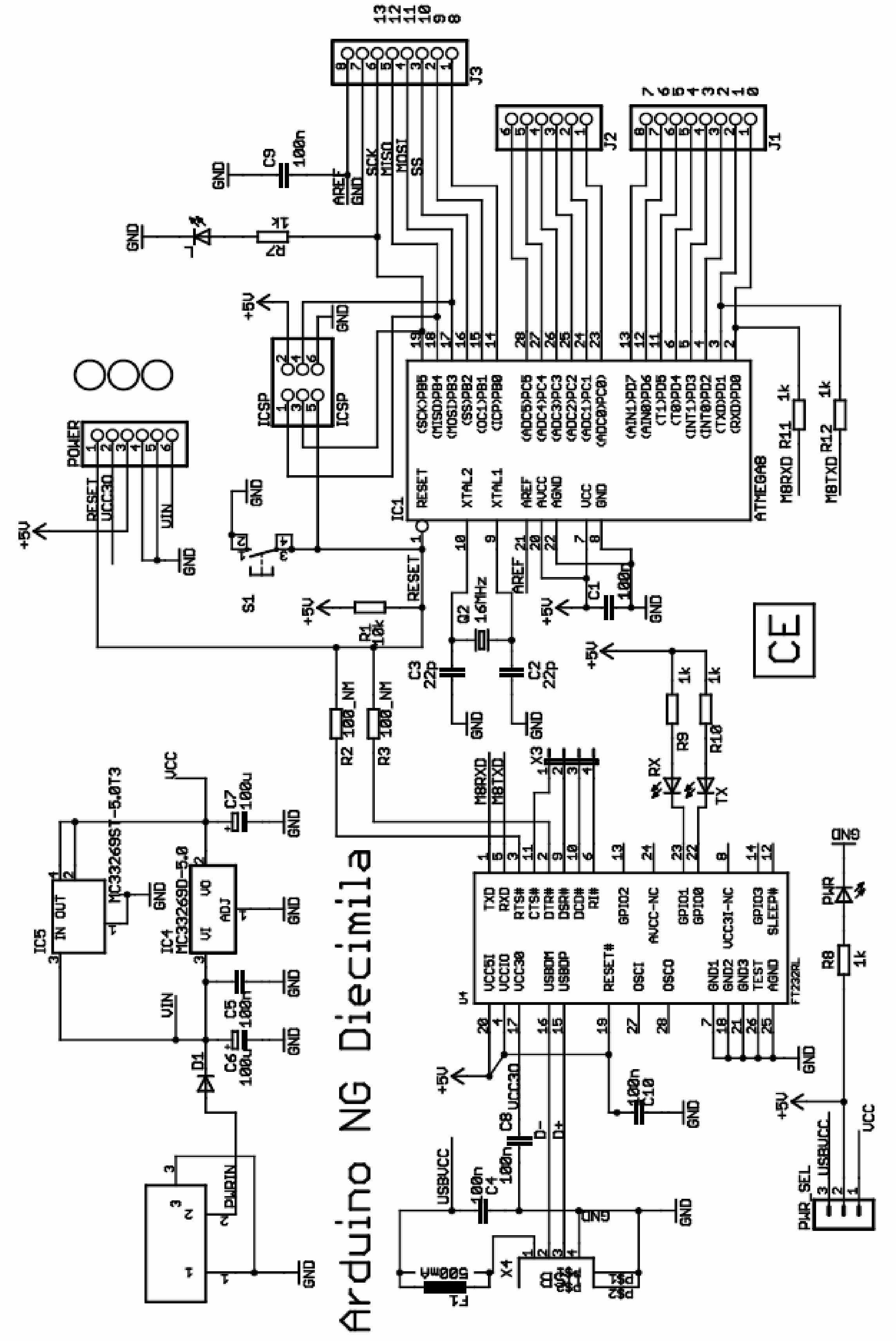
As time permits

7. Include rotational motion in the motions of the joystick
8. Human trials

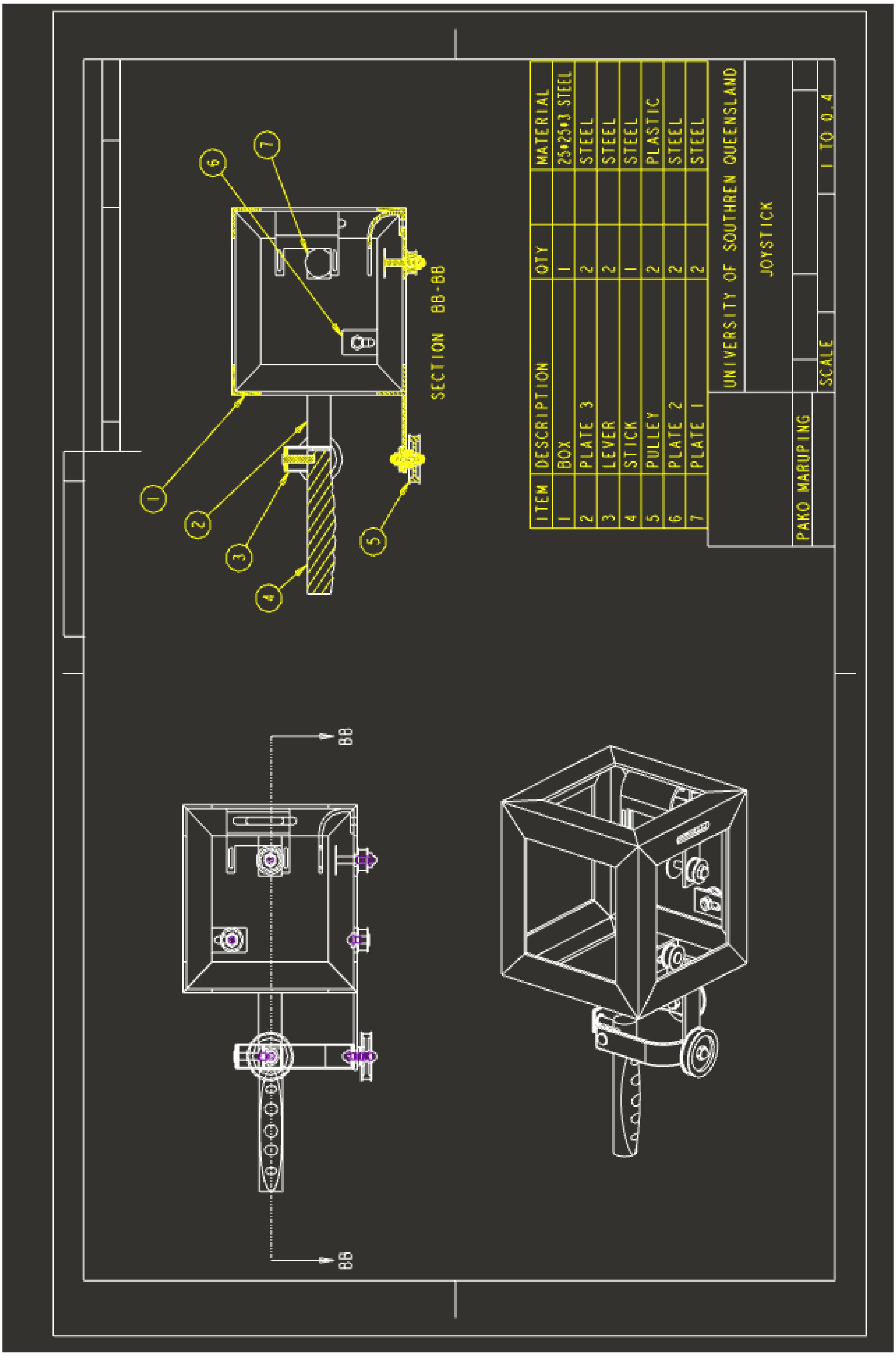
AGREED:

X______________________ __ /__ /__ X______________________ __ /__ /__
STUDENT SUPERVISOR

APPENDIX B: ARDUINO SCHEMATIC



APPENDIX C: DESIGN DRAWINGS



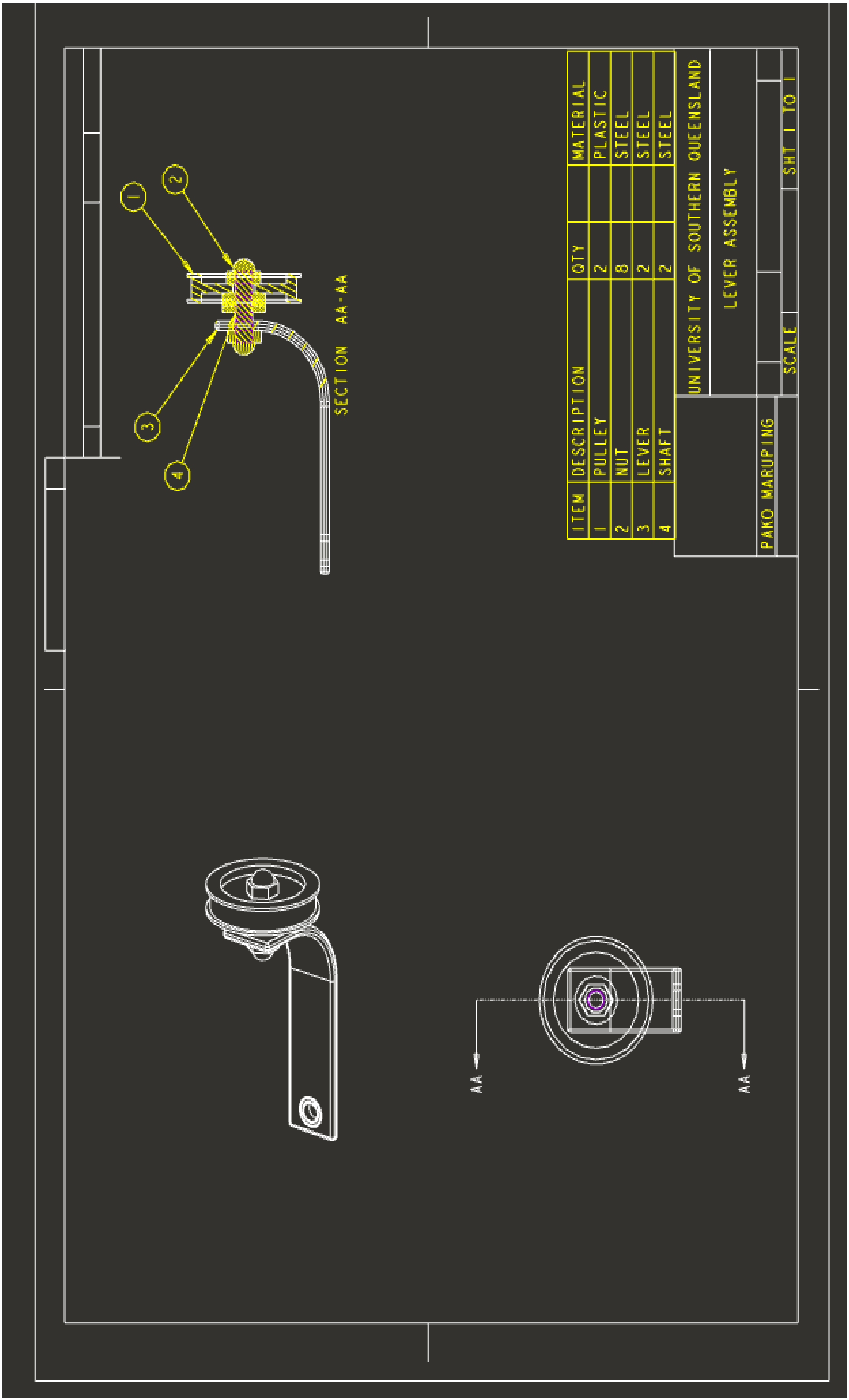
ITEM	DESCRIPTION	QTY	MATERIAL
1	BOX	1	25*25*3 STEEL
2	PLATE 3	2	STEEL
3	LEVER	2	STEEL
4	STICK	1	STEEL
5	PULLEY	2	PLASTIC
6	PLATE 2	2	STEEL
7	PLATE 1	2	STEEL

UNIVERSITY OF SOUTHREN QUEENSLAND

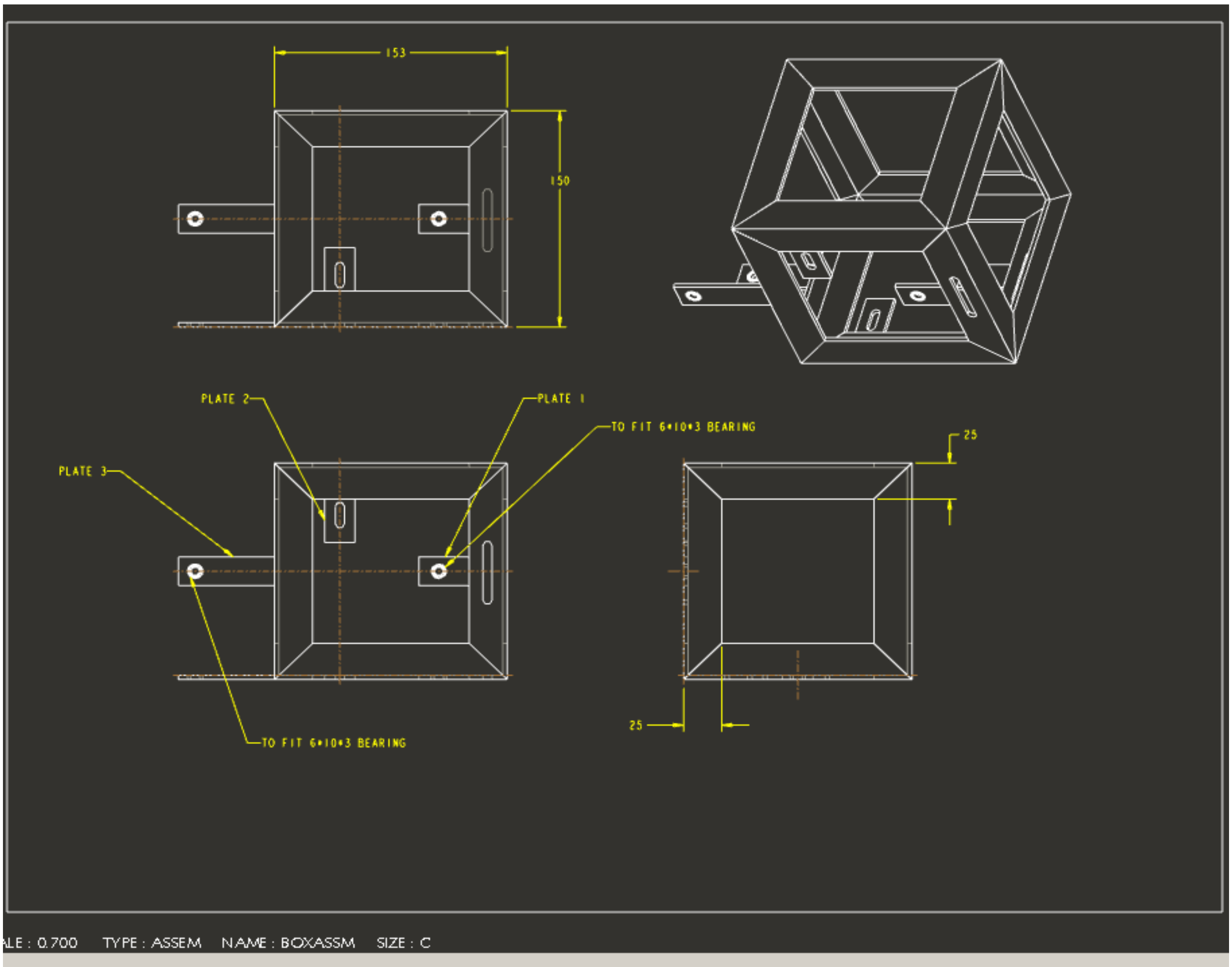
JOYSTICK

PAKO MARUPING

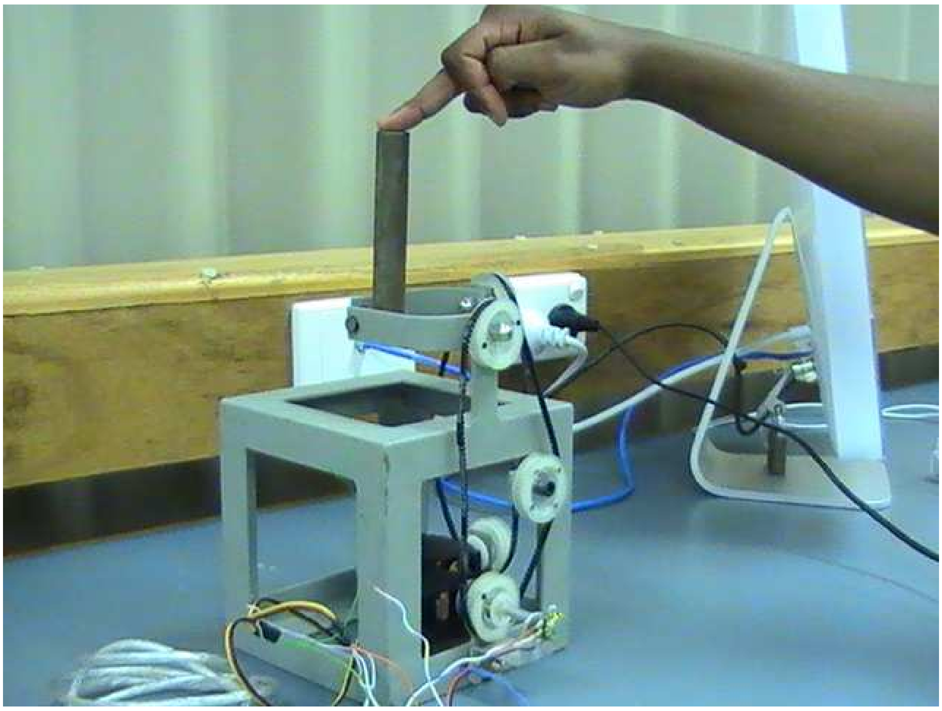
SCALE 1 TO 0.4

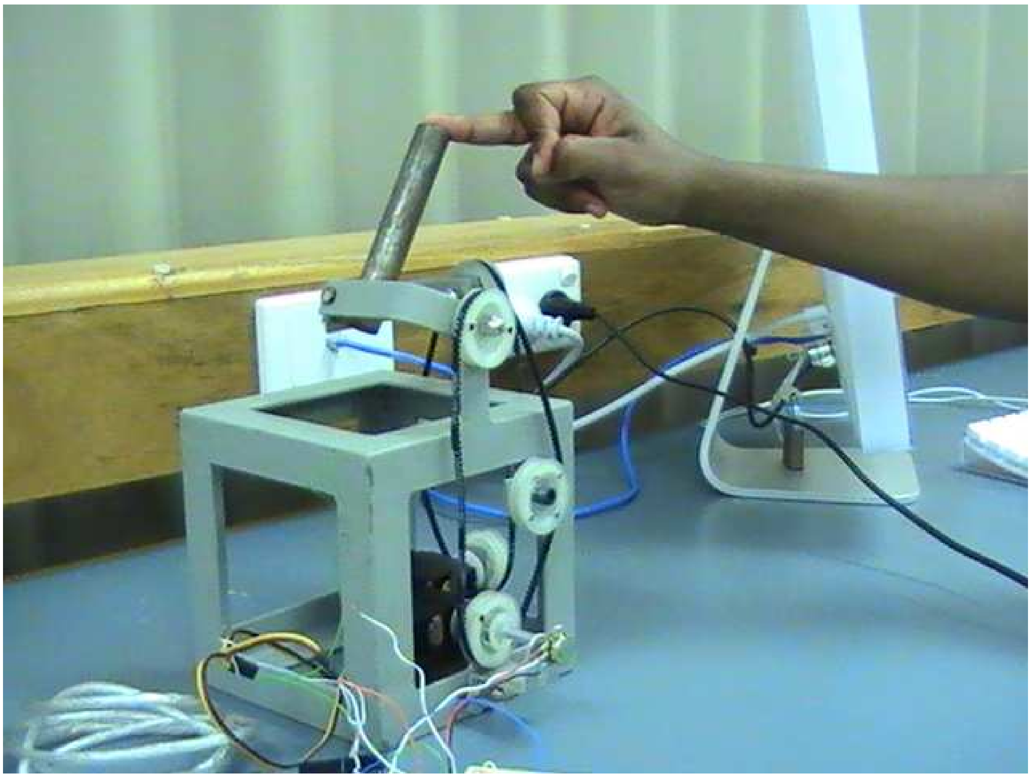


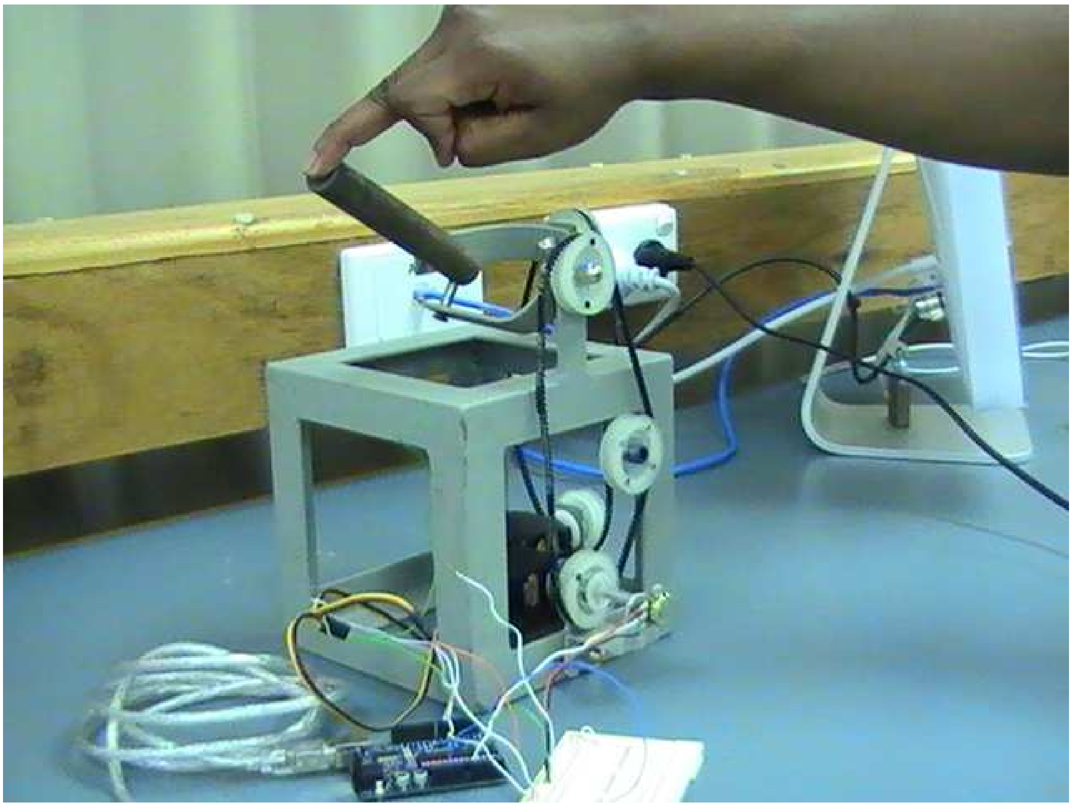
ITEM	DESCRIPTION	QTY	MATERIAL
1	PULLEY	2	PLASTIC
2	NUT	8	STEEL
3	LEVER	2	STEEL
4	SHAFT	2	STEEL

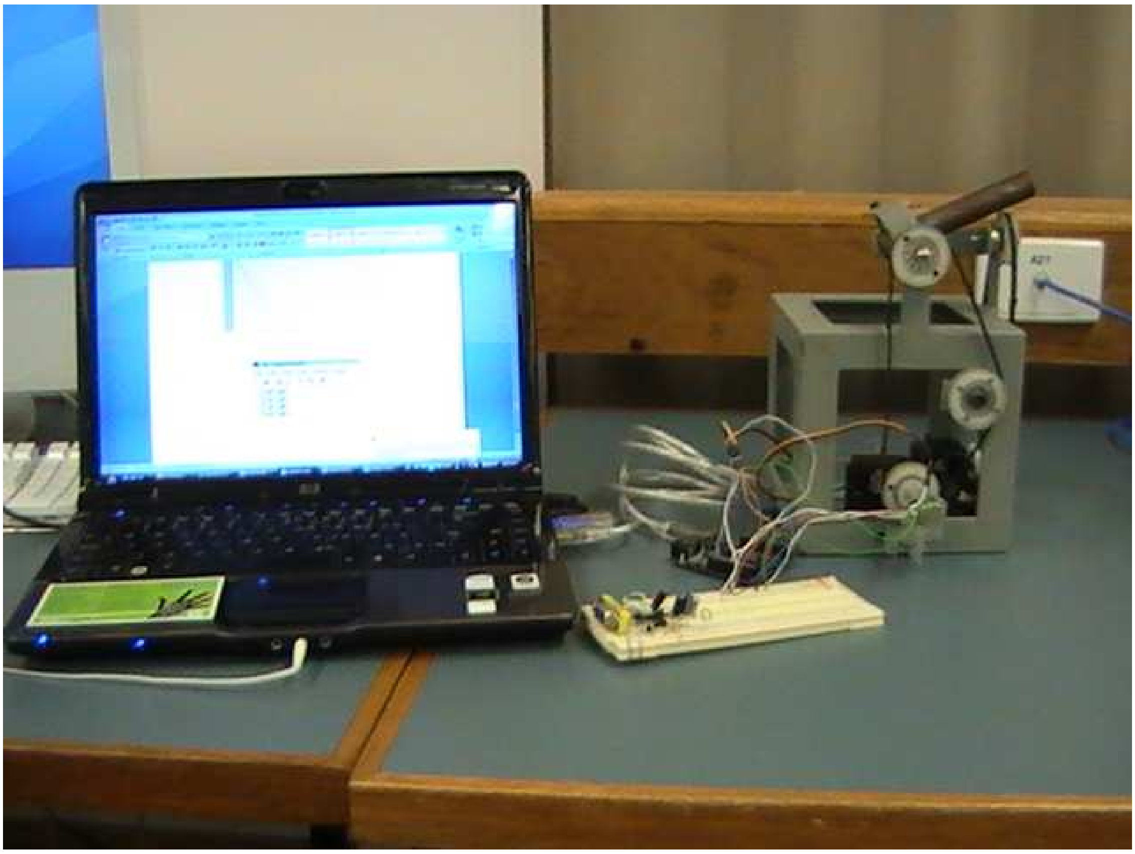

153
150
PLATE 2
PLATE 1
TO FIT 6*10*3 BEARING
PLATE 3
25
25
TO FIT 6*10*3 BEARING
ALE : 0.700 TYPE : ASSEM NAME : BOXASSM SIZE : C

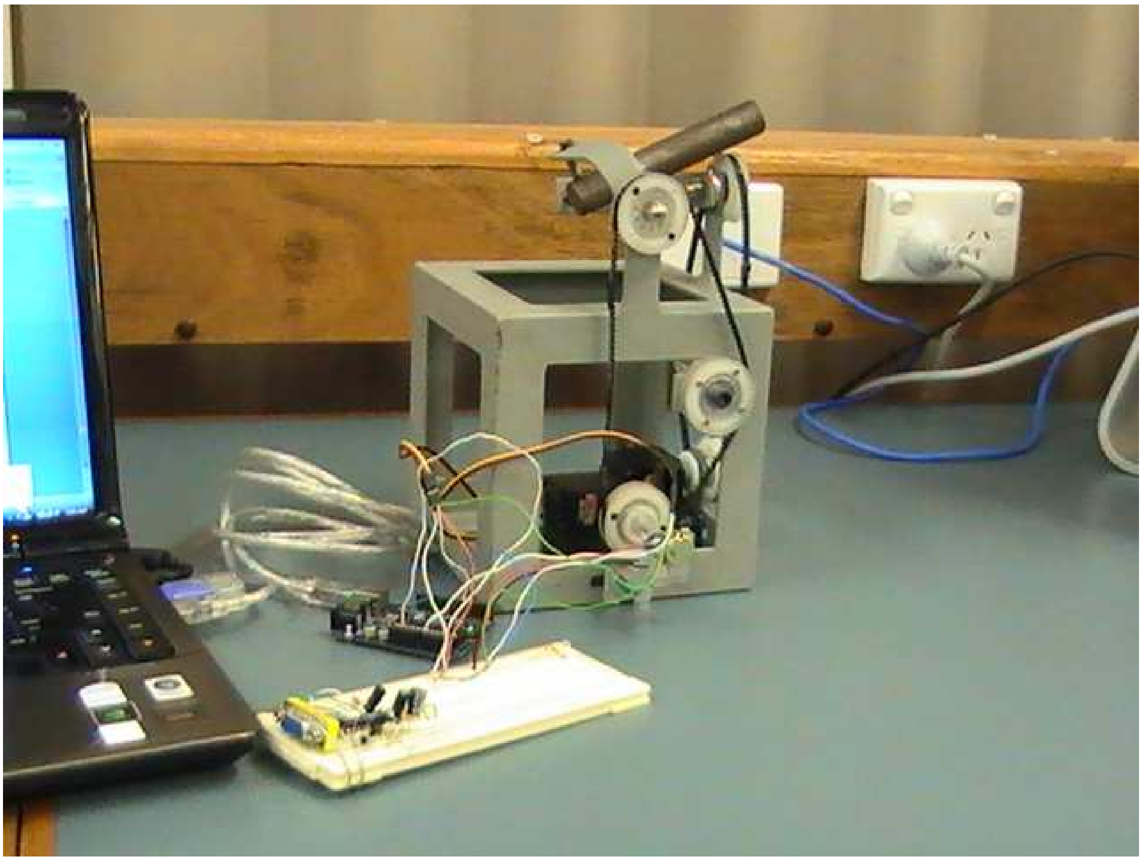
APPENDIX D: MODEL PICTURES

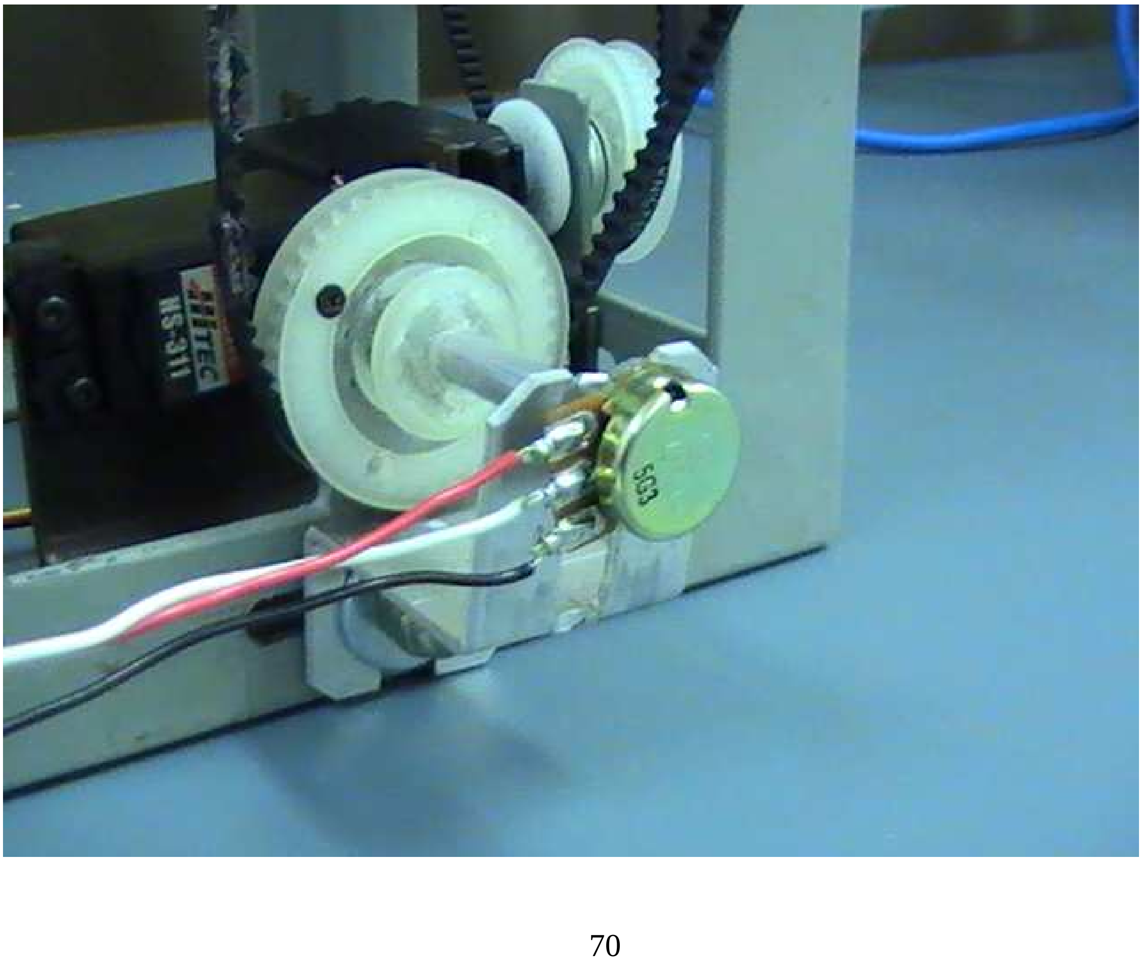


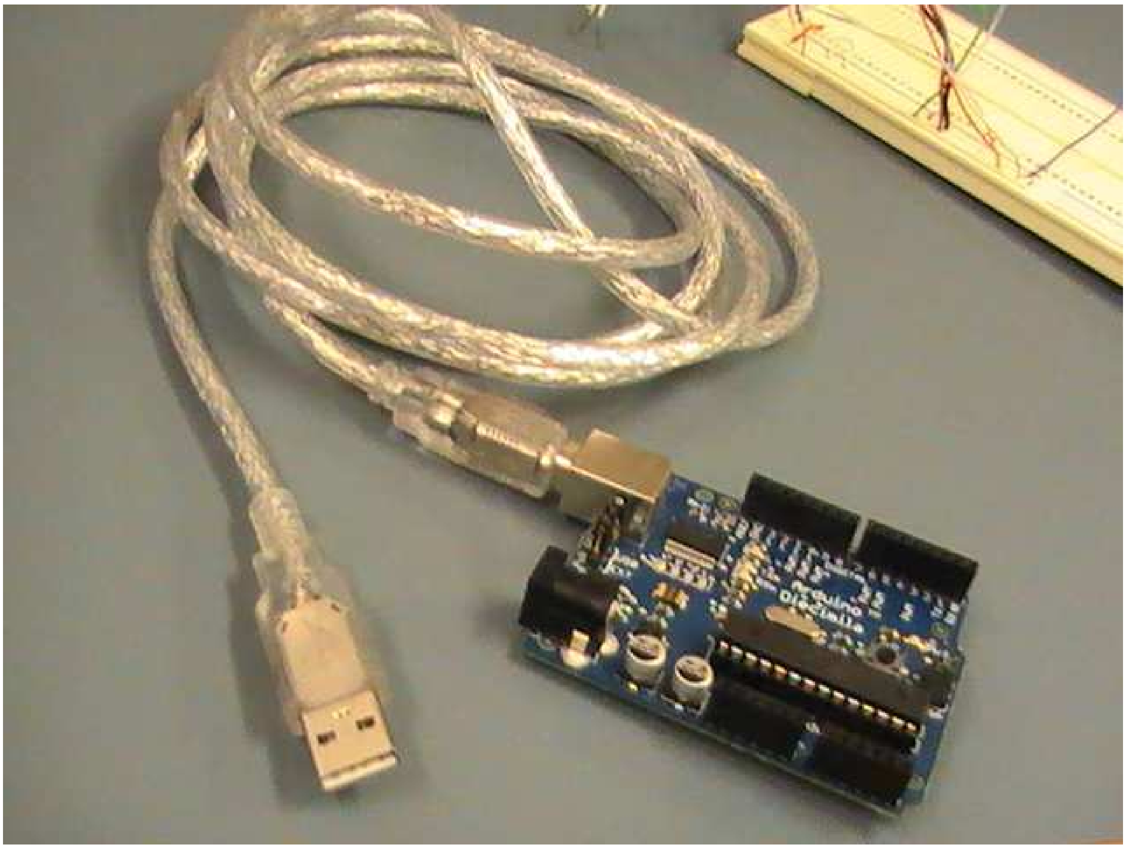









HiTEC
HS-311
503



APPENDIX E THE PROGRAM

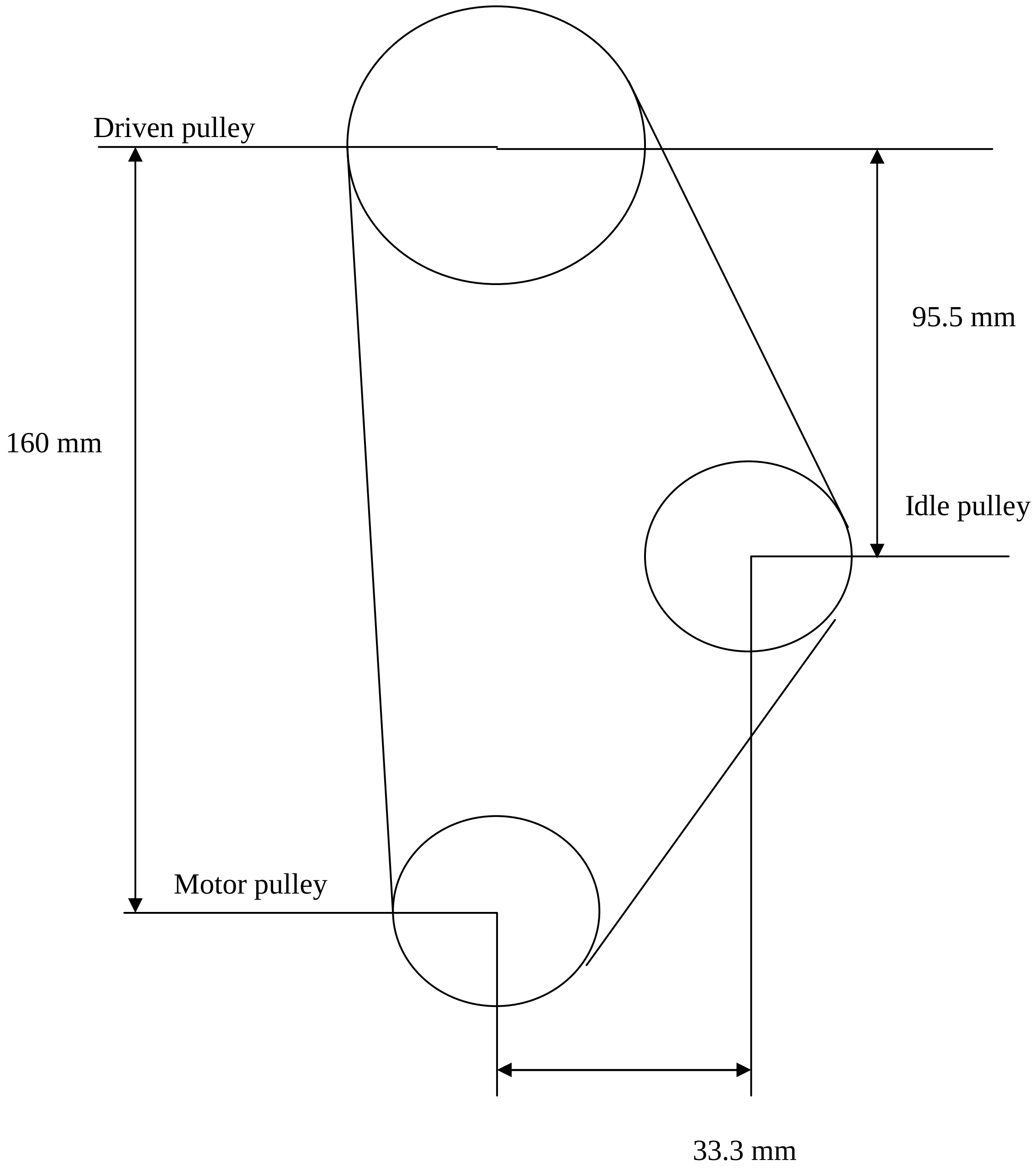
```
//Input/outputs
#define Pot2Pin 0 // analog pin 7
#define PotPin 5  // analog pin 5

#define Lservo  4 // digital pin 4
#define Rservo 5  // digital pin 5


//Variables
unsigned int Ptime = 0;                          // raw value from sensors
int Drive = 0;                                  // PWM value sent to Servos
int Error[5];                                  // array of 5 Error elements
int P = 0;                                    // proportional term
int I = 0;                                   // integral term
int D = 0;                                    // derivative term
int SetPoint = 510;                              // perfect balance point value
byte Count = 0;                                 // counter
int Lmotor = 0;                                 // left motor
int Rmotor = 0;                                 // right motor
int Pot = 0;                                   // potentiometer variable
int analogValue = 0;
int Pot2 = 0;

//tests should be made to determine acurate Min, Mid and Max values for the servos
```

```
#define Midl 1500                                   // center for servos, they should be stoped
#define Midr 1500



//PID constants
#define Kp 100
#define Ki 0
#define Kd 15

//Meaningful names for error array index:
#define Current 0
#define Sum 1
#define Last 2
#define SecondToLast 3
#define Delta 4



void setup() {

  pinMode(Lmotor, OUTPUT);
  digitalWrite(Lmotor, LOW);
  pinMode(Rmotor, OUTPUT);
  digitalWrite(Rmotor, LOW);

  Serial.begin (9600);
//Serial.println("start");
```

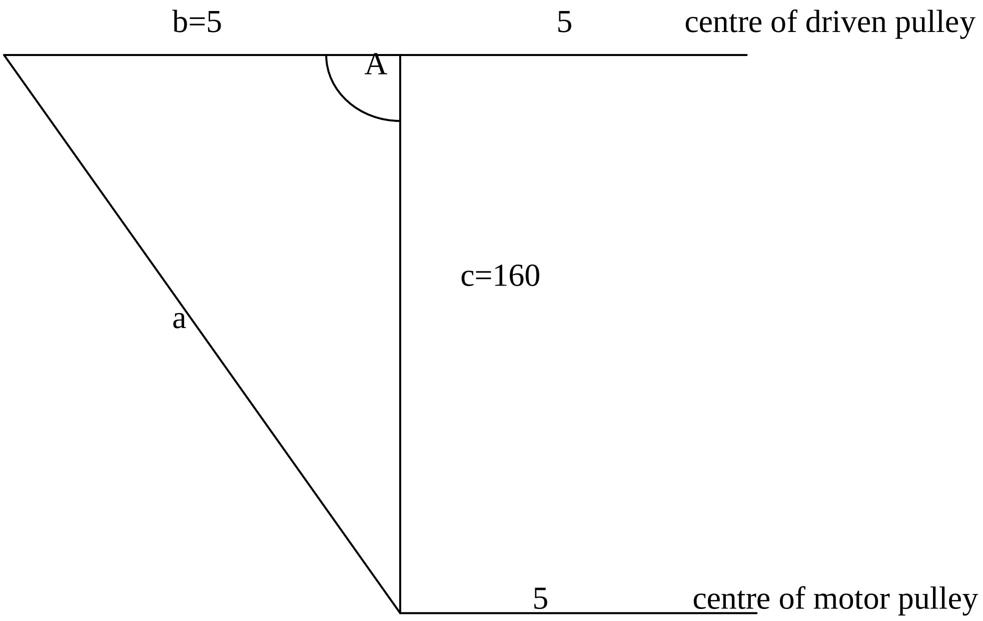
```
  delay(2000);    //wait 2 seconds before start for debug purposes
}

void loop(){
  Read_Pot_Sensor();
    PID();
  Drive_Motors();
  delay(7);     //wait 7 miliseconds, adjust this value for a 18 to 20 ms loop
  if (millis()<30000){
  // print the time, in 100ths of a second:
        Serial.print( SetPoint);
        //Serial.print(",");
            //Serial.print(millis()/10);
            // print a comma:
            Serial.print(",");
            // print the sensor value (connected to analog pin 0):
            Serial.println(analogRead(0));
  }
}

int Read_Pot_Sensor() {
  analogValue = analogRead(PotPin);
   Pot = (analogValue / 10) * 19 + 500;    // convert the analog value
                                            // to a range between minPulse
                                            // and maxPulse.
  //Serial.print ("Pot = ");                  // debug - remember to comment out
  //Serial.println (Pot, DEC);               // debug - remember to comment out
  return Pot;
}

int Read_Pot2_Sensor() {
 Pot2 = analogRead(Pot2Pin);
  Serial.print ("Pot2 = ");                  // debug - remember to comment out
  //Serial.println (Pot2, DEC);              // debug - remember to comment out
 // return Pot2;
}

int PID() {
  Error[Current] = SetPoint - Pot;                          //the target position - the poterntiomenter
position is the error
    P = Error[Current] * Kp;
  Error[Sum] = Error[Current] + Error[Last] + Error[SecondToLast];
  I = Error[Sum] * Ki;
  Error[Delta] = Error[Current] - Error[Last];
  D = Error[Delta] * Kd;
  Drive = P + I + D;
  Error[SecondToLast] = Error[Last];
  Error[Last] = Error[Current];
  Drive = Drive * 4;
  //Serial.print ("PID = ");
  //Serial.println (Drive, DEC);
  return Drive;
```

```
}


void Drive_Motors() {
  Lmotor =  Drive;
  digitalWrite(Lservo, HIGH);
  delayMicroseconds(Lmotor);
  digitalWrite(Lservo, LOW);
  //Serial.print ("Left = ");
  //Serial.println (Lmotor, DEC);

  Rmotor =  Drive;
  digitalWrite(Rservo, HIGH);
  delayMicroseconds(Rmotor);
  digitalWrite(Rservo, LOW);
  //Serial.print ("Right = ");
  //Serial.println (Rmotor, DEC);
}
```

APPENDIX F: Calculating the length of the cord around the pulleys



Calculating the shortest possible length of the cord



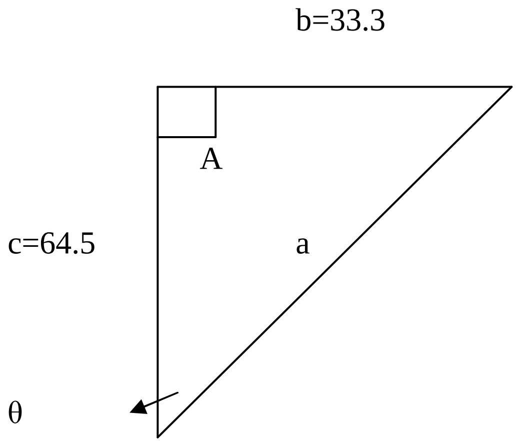
Using Cosine Rule

$a^2 = b^2 + c^2 – 2 * b * c * \cos A$

$a^2 = 5^2 + 160^2 - 2 * 5 * 160 * \cos 90^o$

$a^2= 25625$

$a = 160.08 \approx 160$ mm

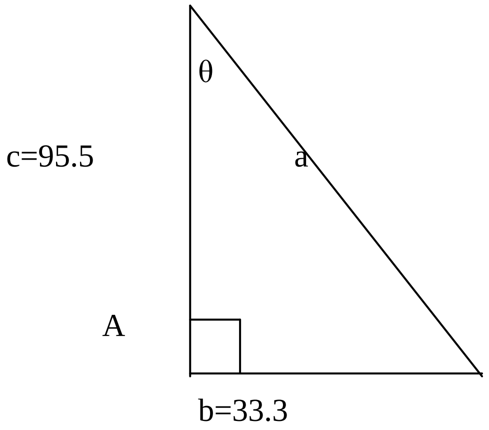


$a^2 = 33.3^2 + 64.5^2 - 2 * 33.3 * 64.5 * \cos 90^o$

$a = 72.6$

$\sin \theta = 33.3/72.6$

$\theta = 27.3^o$



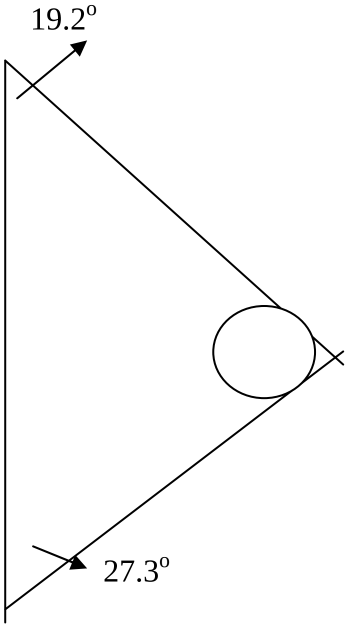
$a^2 = 33.3^2 + 95.5^2 - 2 * 33.3 * 95.5 * \cos 90^o$

$a = 101.1$

$\sin \theta = 33.3/101.1$

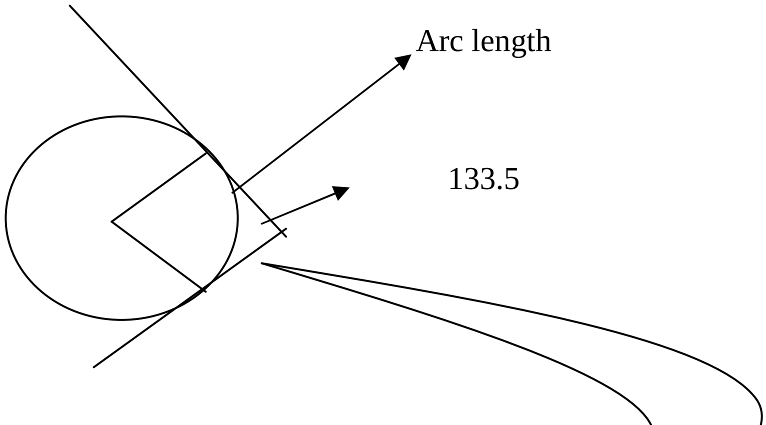
$\theta = 19.23\ ^{o}$

Calculating arc lengths

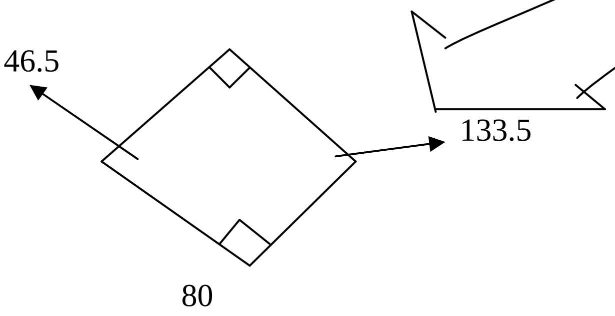


So   180-(19.2+27.3)= 133.5

Idle pulley



As the radius lines meets a tangents at 90 degrees this makes a kite shape



Therefore θ is: 360-(133.5+19.2+27.3) = 46.5

$$\text{Length of arc } = \frac{\theta}{360} * 2\pi r$$

$$\text{Length of arc } = \frac{46.5}{360} * 2\pi * 5$$

Length of arc =4.1

As for the motor and driven pulley it’s going to be assumed that the arc angle θ is 180

Motor pulley

$$\text{Length of arc } = \frac{180}{360} * 2\pi * 5$$

Length of arc =5π=15.7

Driven pulley

$$\text{Length of arc } = \frac{180}{360} * 2\pi * 10$$

Length of arc=10π=31.4

<u>The shortest possible length of the cord</u>

72.6 + 101.1 + 160 + 4.1 + 15.7 + 31.4 = 384.9 ≈ 385mm

APPENDIX G: NOTES ON PID

PID CONTROL

The PID controller has been in use for over a century in various forms. It has enjoyed popularity as a purely mechanical device, as a pneumatic device and as an electronic device. The digital PID controller using a microprocessor has recently come into industry.PID stands for "Proportional, Integral, and Derivative." These three terms (parameters) describes the basic elements of a PID controller.

Proportional

The proportional part of the PID simply closes the loop and gives the system its stiffness and responsiveness. The proportional term give the system a control input proportional to the error. Try to drive the system with only the proportional term makes the system unstable. The proportional is usually the distance you want the robot to travel, or perhaps a temperature you want something to be at. Using a too large P gives an unstable system.

$M(t)=Kp*e(t)$

Derivative

The derivative term gives an addition from the rate of change in the error to the system control input. This improves the response to a sudden change in the system state or reference value. Driving the system on the proportional term only makes the system to be unstable so introducing the derivative term makes the system stable, the D term reduces the delay in the system and make the system stable. The D term is usually the velocity.

$M(t)=Kd*de(t)/dt$

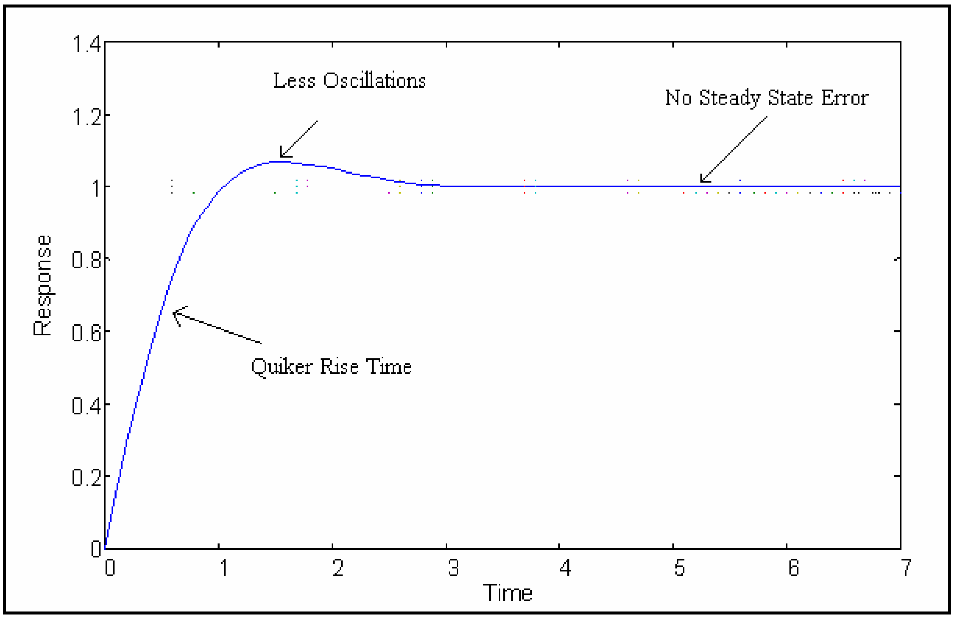
Integral

The integral term gives an addition from the sum of the previous errors to the system control input. The system could actually be drive using the P and the D term only the system will have stiffness and responsiveness and with D term the system will be stable, but if there is any friction, any load or any offsets in the system, the system will drive to approximately to the right direction but not to the exact position which means we will end up with position errors, to avoid this we add the I (integrator) term, the way the integrator woks is such that as long as there is any error the integrator will keep increasing the signal in the right direction until the error is eliminated. This is usually the acceleration.

M(t)=Ki*∫e(t)dt

PID equation

M(t)= Kp*e(t) + Ki*∫e(t)dt + Kd*de(t)/dt



PID response

Properties in response to step input

- The response has a quicker rise time
- It has less or no oscillations
- The steady state error is eliminated

TUNING THE PID loop

There are about four ways to go about selecting PID gains. Trial and error (manual tunning), analytical approach (Ziegler and Nichols), software tools and Cohen-Coon. Using a trial and error approach relies significantly on the operator's own prior experience. The one significant downside to this is that there is no physical insight into what the gain mean and there is no way to know if the gains are optimum by any definition. However this technique has been used for decades and it is still used today for low performance systems.

Table 3: Choosing a Tuning Method

Choosing a Tuning Method		
Method	**Advantages**	**Disadvantages**
Manual Tuning	No maths required. Online method.	Requires experienced personnel.
Ziegler–Nichols	Proven Method. Online method.	Process upset, some trial-and-error, very aggressive tuning.
Software Tools	Consistent tuning. Online or offline method. May include valve and sensor analysis. Allow simulation before downloading.	Some cost and training involved.
Cohen-Coon	Good process models.	Some maths. Offline method. Only good for first-order processes.

Ziegler and Nichols proposed an analytical method, although they originally intended their tuning method for use in process control, their technique can be applied to servo control. This technique has two steps which in this case they general steps.

Step 1:

Set Ki and Kd to zero. The system is excited with a step command. The Kp is then increased slowly until the shaft position starts to rotate. At this point, the value of Kp is recorded and Ko set to equal this value. The oscillation frequency is recorded *fo*

Step 2:

The final PID gains are set using the equations.

Kp = 6Ko, Nm/rad

Ki = 2*fo*Kp, Nm/ (rad · sec)

Kd = Kp/8*fo* Nm(rad/sec)