Robot Controller and Manipulator

by

Zulkifli Bin Md Zin

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)

JUNE 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Robot Controller and Manipulator

by

Zulkifli bin Md Zin

A project dissertation submitted to the
Electrical and Electronics Engineering Program
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL AND ELECTRONICS ENGINEERING)

Approved by,

(Assoc. Prof. Dr. Mohd. Noh bin Karsiti)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

ZULKIFLI BIN MD ZIN

ABSTRACT

This project, entitled Robot Controller and Manipulator comprises of development of a manipulator and controller. The project scopes are to develop an articulated manipulator with three degrees of freedom (3-DOF), a PC-based robot controller, capability of robot to determine path plan and some graphical user interfaces. The invention of robot manipulator aggressively carried out during the industrial revolution, starting from twentieth century. The robot manipulator is extensively used in industry since it is believed that its implementation has several advantages over conventional ways, such as several jobs executed by human are very dangerous, designed components are sometimes having defects and long period of time taken to execute repetitive jobs.

Generally, the scope of this project is to develop 3-DOF robot manipulator by considering dynamic properties. 3-DOF is purposely for positioning function. If the manipulator is needed to be equipped with manipulating function, it is required higher than 3-DOF system, usually 6-DOF.

The project is divided into two parts; manipulator design and controller design. The manipulator design involves construction of manipulator from suitable material, implementation of forward and reverse kinematics, workspace determination, movement analysis and finally integration of the designed components.

The controller design consists of development of dc motor drive system, controller chip by using PIC microcontroller, motor feedback system and graphical user interface at host PC. The user will control the manipulator through this user interface and this host PC will communicate with PIC microcontroller through serial communication.

ACKNOWLEDGEMENT

First of all, I would like to express my highest gratitude to Allah S.W.T for His bless and help that finally, I able to complete my project.

In preparing this project and report, I would like to acknowledge the important contribution from the following persons:

My father and mother, for their supportive help,
My supportive supervisor, Assoc. Prof. Dr. Mohd. Noh bin Karsiti,
Supportive lecturer, Mr Haris bin Md. Khir,
Lab technicians, Mr. Isnani bin Alias, Miss Siti Hawa and Mr. Azhar, and
All of my supportive collegues for their help and support.

Also this acknowledgement goes to the companies that involve in my project, where I get all of my project materials and support:

Eng Huat Hardware Trading,
Perniagaan Aluminium Parit,
Keang Sen Kun Kee Foundry,
IP Bearing & Belts Sdn Bhd,
Syarikat Ipoh Hardware Sdn Bhd,
State Electronic Trading Co, and
Caltrus (Electronic) Sdn Bhd

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

A/D	- Analog/Digital
DOF	- Degrees of Freedom.
EEPROM	- Electrically Erasable Programmable Read Only Memory
LED	- Light Emitted Diode
ISR	- Interrupt Service Routine
MCU	- Microcontroller Unit
MOSFET	- Metal Oxide Semiconductor Field-Effect Transistor.
PID	- Proportional, Integral, Derivative
PWM	- Pulse Width Modulation.
SCARA	- Selective Compliance Assembly of Robot Arm
SPI	- Serial Peripheral Interface
UASART	- Universal Asynchronous Receiver Transmitter

CHAPTER 1

INTRODUCTION

1. BACKGROUND OF STUDY

Since the industrial revolution, there are increasing demands to improve quality and reduce manufacturing cost. Traditionally, the products are manufactured by craftsmen and the quality of the product totally dependant on the skill of the craftsmanship.

At the beginning of twentieth century (1905), Ford Motor Company introduced the concept of mass production [1]. This mass production has been proven to reduce production cost and a big amount of products can be manufactured. This mass production technique uses production machines and called as hard automation. From that, the process automation widespread until the introduction of robot manipulators that able to perform specific tasks, such as material handling, spot welding, spray painting and assembling.

In this project, a robot controller and manipulator need to be designed. The main characteristics of this robot controller and manipulator are movement in three degree of freedom (3-DOF) and manipulator design with kinematics properties. The degree of freedom is the number of independent inputs required to precisely position all links of the mechanism with respect to the ground. The number of DOF of a mechanism is also called the mobility. In this project, by specifying three-coordinates system (x, y and z) is adequate to represent any points in space.

The aim of this project is to develop an articulated robot manipulator and controller that exactly locate at any position in space with 3-DOF. To the extent of application of this design, the robot manipulator can be improved by introducing machine loading application, pick and place operation, welding, painting, inspection, medical application and remote location.

2. PROBLEM STATEMENT

At the beginning of the project, several problems have been identified that yield the relevancy of implementing this project as listed below:

- Humans are sometimes exposed to the hazardous environment at workplace.
- Forming and joining of designed components usually problematic especially in automotive industry
- Longer period of time taken for human to execute repetitive works.
- Technical fault may occur as the result of careless and inadequate skills of worker.
- It is dangerous and not worth for human to carry out hazardous work especially in military, underwater and remote locations.

3. OBJECTIVE

The objectives of this project are as follow:

- To carry out research, design, implement and fabricate an articulated 3-DOF robot controller and manipulator with the ability to locate the manipulator exactly at any position in space.
- To design a robot controller and manipulator with specific kinematics properties.

4. SCOPE OF STUDY

The scope of study could be understood as per described below:

- Manipulator design with the selection of appropriate material especially that having light property. The manipulator is composed of three joints and those joints are designed in such a way that having minimum friction and reliable to move in its workspace. The gearing and belting system also specified under manipulator design.
- Controller design hearted by four PIC microcontrollers that have main functions to drive the motor appropriately and communicate with PC.

5. PROJECT RELEVANCY

The demand for robotic manipulator, especially in fully automated manufacturing company keeps increasing for at least two reasons; firstly, to acquire fast processing time and secondly, to avoid manufacturing defects commonly done by human. In factory, most of the processes are done by automated machines and human will only monitor that process or work at the end of production line.

As the technology keeps changing, the robot manipulator has been experiencing advancement of its design and construction to cater the industry's needs. Thus, the study and design of robot controller and manipulator could be seen as relevant project, as to understand and follow the technology in robot manipulator.

6. PROJECT IMPLEMENTATION PLAN

The project planning is divided into two frames of time, Semester 1 and Semester 2. Table 1.1 shows the project implementation plan for both Semester 1 and Semester 2.

Table 1.1 Project Implementation Plan of Semester 1 and 2.

Division	Semester 1	Semester 2
Manipulator Design	• Study of existing design. • Manipulator design (drawing and dimensioning). • Workspace determination. • Forward and reverse kinematics. • Gear design. • Finding the most suitable material for robot manipulator. • Fabrication of robot manipulator.	• Touch-up on the fabricated manipulator. • Integration of controller and manipulator.
Controller Design	• Path planning. • Pittman motor testing and understanding of its operation. • Development of H-Bridge circuit.	• Microcontroller selection. • Microcontroller programming. • Development of software, as an interface between PC and robot controller. • Development of motor drive system.

CHAPTER 2

LITERATURE REVIEW

1. THEORY

The word robot means different thing to different people. Robot is defined as a mechanical device that is capable of performing human tasks or behaving in a human-like manner [2]. Robot also means a device that controlled by computer and easily reprogrammable. Thus, manual handling devices (i.e., a device that have multiple degrees of freedom and it is actuated by an operator) or fixed-sequence robots (i.e., any device controlled by hard stops to control actuator motions on a fixed sequence and difficult to change) are not considered to be robots [3]. Another definition is a robot is a re-programmable multi-functional manipulator designed to move material, parts, tools or specialized devices, through variable programmed motions for the performance of a variety of tasks [4].

Based on definitions above, we define robot as a combination of mechanical and electronic device that can be multi-functionally reprogrammable to execute a specific task from variety of tasks depend on its program.

The word manipulator means the executor of process of handling or controlling something with skill or predefined program that is made up of several links connected by joints. The word controller means a device that has data processing ability to instruct, manage and direct specific function or task from data inputted.

When talking about robot manipulator, there are several important elements that related to manipulator, which are links and joints, robotic system and robot classifications. The individual bodies that make up a mechanism are called links. The connection between two links is called a joint. The number of degrees of freedom (DOF) depends on the number of links and joints and the type of joints used. DOF is the number of independent parameters or inputs needed to specify the configuration of mechanism completely. There are five basic types of joints that frequently used in robot manipulator, which are revolute joint, prismatic joint, cylindrical joint, helical joint and spherical joint. In this project, revolute joint type is selected which it permits two paired elements to rotate with respect to each other about an axis.

Robotic system consists of three major parts, which are mechanical manipulator, end effectors and controller. Manipulator and controller have been defined above, and the end effectors are devices that connected to the output link of a mechanical manipulator to grasp, lift and manipulate works piece. In this project, the design only considers the development of manipulator and controller without end effectors since the required DOF is three. In most cases, the first 3-DOF is important for positioning and normally another 3-DOF is required for manipulating. Thus, to obtain a complete manipulator, it requires at least 6-DOF. 3-DOF is quite similar with the representation of a point in space with three coordinates system as shown in the equation below.

$$P = a_x\hat{i} + b_y\hat{j} + c_z\hat{k} \tag{1}$$

Normally, the robot is classified by DOF, kinematics structure, drive technology, workspace geometry and motion characteristics. The kinematics structure is differentiated by open-loop and closed-loop chain, drive technology by electric, hydraulic or pneumatic, workspace geometry by cylindrical robot, articulated robot or spherical robot and finally motion characteristics by planar or spherical mechanism.

In this project, an articulated, electrical drive 3-DOF of robot controller and manipulator will be developed. Since the articulated type is chosen, the joint is selected as revolute type. Under articulated type, all joints must revolute.

The designing of robot manipulator deals with kinematics that specify the aspects of motion without regard to the forces or torques that cause it. The kinematics deals with the positioning of manipulator, velocity and acceleration of the position variables with respect to the time. The joint variables are related to the positioning of the manipulator and kinematics study is said as focal point of robot manipulator since it relates positioning of manipulator with the acceleration, velocity and position variables. That is why the kinematics study is the first concern in robot manipulator design.

However, kinematics study has two different approaches; kinematics analysis and kinematics synthesis. Kinematics analysis deals with the derivation of relative motions among various links of a given manipulator [5]. There are two types of kinematics analysis, which are forward kinematics and inverse kinematics. For the case of to find the set of position and orientation, the corresponding time derivative which will bring the manipulator to the desired position is called inverse kinematics. On the other hand, the joint variables and its time derivative is obtained from sensor or reading, and this data is used to position and orientate the manipulator is called forward kinematics.

Kinematics synthesis is the inverse process of kinematics analysis. In this case, with certain desired kinematics characteristics, the manipulator can be developed. For example, with a given time derivative and position of manipulator, corresponding joint variables and the type of geometry and manipulator can be determined.

2. CONCEPT DESIGN

In general, robot manipulator consists of two parts; mechanical and electronics. The mechanical part usually related to physical design and construction of manipulator whereas electronics part is related to controller and drive system. Figure 2.1 shows the general block diagram for controller and robot manipulator.

The function of microcontroller is to receive instruction commanded by user, converts the input into digital signal and finally sources the signal to the h-bridge to rotate the motor. The rotation of the motor is indicated by high and low signal supplied by microcontroller into two h-bridge inputs. The coordination of 10 or 01 at a time will result the motor to rotate either clockwise or counter-clockwise while 00 will give no signal, thus the motor will static. The microcontroller used is PIC 16F877, manufactured by Microchip. The details about selection of microcontroller and the concept of dc motor application by using PIC 16F877 is discussed in the Chapter 4 under Controller Design.

The motor is specified to rotate in its workspace with the specific degree of rotation. The RS motors with add-on encoder are chosen to be used in this design. Once the motor is rotating, encoder circuit will detect the rotation of motor's shaft and generates a pulse signal. A pulse signal generated is very significant to the degree or distance of rotation, and inputted back to the microcontroller. In order to achieve the required degree of rotation, microcontroller will do some calculation works and will stop the motor as the pulse count achieved.

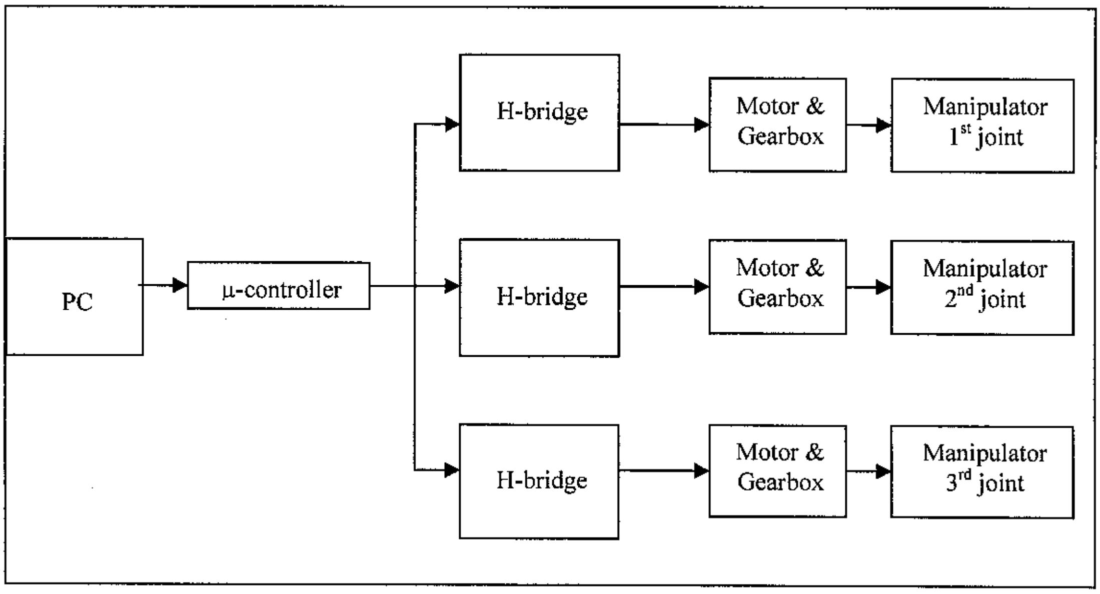


Figure 2.1 Robot Controller and Manipulator Block Diagram.

The motor is not directly driven from output pin of microcontroller, but using a device, called h-bridge. This circuit will act to determine the direction of motor, normally set as clockwise or anti-clockwise and to totally stop the motor. Note that, this circuit cannot control the speed of the motor and only microcontroller is able to control the speed with the variety of input voltage signal supplied to the motor.

The function of motor is to move or revolute the joint. One motor is assigned for a joint. The rotational angle of motor must be accurate to ensure that manipulator is exactly located at desired coordinate. Manipulator first joint, second joint and third joint have the task to position at any coordinate in 3-dimension workspace, particularly in x-axis, y-axis and z-axis.

In this design, a PIC 16F877 microcontroller is used. The controller controls a motor and responsible for positioning the joints in its axis. This microcontroller also programmed to communicate with PC. The communication with PC is carried out by using RS-232 connection through serial communication.

In order for controller to communicate with PC, the interface must be developed where software in PC and programming for microcontroller must be established. This user interface is developed by using Visual Basic or HyperTerminal.

The relationship between computer software and microcontroller is like this; the user will specify the location or coordinate for manipulator in computer program, computer program will convert it to instruction that understood by microcontroller and finally that microcontroller will control motor's operation. The motor will rotate accordingly and stops if the required degree is achieved indicated by adequate encoder pulse signal. Figure 2.2 shows how robot manipulator and controller work.

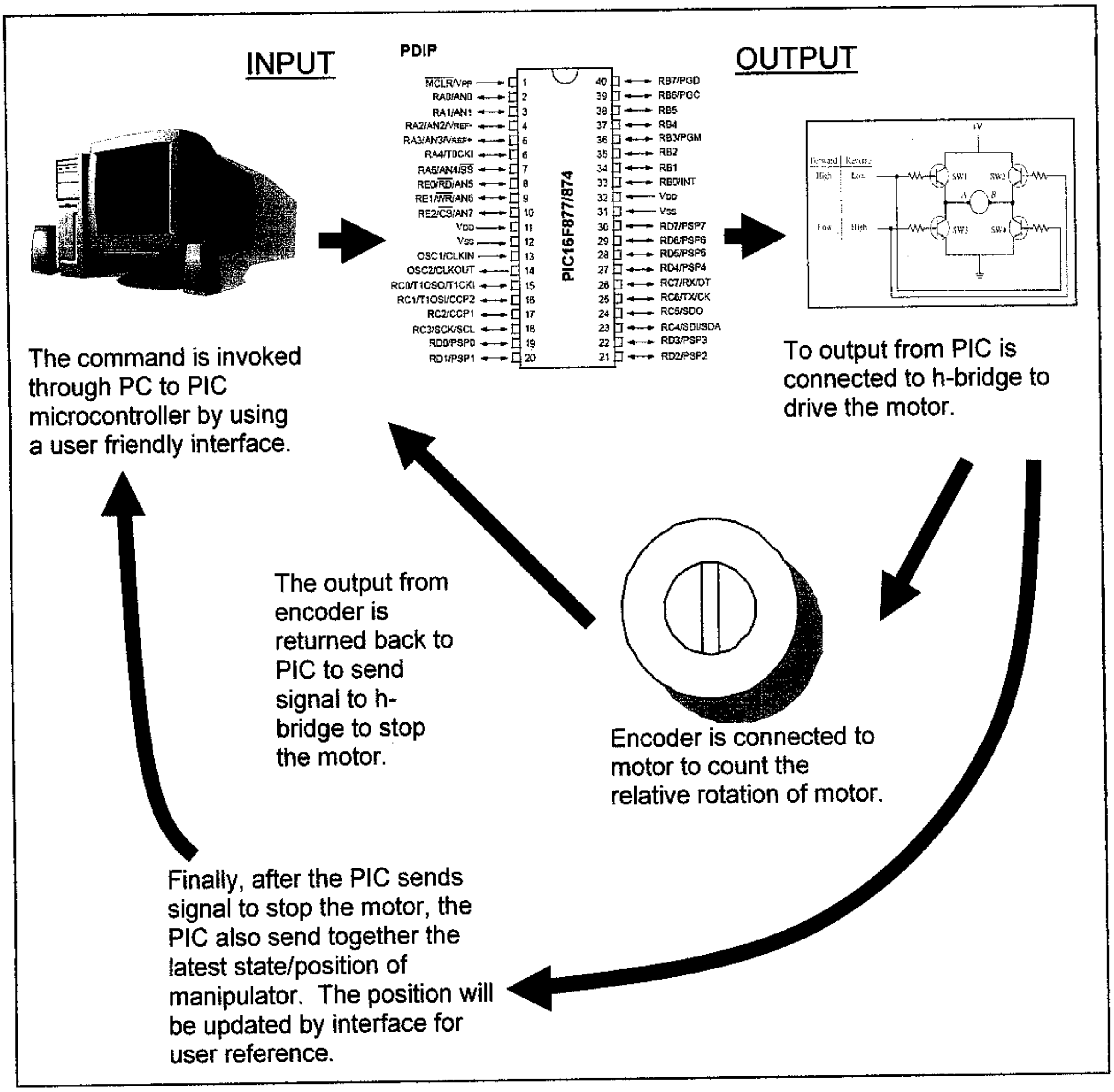


Figure 2.2 The Operation of Robot Controller and Manipulator.

CHAPTER 3

MANIPULATOR DESIGN

1. SELECTION OF ROBOT COORDINATE FRAME

As has been discussed earlier, the manipulator has several coordinates, namely cartesian, cylindrical, spherical, articulated and Selective Compliance Assembly Robot Arm (SCARA). The illustration for these types of robot coordinate is shown in Appendix A.

For the purpose of this project, an articulated coordinate frame is more interesting than the others. The reason is an articulated robot's joints are all revolute and similar to human arm. This configuration is very famous and perhaps mostly used in industry. The advantage of this configuration over other types is an articulated has greater accessibility in its workspace with the same robot's dimension.

2. STUDY ON EXISTING DESIGN

A study on the existing design has been conducted to find the ideas for manipulator design. The area of interest is only on developing mechanical and physical design and manipulator's orientation. The observations include the method for each joint is connected, physical orientation and arrangement and shaft connection with manipulator's joint.

Study and observation of design has been made in Industrial Automation Lab and through robotic books and sites. Among the type of robots that has been studied are Mitsubishi Robotic Arm, ED Robot 5-Joint Robotic Arm Trainer, Motoman ES165, Adept-One robot and Fanuc S-900W robot. The illustration of these robotic types is shown in Appendix B.

Among the important observations found is the arrangement of motor, which to make sure that the accessibility of manipulator at its maximum with the proper arrangement. Another observation is the mounting of gear and connection of joint with the base and joint with the joint.

3. CONCEPTUAL DESIGN AND DIMENSIONING

The design of robot manipulator consists of four parts: a base, joint 1, joint 2 and joint 3. Figure 3.1 and 3.2 shows the component of robot manipulator.

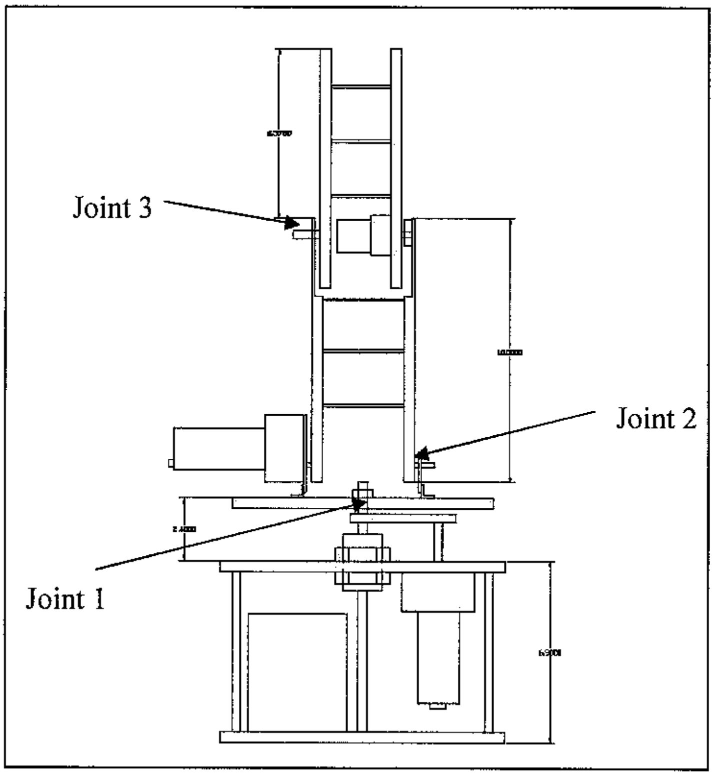


Figure 3.1 The Components of Robot Manipulator In Terms of Joints

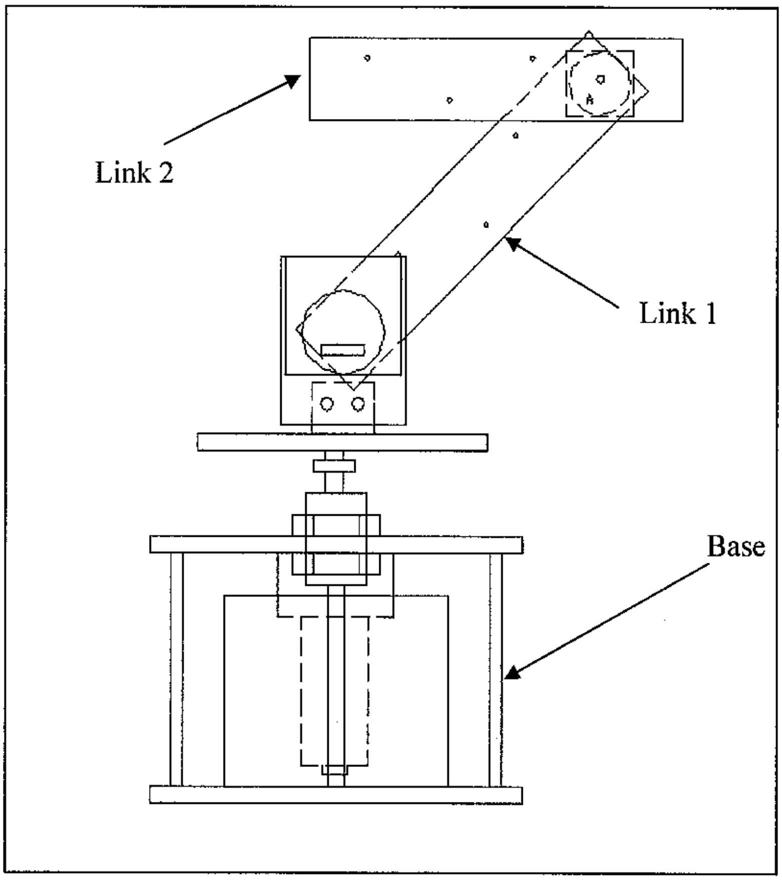


Figure 3.2 The Components of Robot Manipulator In Terms of Links

Controller box and dc motor for joint 1 are placed in the base. Joint 1 is used to rotate the manipulator with respect to z-axis, joint 2 is used for y-axis and joint 3 is also for y-axis. Even though there is no specific joint for rotating with respect to x-axis, by catering the rotation in both y- and z-axis for all joints, any 3-dimension position in its workspace can be achieved. With the accurate revolution of joint 1, joint 2 and joint 3 ensures that the exact location of manipulator could be established. Appendix C shows the revolution of each joint with respect to the specific axis, indicated by θ_1 for first joint's revolution angle, θ_2 for second joint's revolution angle and θ_3 for third joint's revolution angle.

The base width is 11.00 inches. The wide base compared to joint's width is important to make sure the stability of manipulator is optimum. The height for base, link 1 and link 2 are 6.9 inches, 9.3 inches and 7.00 inches respectively. The height of this manipulator, as the all joints are fully stretched upward achieves 26.275 inches including the space occupied by base gear and base joining. The illustration for manipulator dimension and its front, side and top view could be seen in Appendix D.

The manipulator design uses two units of adaptable gearbox, to suit with two dc motors. The function of these gearboxes is purposely for speed reduction which robot manipulator normally requires quite slow movement, approximately 20 rpm.

4. WORKSPACE DETERMINATION

Workspace is defined as the volume of space the end effectors can reach [6]. Two different types of workspace are frequently used which are reachable workspace and dexterous workspace. A reachable workspace is the volume in workspace which every point that can be reached by end effecters by at least one orientation while dexterous workspace is the volume in workspace which every point can be achieved with any orientations.

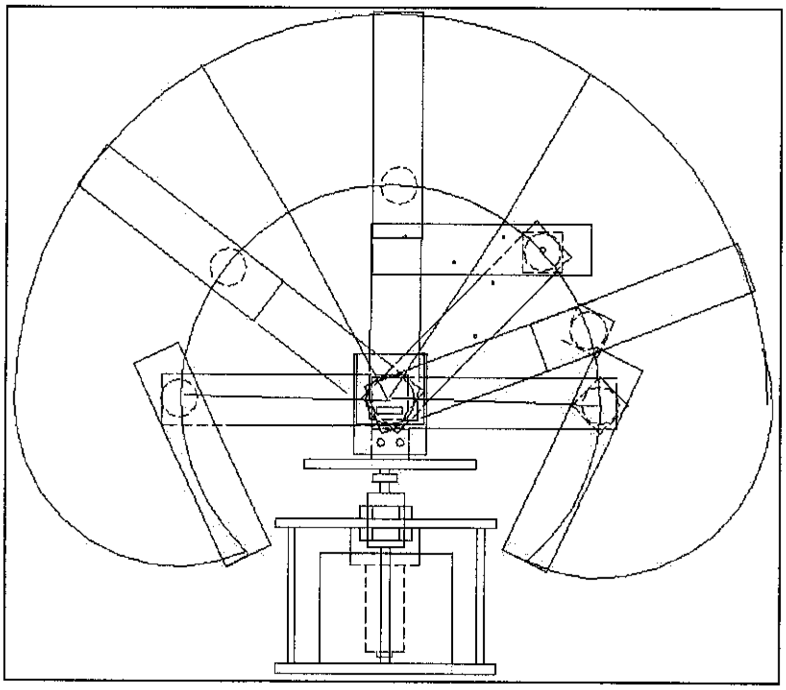


Figure 3.3 The Process of Determining Workspace

The workspace of manipulator is firstly determined by specifying the limit of rotational angle for each joint. In the earlier design stage, the limit of rotational angle is determined and then the volume of workspace that can be achieved by manipulator might be developed. Then, the movement of robot's manipulator is analyzed to determine its workspace. Appendix E shows the workspace for the manipulator.

Thus, from the design, the workspace of rotational angle for each axis can be developed as follow:

$$
\begin{aligned}
&-150^\circ \leq \theta_1 \leq +150^\circ \\
&-90^\circ \leq \theta_2 \leq +90^\circ \\
&-135^\circ \leq \theta_3 \leq +135^\circ
\end{aligned}
\tag{2}
$$

5. FORWARD AND INVERSE KINEMATICS

Forward and inverse kinematics is very important and vital part of robot manipulator since it serves as the basic and fundamental element of positioning the manipulator. Forward and inverse kinematics is always related to as compulsory principle in position analysis.

Briefly, with forward kinematics, we are able to determine the position of manipulator if all joint variables are known. On the other hand, with inverse kinematics, we are able to find the joint variables if the specific position of manipulator is specified. Normally, the process of describing objects, locations, orientations and movements is done by using matrices.

It is important to realize that, in reality, the end effectors that attached to the manipulator is different size in its length, height and width. For short end effectors, the end will be at different location than long end effectors. During the design stage, most of the forward and inverse kinematics considers only manipulator. Thus, for the accuracy of positioning, the forward and inverse kinematics must consider the end effectors that we want to attach to.

In a 1-DOF system, all variables can be known upon specification of any variables. For example, when a variable is set to a certain value, the mechanism is totally set and other variables are known. Normally, the mechanism consists of closed-loop and open-loop. Most of the closed-loop system mechanism is in the form of 1-DOF. The example of closed-loop mechanism is given by closed-loop four bar mechanism, which if we set the movement of one link to certain degree, let say 120°, the other links' angle also can be detected.

3-DOF robot manipulator is open-loop mechanism. Unlike in the closed-loop mechanism, if all of the variables are specified in open-loop system, there is no guarantee that the manipulator can exactly position at the given location. This is because if there is any deflection in one link, it may affect the location of the manipulator. As a result, the construction of 3-DOF robot manipulator must strong enough to eliminate all deflections. Figure 3.4 below shows the closed-loop and open-loop mechanisms.

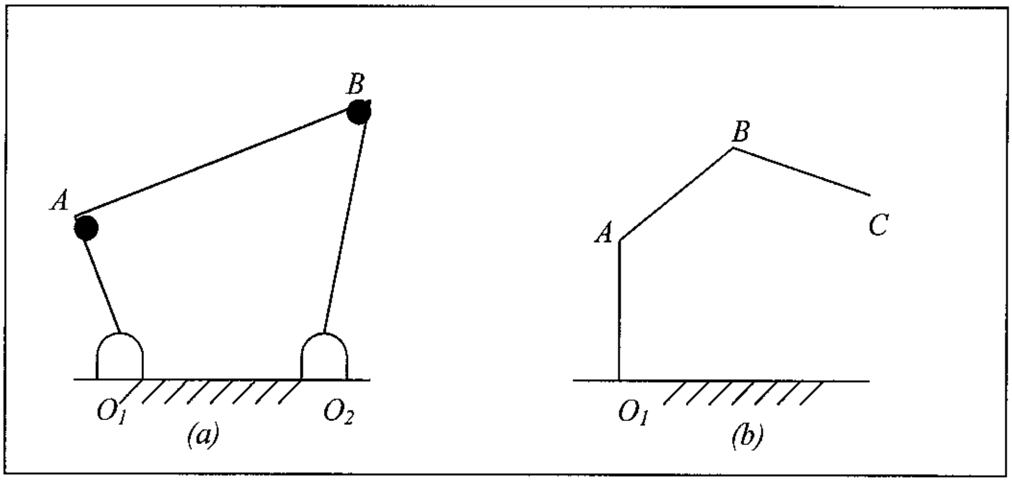


Figure 3.4 Robot Mechanisms, (a) Closed-Loop (b) Open-Loop

Another difference between closed-loop and open-loop mechanism is closed-loop mechanism provide feedback while open-loop mechanism is does not. In a feedback system, any deflection of a link or error occurs can be detected. For example, if deflection occurs at link AB, the link O_2B will detect the changes but in open-loop system, there is no detection since the deflection will move all succeeding members without feedback.

To remedy this problem in open-loop robot, the position of the hand is constantly measured with devices such as camera, the robot is made into closed-loop system with external means such as the use of secondary arms or laser beams [7].

In this project, the position analysis is done by analyzing at the given location or coordinate, what is the angle required for joint 1, joint 2 and joint 3 to revolute. This approach is done by applying trigonometric function. For example, if an articulated 3-DOF robot manipulator in this design want to move to location from (0,0,0) to (7, 4,10) the joint variable required is as below:

$$\begin{aligned} \theta_1 &= +29.75^\circ \\ \theta_2 &= +16.71^\circ \\ \theta_3 &= +93.24^\circ \end{aligned} \tag{3}$$

where θ_1, θ_2 and θ_3 is rotational angle for joint 1, 2 and 3 respectively. In this study, the joint rotational angle can be determined by applying forward kinematics with the given position. Inversely, if we know the rotational angle, the position can be determined by applying inverse kinematics. Appendix F shows the detailed calculation of position analysis.

6. SELECTION OF GEAR RATIO

The normal running speed for motor is quite high. The motor for positioning manipulator must less than 20 rpm, normally around 10 rpm. There is a need to decrease the motor speed and a general way to reduce the speed is by implementing gear ratio.

Gears perform two important duties. First, they can make the revolution of a gear greater or lesser than normal motor speed. Second, they also can increase and decrease the power depends on how the gears are oriented. Decreasing the speed of motor normally will increase its torque.

There are several types of gear, namely spur, helical, bevel and worm. In this project, two sets of spur gear combined in two units adaptable gearbox, manufactured by RS are used. Spur gear has teeth parallel to the axis of rotation and used to transmit motion from one shaft to another parallel shaft.

The motor used in this project is RS 12V dc motor. From data sheet, the nominal no load speed for this motor is 2000 rpm. The gear ratio calculation is as below.

If N_1 is the speed for motor and N_2 is the required speed, the gear ratio needed to reduce the speed from 2000 rpm to 20 rpm will be

$$\begin{aligned} Gear\ ratio &= \frac{N_1}{N_2} \\ &= \frac{2000 rpm}{20 rpm} \\ &= 100:1 \end{aligned} \tag{4}$$

It is fortunate since RS provides a gearbox with the ration 100:1. The torque for this adaptable gearbox is 4 Nm. Thus, with the proper gear ratio and torque, this adaptable gearbox is used in the design. The adaptable gearbox is assembled together with RS dc motor. Figure 3.5 below shows the arrangement of RS dc motor and adaptable gearbox.

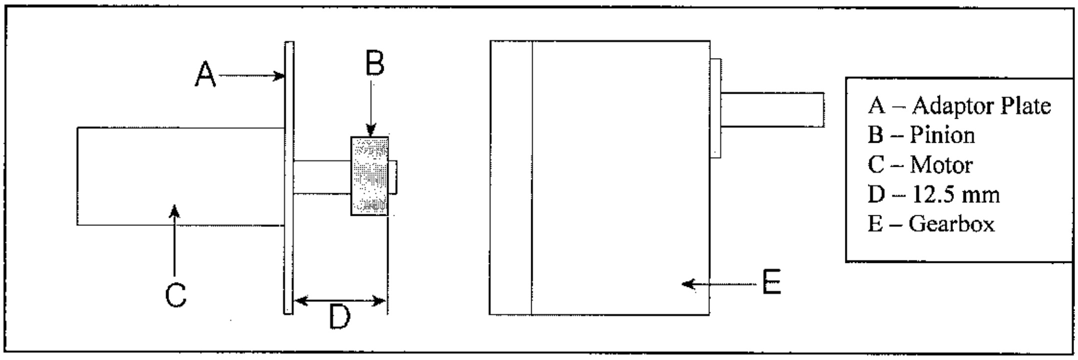


Figure 3.5 Fitting The Motor into Gearbox

The size of gear is expressed as pitch, which is roughly by counting the number of teeth on the gear and dividing it by the diameter of the gear. In order to ensure that the transfer of energy between gears is efficient, the pitch between these gears must be the same. The same pitch also important since different value of pitch may result easy to wear over several period of operation time and non-matching between gears.

7. MATERIAL SELECTION

There are several suitable materials to be used in fabricating the manipulator. Among the materials that can easily be found in market are carbon steel, stainless steel and aluminum. Table 3.1 below shows the comparison and characteristics of these materials.

Table 3.1 Comparison and Characteristics of Materials between Carbon Steel, Stainless Steel and Aluminum.

Characteristic	**Carbon steel**	**Stainless steel**	**Aluminum**
Weight	Highest	Medium	Lightest
Strong	Acceptable	Acceptable	Acceptable
Cost	Cheapest	Medium	Highest
Easiness to fabricate (in terms of easiness to drill, cut and shape)	Lowest	Medium	Highest

Thus, from the above comparison, aluminum is selected as material for constructing robot manipulator. The highest weightage of material characteristic falls to materials' weight since we need to find the lightest material as possible in order for motor to revolute the joint efficiently. Note that, for RS motor, the maximum affordable torque is 4 Nm, if equipped with adaptable gearbox.

From the comparison, the price for aluminum is the highest. Even its price is high, it is still affordable to use this material for fabricating purpose.

8. FABRICATION AND DEVELOPMENT OF MANIPULATOR

The fabrication of manipulator's model is carried out in the lab. The fabrication process needs machining and drilling skills, and all other skills that related to metal work. The process of constructing robot manipulator takes quite long time and the most important thing during the fabrication process is the machined and cut material is capable to combine together with a stable joint to develop several strong links. Figure 3.6 below shows the picture of developed manipulator.

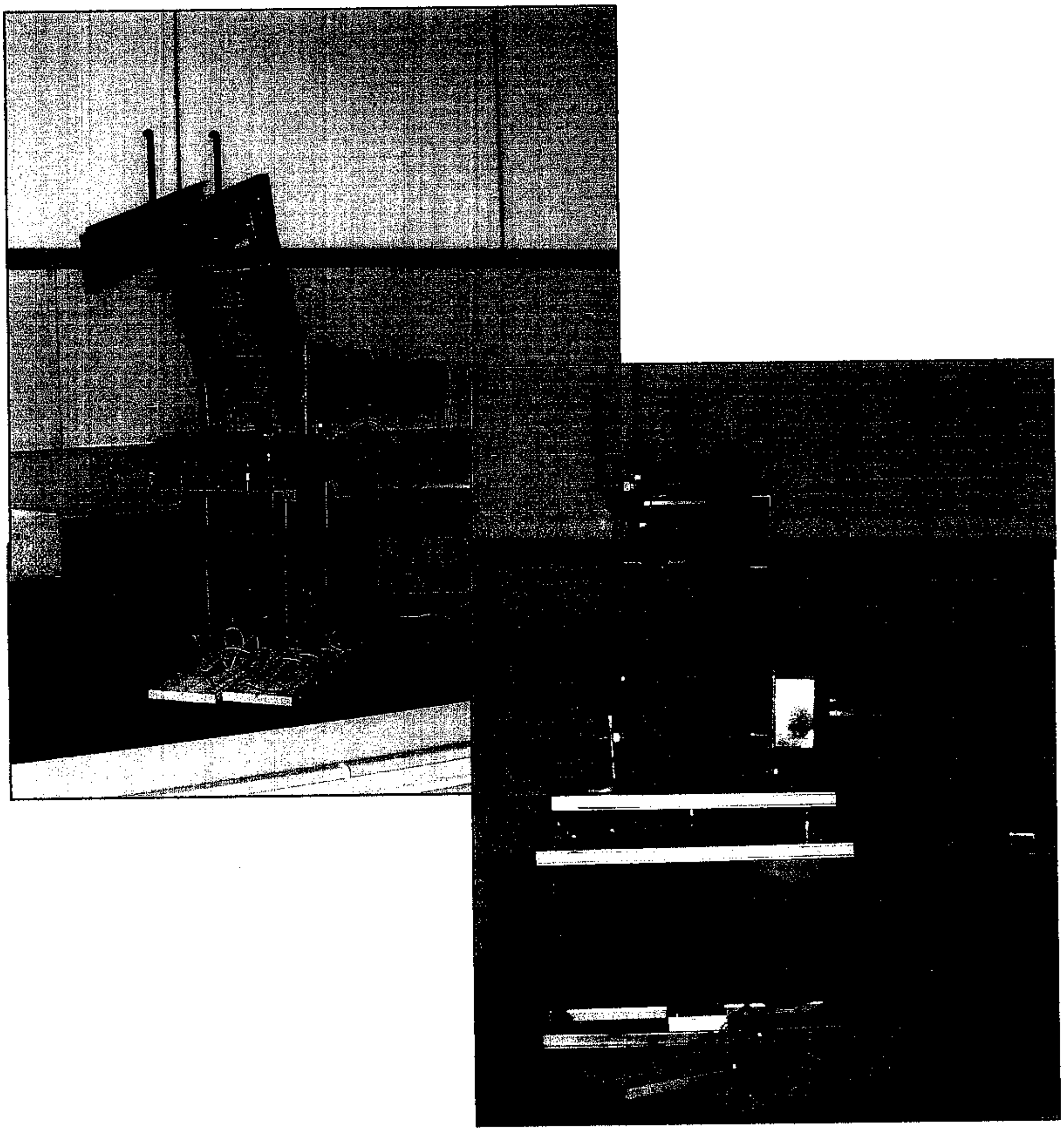


Figure 3.6 The Constructed Manipulator

It is targeted that, the manipulator is constructed from used material as to minimize the total cost, since the cost for new material, especially aluminum is quite high. The suitable used materials can be obtained from thorough finding process from used equipments. Among the potential places to get this such materials are second hand goods dealer, bicycle, motorcycle and car workshop and used equipments at home e.g VCRs, CD player, washing machine and mechanical toys. In case of there is difficult to find used material for certain manipulator's part, new material is chosen.

In manipulator design, we have specified our own dimension and all materials needed must be suited with the dimension. The most probable problems for this condition is unable to find the suitable specific material for needed parts. That is the disturbance and defector while doing robotic project. If the material cannot be found over set time range, there is a solution here which is by fabricating ourselves the manipulator's parts.

9. CHANGE OF MANIPULATOR DESIGN

The current manipulator design has been reviewed for several times. The initial design is quite distinct compared to final design. As noted, the big differences between initial design and final design particularly on its gear implementation and type of motor used.

During initial design, the design involves a lot of gear to be located at each link, to rotate the joint. For example, instead of gear boxes used, there is another gear purposely to mount the joint with motor, driven by belt. Several problems have been identified and difficult to solve, which are difficult to find the suitable gears with specific gear ratio and belt that suits with that gear. Most of the industrial suppliers only supply belts or gears for heavy usage for example big machines, tractors and other industrial equipments. The detail of initial robot manipulator design is shown in Appendix G. The pictures for initial design are shown in Figure 3.7.

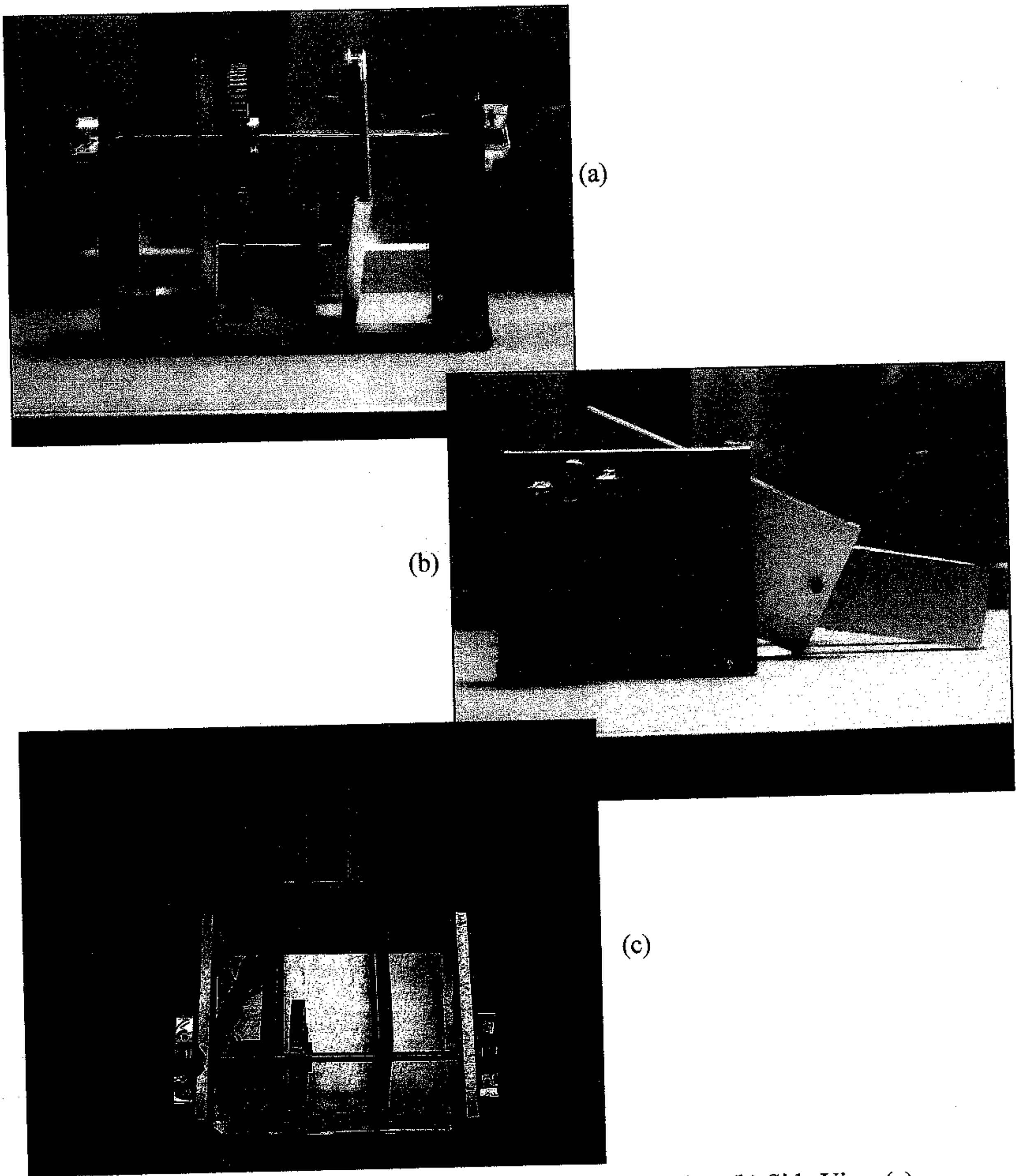


Figure 3.7 The Midway-to-Complete Manipulator; (a) Front View (b) Side View (c) Top View.

Thus, due to unsolved problems for nearly one semester, a change of design must be carried out. This new design (Appendix D) implements as less as possible the usage of gear. It stresses more on implementation of adaptable gearbox and direct drive from the gearbox to manipulator's joint. According to this design, the motor used for initial design which Pittman motor need to be changed to RS since RS motor offers an attractive adaptable gearbox. Furthermore, the gearboxes offered by RS are having a lot of gear ratio varieties. In the new design, several suitable couplers also must be fabricated for the purpose of to obtain direct drive from the output of gearbox to respective joint.

CHAPTER 4

CONTROLLER DESIGN

1. PATH PLANNING

In the previous chapter, under forward and inverse kinematics topic, the kinematics analysis of robot manipulator and its links has been discussed. Under kinematics analysis, by using equation of motion, we can know where the position of manipulator will be if we have joint variables. Path planning actually refers to the way of robot will move from one location to another location. In manipulator study, the movement of robot will be in two types: the normal movement and relative movement.

Normal robot movement is specified as the movement of manipulator to certain position from origin. For example, the positioning of manipulator from (0, 0, 0) or base to (4, 10, 3) is said as normal movement. On the other hand, the relative movement is specified as the movement of manipulator from specific coordinate point to another coordinate point, except base coordinate. For example, the movement of manipulator from coordinate point (4, 10, 3) to (-3, 9, 5) is said as relative movement since the positioning is referred to as another specific point, not at the origin. During setting up the equation of motion, the movement of manipulator always refers to the difference between these points.

A path is defined as a sequence or robot configurations in a particular order without regard to the timing of these configurations. So, if a robot goes from point (and thus, configuration) A to point B to point C, the sequence of configuration between point A and B and C constitutes a path [8]. The most important concern in developing a path is the movement of robot from one point to another point regardless of the time consumed to achieve the desired point.

As the concept applies, in order to move a robot manipulator from a point to another point, the microcontroller signals motor to rotate the gear together with the arm. By realizing the motor behavior and manipulator's path planning, it will accelerate at the beginning of operation until motor's speed reaches maximum speed, indicated as phase 1 and decelerate at the end of operation, indicated as phase 2. For instance, let say if we want to move a link for 10°. The microcontroller gives a signal to motor and motor will rotate. The encoder signal pulse becomes another input to the microcontroller. If an encoder signal for each high pulse indicates 1° of rotation, it requires 10 pulses for microcontroller to stop the motor rotation.

The motor will exactly locate an arm if the speed of motor always constant, starting from beginning until the end. But the reality is not that way. In path planning concept, motor acceleration takes several times, then the motor will be constant in its speed and finally decelerates for several times, Figure 4.1, yielding a small variance in the final destination.

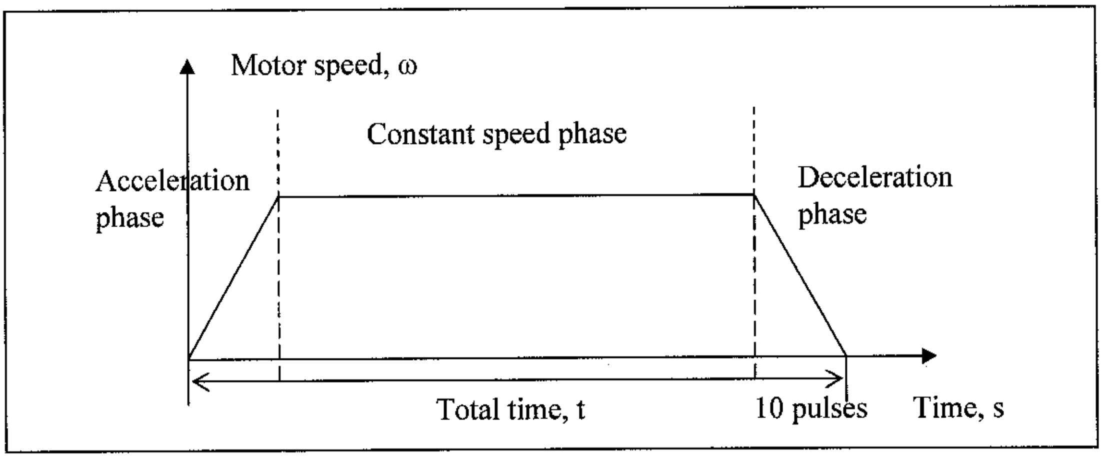


Figure 4.1 Path Planning for Motor Speed.

Note that, the variance in final destination is due to variance in speed, smaller during both acceleration and deceleration phase, making the total travel would be lesser. It will be getting a clear picture from graph of acceleration, as shown in Figure 4.2, which positive during acceleration phase, zero during constant speed and finally negative for deceleration phase.

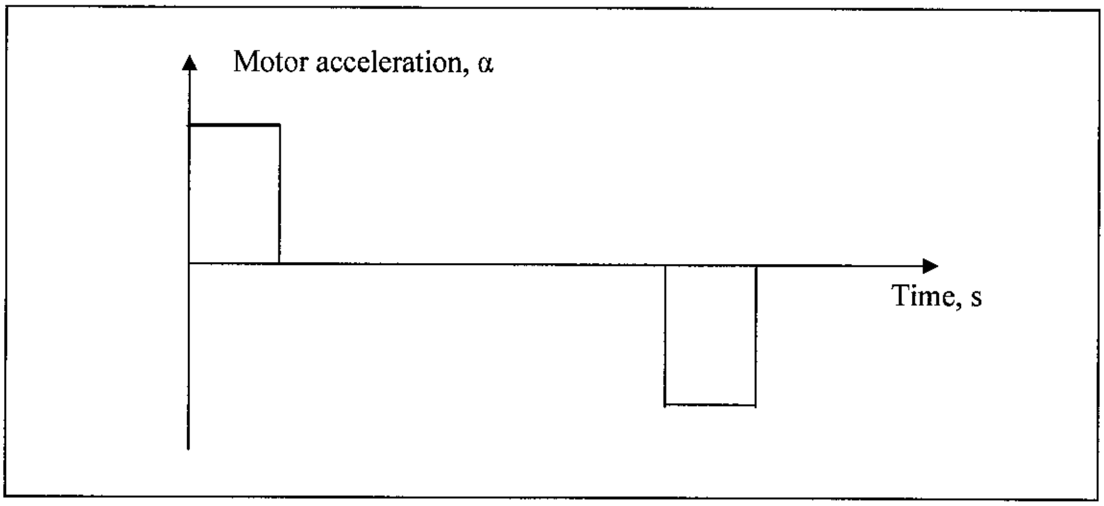


Figure 4.2 Acceleration Behavior for DC Motor.

The above-mentioned problem is a kind of challenge in path planning. This problem is difficult to eliminate but can be minimized by using light material for manipulator and also using intelligent microcontroller that can compensate between operation time, acceleration and distance traveled. Beside that, error calculation function in source code is very useful to minimize this effect.

2. RS DC MOTOR

There are three motors used in this robotic project, type RS 12V dc motor. The motor has ratings of 12V input voltage, 2000 rpm nominal speed at no load and 4 Nm maximum torque when attach to adaptable gearbox.

This motor has a built in encoder at the bottom part. The function of encoder is to calculate the rotation generated by the motor, indicated by pulse. A pulse generated equals to a rotation has been done.

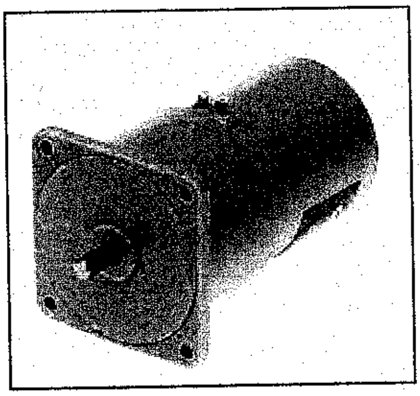


Figure 4.3 12V 60 rpm RS DC Motor for Joint 1

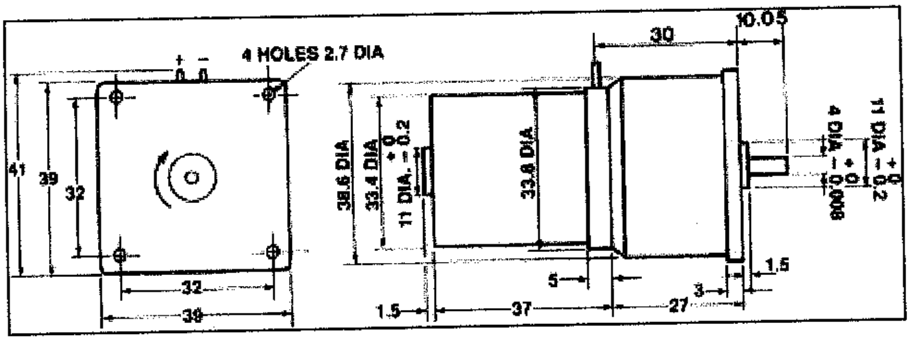


Figure 4.4 Drawing and Dimension of 12V 60 rpm RS DC Motor for Joint 1

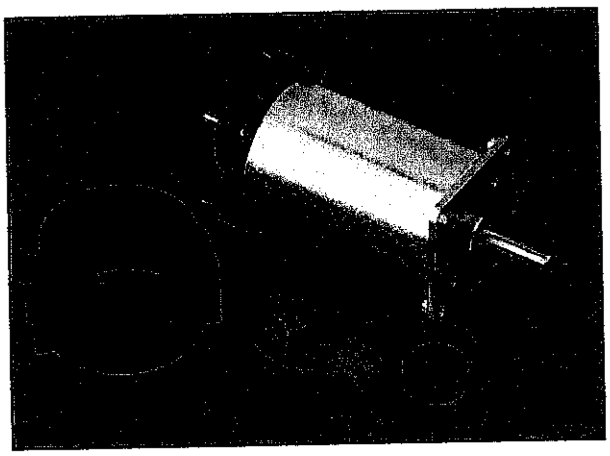


Figure 4.5 12V 2000 rpm RS DC Motor for Joint 2 and 3

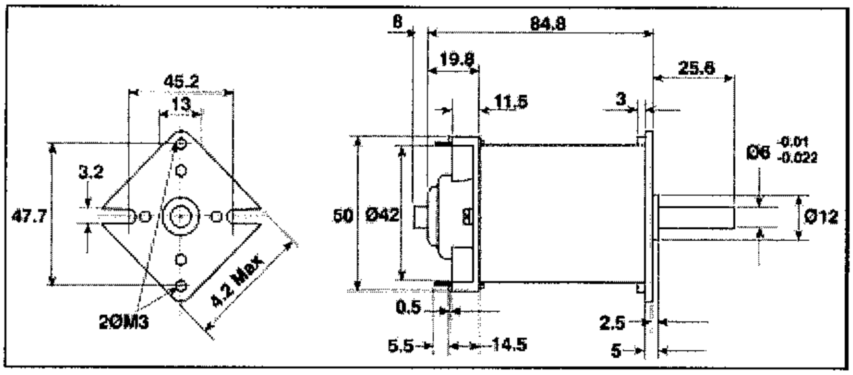


Figure 4.6 Detail Drawing of 12V 2000 rpm RS DC Motor

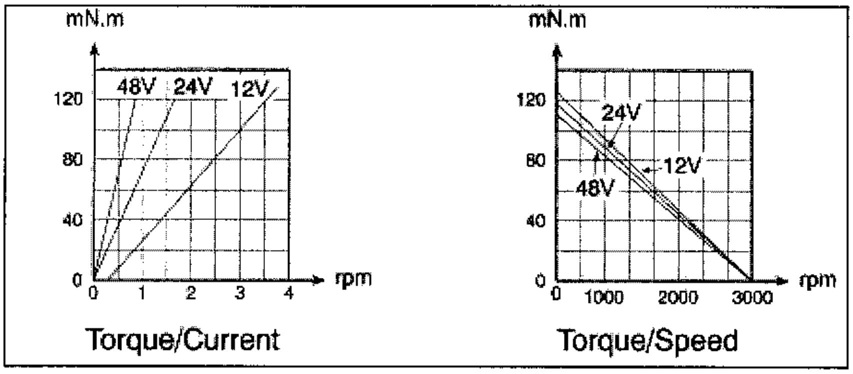


Figure 4.7 Torque – Current and Torque – Speed Characteristic of 12V 2000 rpm RS DC Motor

3. H-BRIDGE CIRCUIT

Another aspect under controller design is development of h-bridge circuit. The function of h-bridge is to control the direction of motor, whether clockwise or anti-clockwise. This circuit actually does not have the ability to control the motor speed. In robotic design, it is desirable to change the direction of motor by changing the direction of current flow. In other words, if the h-bridge is connected to microcontroller, the changing of motor direction can be performed by changing the bit information signaled to h-bridge circuit.

A simple h-bridge circuit can accomplish this as shown in Figure 4.8. As the output from microcontroller change from high to low or vice versa, the direction might be changed. When the transistor SW1 and SW4 are high, the current flows in the direction from A to B. If the transistor SW2 and SW3 are high, the current flows in the direction from B to A. Thus, the polarity will change and motor direction also will be changed.

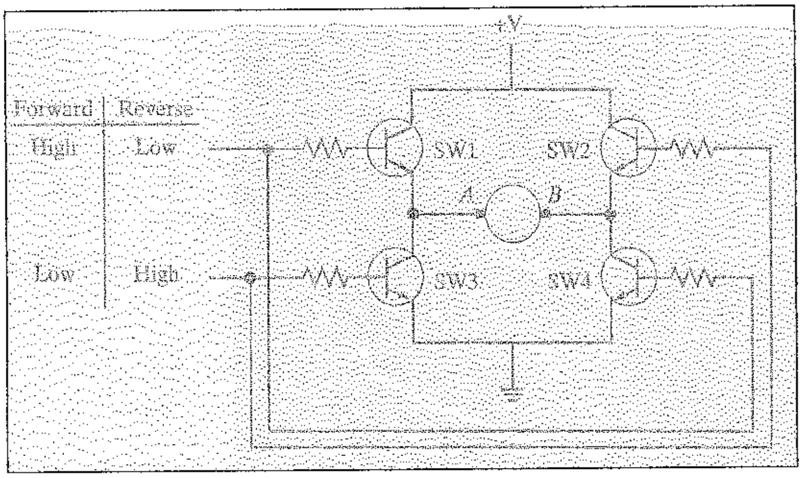


Figure 4.8 Application of H-Bridge for Motor Direction.

The circuit in Figure 4.8 is a simple and for illustration purpose only on its operation. The h-bridge circuit design in this project is shown in Figure 4.9.

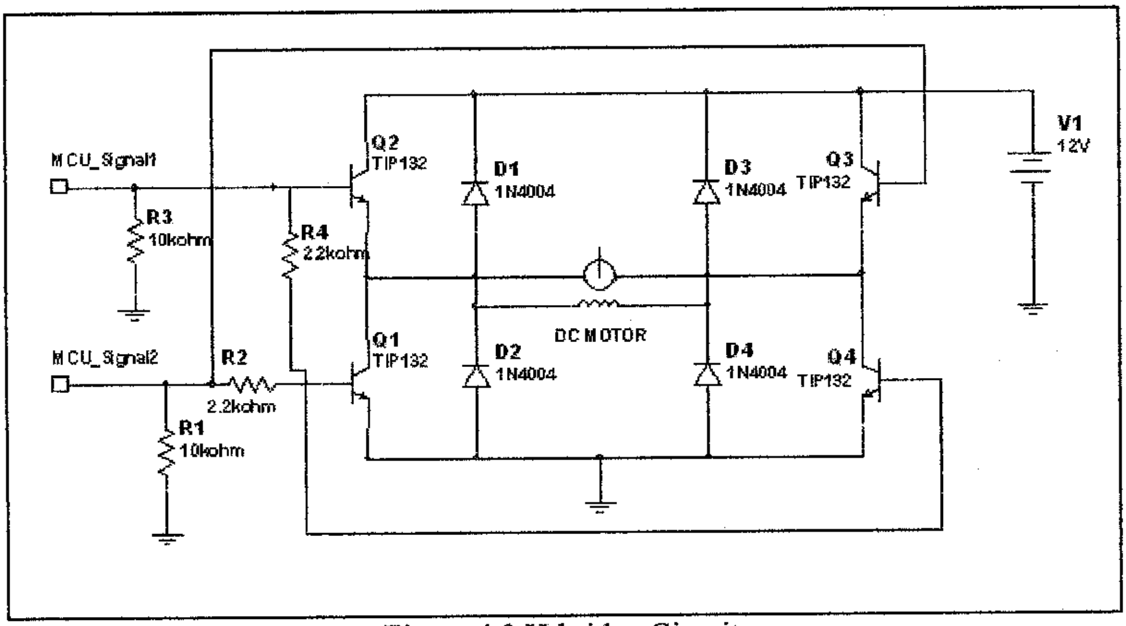


Figure 4.9 H-bridge Circuit

In h-bridge circuit above, R1 and R3 consist of 10 kΩ pull-up resistors. Its function is to eliminate jerk when the microcontroller powers up and down. There are four D1N4004 diodes used. D1 and D3 function is to protect the chips from over voltage by turning on when more voltage is coming from the motor than is coming from the batteries. Finally, D2 and D4 protect the chips from under voltage by turning on when the voltage in the motor is below GND.

The h-bridge circuit shown in Figure 4.9 is the first design of motor drive control. It uses low complementary silicon power darlington transistor, TIP 132 that has current gain, h_{FE} typically at 1000. The required current to drive dc motor is 0.5 A at its nominal speed. Even the current gain is quite high for TIP 132, the transistor still not able to provide sufficient current to drive the dc motor. Note that, the measured output current from PIC output pin is 0.02 mA.

As the above problem is identified, the design is modified by adding TIP 132 at the PIC output pin to amplify the MCU signal as shown in Figure 4.10. The result is also the same where the motor cannot operate properly.

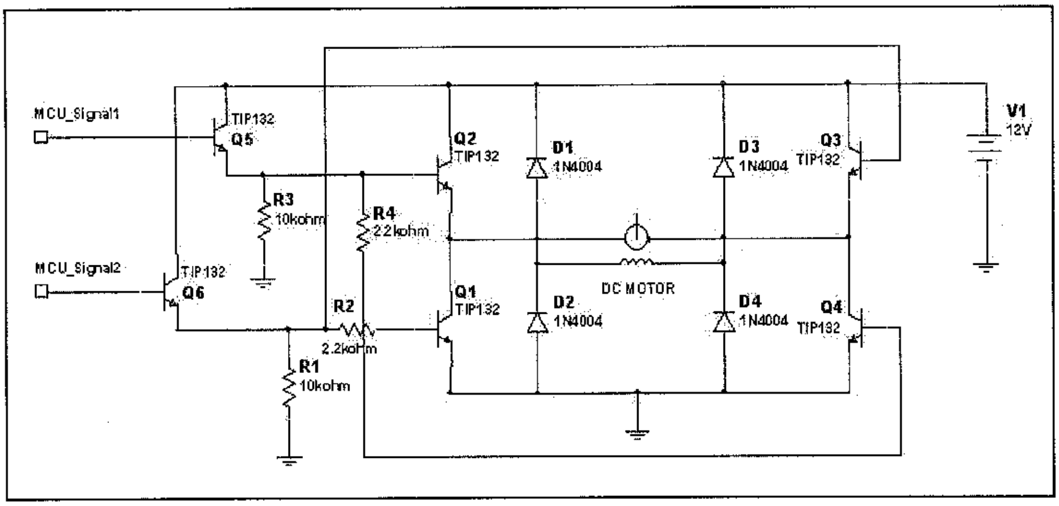


Figure 4.10 Modified H-Bridge Circuit by Adding TIP 132 At the MCU Output Signal.

The final design is specified by using n-channel MOSFET, BS 170. In this design, BJT transistor is also used since BJT provides more switching efficiency compared to MOSFET. In this circuit, the MOSFET function is to increase the current supplied by MCU whereby the switching work is handled by TIP 132. Again, the implementation of BS 170 is essential in order for dc motor to work properly with sufficient current. Figure 4.11 shows the final design of h-bridge circuit.

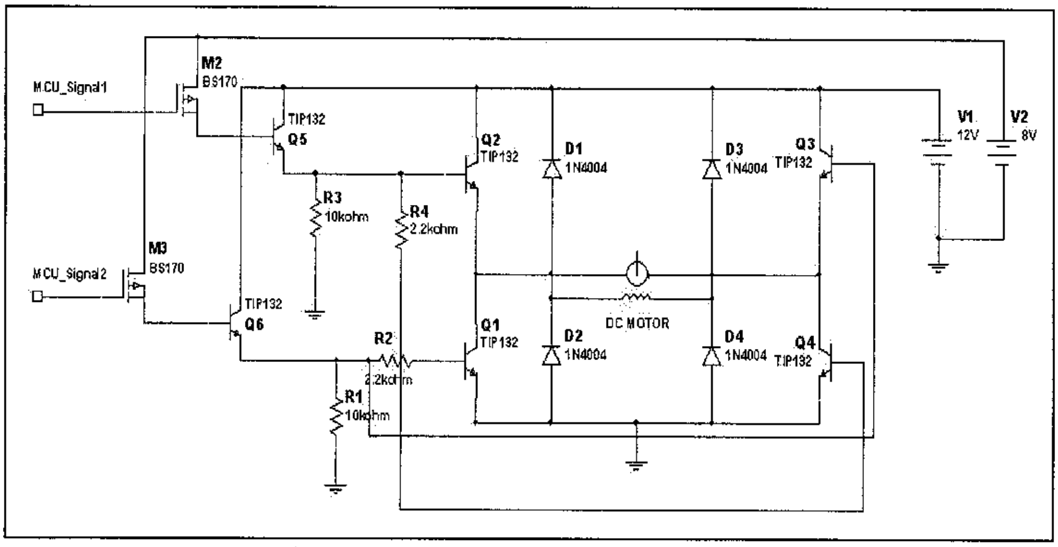


Figure 4.11 The Final Design of H-Bridge

For the implementation of this project, a specific h-bridge IC, L6203 manufactured by SGS-Thomson is also used. This kind of driver IC uses a technology called Multipower BCD which allows the integration on the same chip of isolated power DMOS elements, bipolar transistors and C-MOS logic. The maximum supply voltage that this IC can withstand is 60V, maximum current of 1.5A and total power dissipation of 1.5 W.

4. OPTICAL ENCODER

Optical encoder is one of the famous encoders since it is least expensive to buy and the easiest to build its parts. The encoder consist hollow disk or marked transparent disk and an optical sensor. Commercial encoders have segments in the range of 16 to 1024. The more segments we have, the finer the measurement is possible.

There are two ways to detect light and dark pattern on the disk which are transmission and reflection. In transmission type, the light will go through transparent segment and received by phototransistor or photodiode. On the other hand, in reflection type, the emitter and receiver will be placed on the same side of disk. The light will be reflected by white areas while dark areas will absorb the light.

In this project, transmission type is chosen. For transmission type disk, the pattern can be printed on clear plastic or heavy white paper with a laser printer. Then, the pattern is laminated to ensure its toughness and to avoid scratch in case of accidentally contact with the detector or environment.

By utilizing an encoder, the idea is this device will give the pulse for every angle of rotation. A pulse given by an encoder represents certain angle of rotation. Let say that the encoder disk has 64 segments (32 white and 32 black). The angle of one pulse is suggested by calculation below:

$$Angle\ per\ pulse = \frac{360°}{32\ segments} = 11.25°\ per\ pulse \tag{5}$$

The pulses generated by this encoder become an input to microcontroller. For every process, the microcontroller will remember the last pulse generated by encoder and will count the new pulse inputted from encoder. If the required angle is achieved, the microcontroller will send a signal to motor to stop the movement. Figure 4.12 below shows the circuit for encoder.

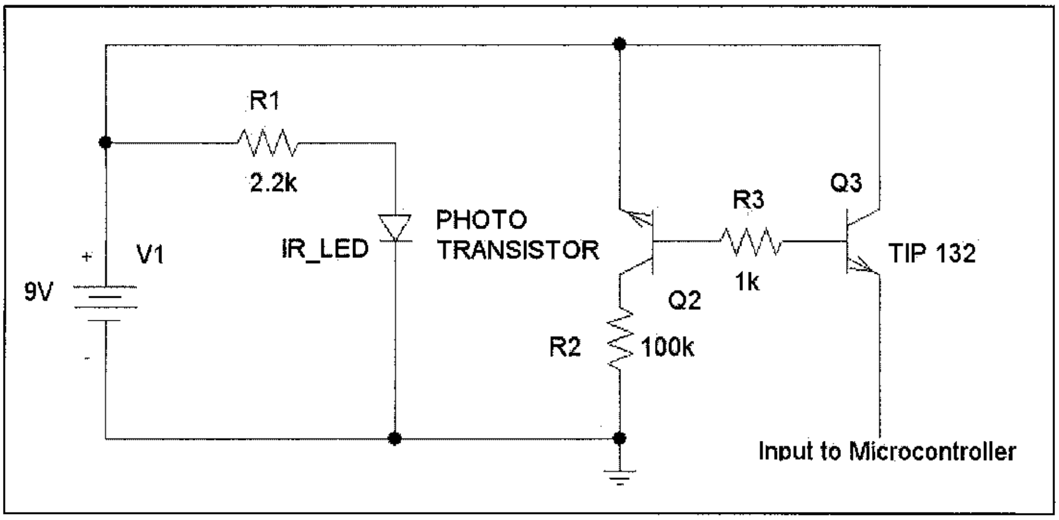


Figure 4.12 Encoder Circuit.

From the circuit above, the pulse could be obtained by placing encoder disk between infra red LED and phototransistor. This circuit only used for 12V 60 rpm dc motor since the motor is not originally equipped with encoder. Note that, the 12V 60 rpm dc motor is used for rotation at joint 3.

5. SELECTION OF MCU

There are various types of MCU in market in the manufacturing line of Microchip Technology Inc. First thing to know is there are lot of MCU manufacturers instead of Microchip Technology Inc such as Motorola, Intel, Atmel, Siemens and several other companies. The selections of MCU from production line of Microchip Technology Inc are based on its program or source code that easily understood, the MCUs that easily obtained and wide availability of its resources in terms of its program, source code and programmer. The type of MCUs is usually differentiated by their function, pin arrangement and application; PDIP, SOIP, SSOP, PLCC and TQFP.

The MCUs also are differentiated by their functions or applications such as EEPROM capacity, A/D conversion, serial communication, addressable universal asynchronous receiver transmitter (UASART) and pulse width modulation (PWM) function.

For controlling robot manipulator function, it requires serial communication function to communicate between PC and MCU. Furthermore, the function also requires PWM generated signal to drive the motor through h-bridge. Beside that, since the program for controller is in the form of receiving previous input, stores it and compare to the new generated input, it requires big EEPROM capacity.

Thus, based on these applications needed, PIC 18C452 and PIC 16F877 meet the above features and the selection is based on the availability of that microcontroller in market and selling price.

6. PICmicro 16F877

There are two types of Microchip microcontrollers that are very suitable for this project, namely PIC 18C452 and PIC 16F877. The suitability is measured to the features that owned by microcontrollers.

As discussed earlier, the controller project for robot manipulator requires at least the following functions; EEPROM capacity, A/D conversion, serial communication, addressable universal asynchronous receiver transmitter (UASART) and pulse width modulation (PWM) function. The PIC 18C452 and PIC 16F877 are said suitable for this project since it has the above-mentioned functions. The comparison between PIC 18C452 and PIC 16F877 is shown in specific tables below.

Table 4.1 Servo Calculation Bandwidth Comparison

Device	Operating Frequency	Hardware Multiplier	Servo Update Period	Maximum Servo Calculation Time	MCU Bandwidth Used
PIC16CXXX	20 MHz	No	563 μsec	540 μsec	96%
PIC18CXXX	20 MHz	Yes	563 μsec	97 μsec	17%

Table 4.2 Program Memory Resources for Servomotor Application

Device	Available Program Memory	Program Memory Used by Application	Percentage Used
PIC16F877	8192 x 14	3002 x 14	37%
PIC18C452	32768 x 8	8265 x 8	25%

Table 4.3 Data Memory Resources for Servomotor Application

Device	Total Data Memory Available	Data Memory Used by Application	Data Memory Used by Compiler(1)	Available Data Memory
PIC16F877	368 bytes	78 bytes	44 bytes	246 bytes
PIC18C452	1536 bytes	78 bytes	386 bytes	1072 bytes
Note 1: The amount of data memory will depend on the compiler used. This memory is used for a software stack, temporary variable storage, etc.				

The difference bandwidth used between PIC 16CXXX and PIC 18CXXX is quite significant, showing that PIC 18CXXX has greater bandwidth. From the table above, there is 17% of bandwidth used for PIC 18CXXX compared to 96% of bandwidth used by PIC 16CXXX. Again, for program memory resources, PIC 18C452 has lower percentage usage than PIC 16F877 with 25% and 37% respectively. From the last table above, it shows that PIC 18C452 has greater data memory resources. Thus, it is concluded that PIC 18C452 is better than PIC 16F877 in terms of its performance.

When come to the decision process, which one microcontroller to be chosen, the requirement might be expanded to certain extent, which another factors might be included. In this case, there are at least two criteria that are interrelated comprising of availability in market and cost. Based on these points, below are the comparisons of these microcontrollers. PIC 16F877 is easier to obtain in market compared to PIC 18C452. Furthermore, PIC 16F877 is cheaper than PIC 18C452 by the difference of price reaches RM 20.00 each. Thus, up to this point, PIC 16F877 is chosen.

The above paragraph discusses in general the requirements needed for the project. This paragraph, in turn will explain further about PIC 16F877 chip. PIC 16F877 is CMOS flash-based 8-bit microcontroller. It features 256 bytes of EEPROM data memory, self programming, an ICD, 8 channels of Analog-to-Digital (A/D) converter, 2 additional timers, 2 capture/compare/PWM functions, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI) or the 2 wire Integrated Circuit bus and a Universal Asynchronous Receiver Transmitter (USART).

PIC 16F877 comes together with 40 pin package. The pin layout chosen is PDIP. The microcontroller layout and pin assignment could be seen as in Figure 4.13 below. On the other hand, the key features of PIC 16F877 is shown in Table 4.4 compared to other PIC 16 series.

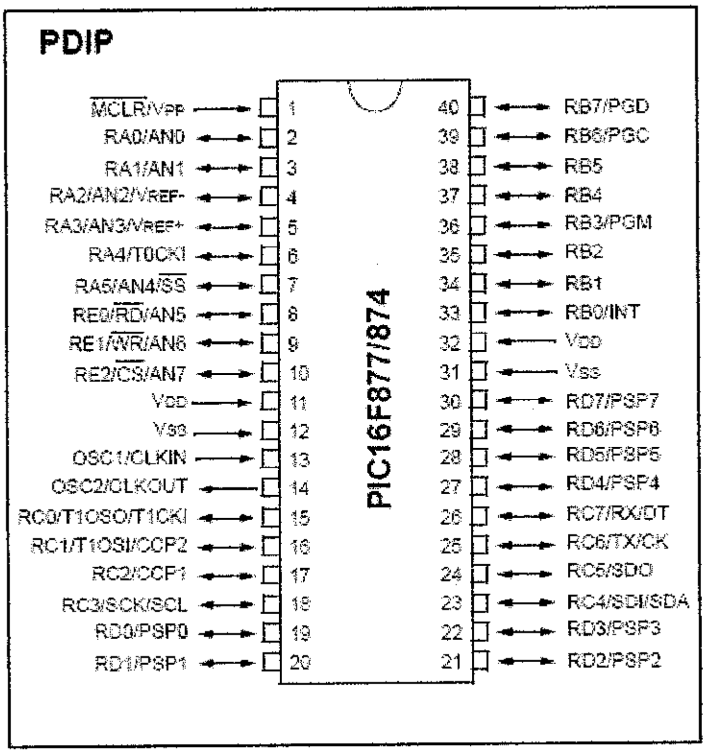


Figure 4.13 PIC 16F877 Pin Layout

Table 4.4 Key Features of PIC 16F877

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

7. DC MOTOR APPLICATION

The controlling of dc motor by setting PIC 16F877 as controller is based on AN 696 Application Note. In the design, it uses Pittman 9000 series motor. The programming of PIC 16F877 for dc motor application is based on this application note and modified so that it suites for our application. The major difference in the design is it uses Pittman 9000 series dc servomotor while this project uses RS dc motor, affects the programming especially in encoder part since RS motor has different type of encoder as Pittman uses.

The PIC 16F877 handles many functions in the servomotor application, such as:

- User control interface,
- Measurement of motor position,
- Generation of motor drive signal, and
- Communication with non-volatile EEPROM memory.

Three RS dc motors are chosen to work with this MCU. The motor is designed for 12 V and has no load speed of 2000 rpm. A schematic diagram for this application is shown in Figure 4.14 (full schematic diagram in Appendix H). The dc motors are driven by self-developed h-bridge with the combination of TIP 132 and BS 170 transistors.

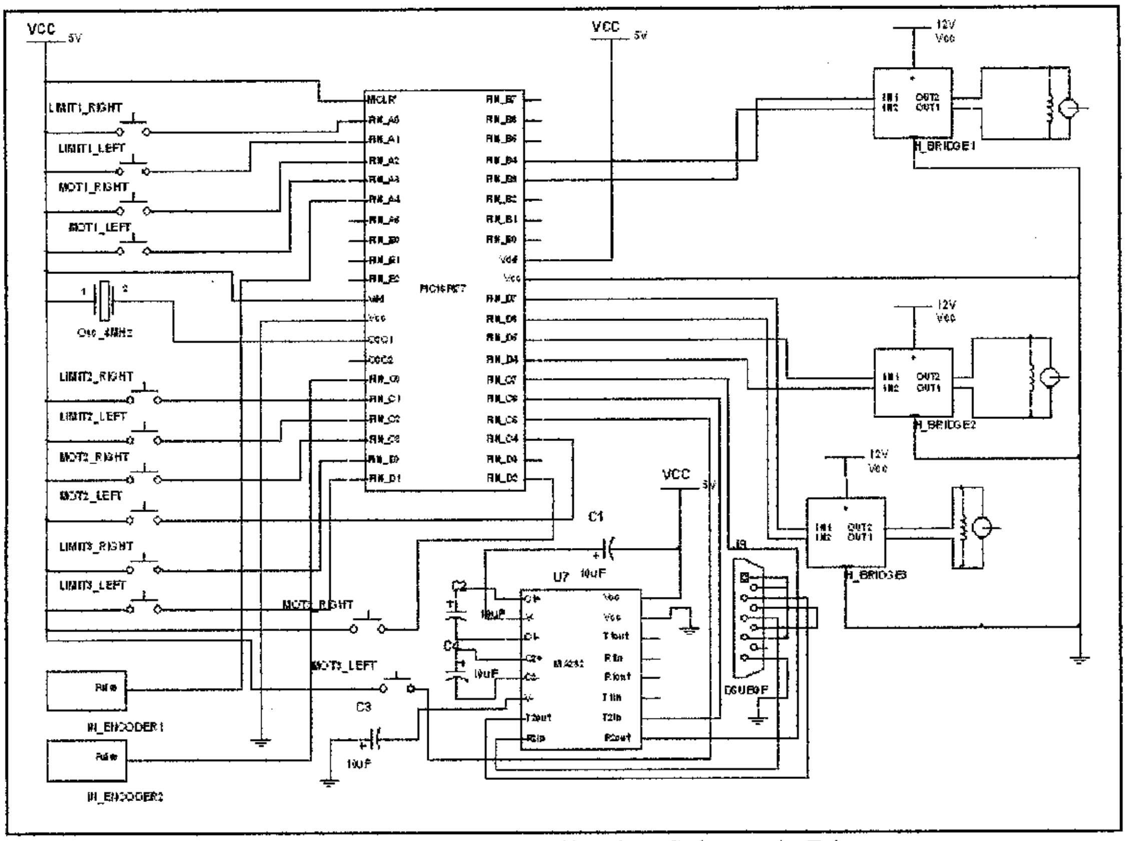


Figure 4.14 DC Motor Application Schematic Diagram

The movement or rotation of motors is controlled by several inputs, namely MOT1_RIGHT, MOT1_LEFT, MOT2_RIGHT, MOT2_LEFT, MOT3_RIGHT and MOT3_LEFT. These switches will give high input signal to the input pin of microcontroller once they are pushed. For example, as the MOT1_RIGHT switch is pushed, the motor will rotate in clockwise direction and vise versa when MOT1_LEFT switch is pushed.

The input from switch will give correspondence output to the h-bridge. IN1 in h-bridge circuit indicates motor to rotate clockwise. IN2 indicates the motor to rotate anti-clockwise. As the MOT_RIGHT switch is pushed, it will correspondently trigger high output signal at the output pin connected to IN1.

The function of LIMIT_RIGHT and LIMIT_LEFT switch is to stop the manipulator if maximum workspace has been achieved. At maximum position for each link, whether rightmost position and leftmost position, the manipulator will hit these limit switches and automatically the manipulator stops from moving.

Practically, when dc motor is rotating, it will generate pulse from encoder circuit. This pulse is very useful to determine how much the motor is rotating. The pulse is then sent to PIN_A4 and PIN_C0, so that the signal can be processed in such a way that how much degree the motor is rotating can be determined.

The primary user interface is a RS-232 connection between the MCU and host PC. A MAX232 IC is used in the design. The serial port in PC uses the voltages from the range of -10 V to 10V while IC chips use 0V to 5V. Thus, the idea of MAX 232 IC is to convert the voltage signal from -10V – 10V to 0V – 5V when communicating from PC to IC chips and vise versa.

The above paragraphs almost describe about the circuit or hardware of controller. For dc motor software, it performs position calculation and provides a command interpreter to create and control motion profiles. Dc motor position calculation is implemented to must perform the following tasks:

- Get current motor position, and
- Get desired motor position.

8. THE PROGRAMMING OF THE CONTROLLER

The general design is the system implements one PIC 16F877 microcontroller. The microcontroller receives variety of input from control console via various input pins. Then, the data is processed internally to generate the required output. The function of microcontroller also to communicate with PC through RS-232 serial communication.

8.1 Control Selection

The mode of operation for controller is designed in two states, namely MANUAL CONTROL and PC CONTROL. During MANUAL CONTROL state, the manipulator is controlled by using a console. As the manipulator moves, the degree of rotation is recorded and transmitted via RS-232 Tx pin, so that the manipulator position is displayed at HyperTerminal or specific user interface. The program only considers inputs from console and output into Tx pin and ignore whatever incoming input from PC.

As PC CONTROL is selected, the controlling of manipulator is done totally by PC. During this condition, all of RS-232 pins are used; Rx, Tx and Interrupt pin. During this state, the program only considers inputs from PC and output into Tx pin while ignoring whatever input coming from console. Figure 4.15 shows the flowchart of Control Selection function.

8.2 Limit Right and Limit Left Function

The limit switch function is designed for several purposes; first, to limit the movement of the manipulator in its workspace and second, for resetting function of the manipulator. This section only discusses the limit of the manipulator's movement while resetting function will be discussed in section 8.3.

From the above section, the workspace is determined as below:

$$\begin{aligned} -150° &\le \theta_1 \le +150° \\ -90° &\le \theta_2 \le +90° \\ -135° &\le \theta_3 \le +135° \end{aligned} \tag{6}$$

As joint 1 is moving, the manipulator will allow the rotation in the range of -150° to +150°. So, limit switch will play the function to stop the movement of the manipulator when maximum limit of workspace has been reached. The concept is same applied for all joints.

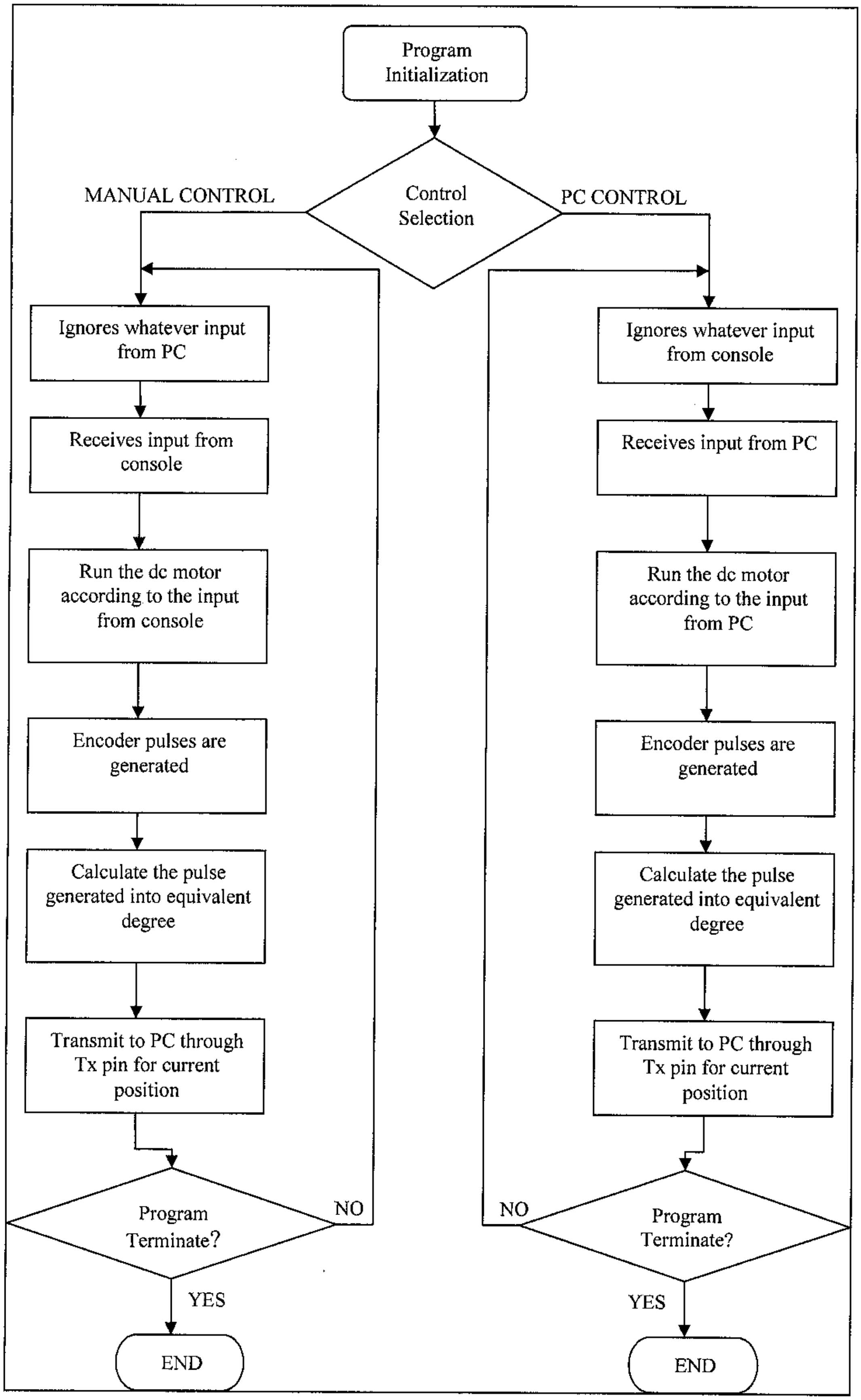


Figure 4.15 The Control Selection Function Flowchart

The idea is every link will be placed two limit switches, placed at both positive and negative maximum position. When one of those switches is hit, the switch will send signal to microcontroller to stop the movement. Figure 4.16 shows the flowchart for limit switch function.

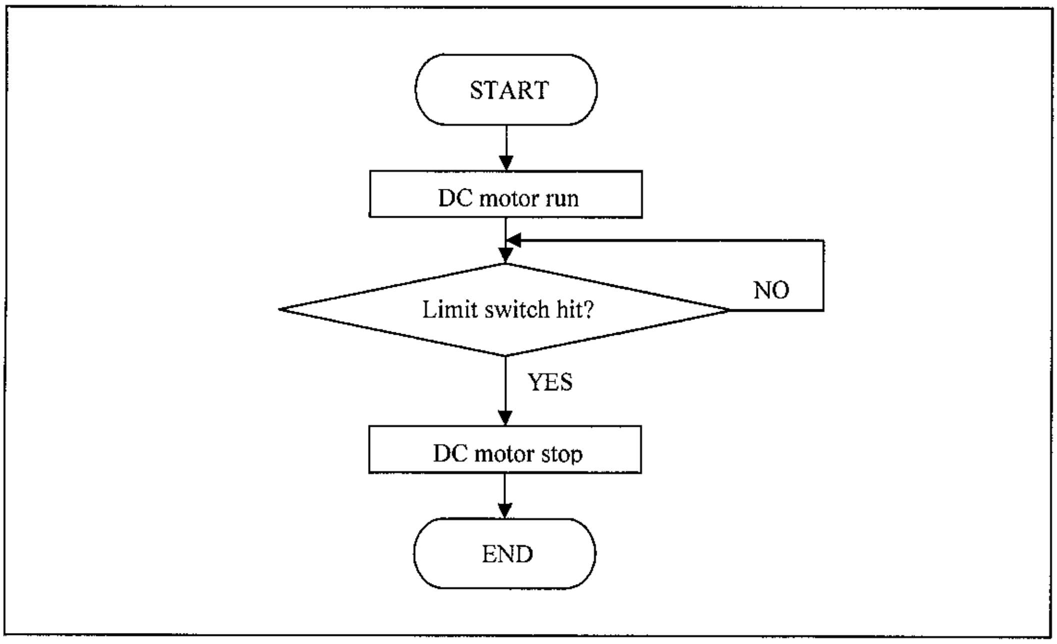


Figure 4.16 The Limit Switch Function Flowchart

8.3 Reset Function

The purpose of reset function is to reset all of the manipulator's operation and return back the manipulator to its original position. The concept is as the reset command is invoked, the manipulator will move to the rightmost or maximum position until the maximum limit switch is hit. Then, the dc motor will run on opposite direction until predefined pulse count is achieved. Now, the manipulator is said to be placed at its original position. Figure 4.17 below shows the flowchart of reset function for a link.

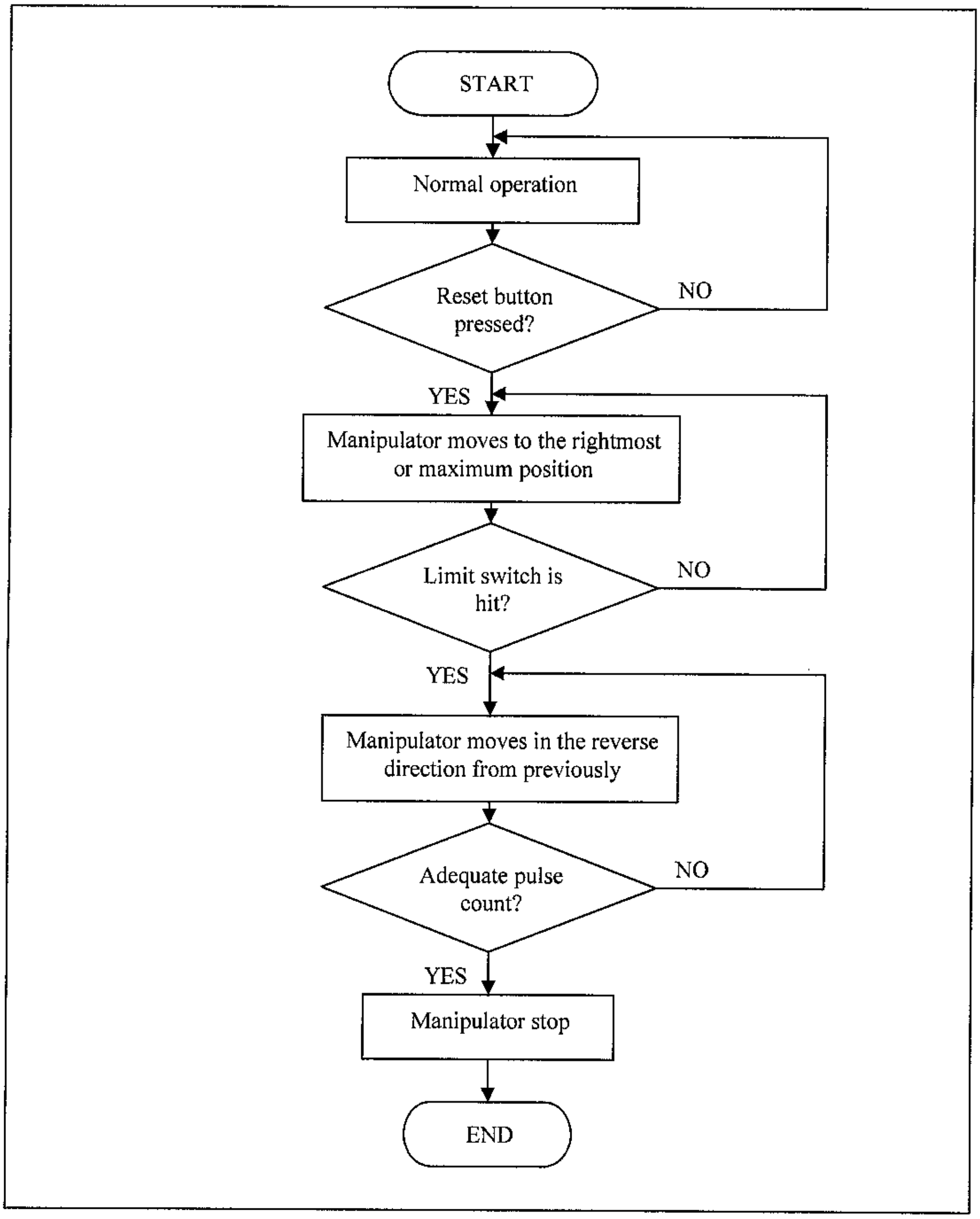


Figure 4.17 Reset Function Flowchart

8.4 Positioning Function

Positioning function is normal operation for the manipulator. The concept is as user instructs the manipulator either using console or PC, the program will do as what user wants. For example, let say the user want to move link 1 by 30°, the program receives the instruction and convert this 30° into equivalent pulse count. Then, the microcontroller signals the dc motor to rotate. The encoder pulse is generated and fed into another input pin of microcontroller. If the pulse count is adequate, the motor will stop, indicating that the link 1 has been moved by 30° of rotation. Figure 4.18 shows the flowchart for positioning function.

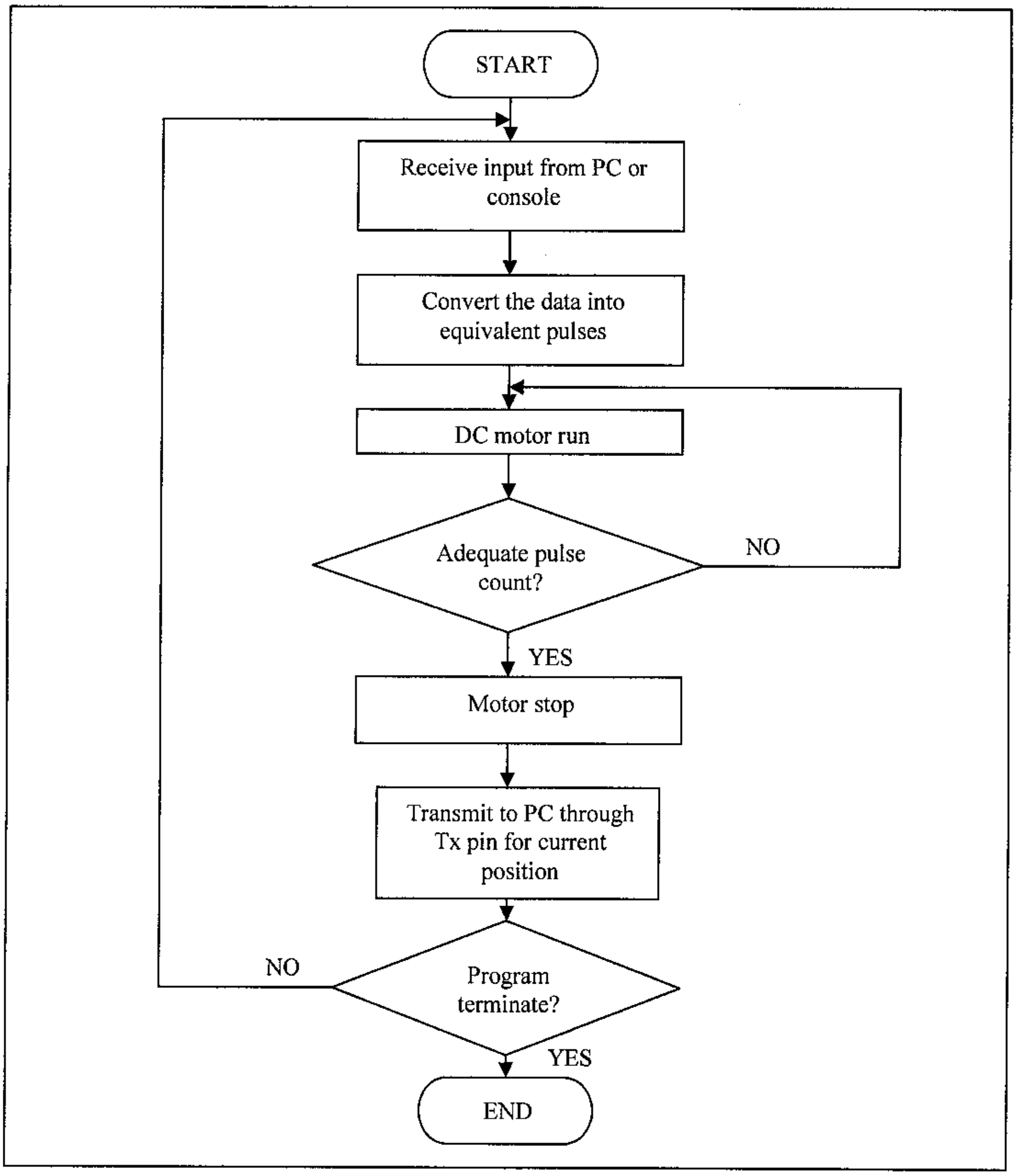


Figure 4.18 Positioning Function Flowchart

8.5 Emergency Function

The emergency function is purposely for halting immediately the operation of manipulator. There might be an unexpected occurrence during the operation, such as data floating or manipulator hitting person in workspace. Figure 4.19 shows the flowchart for emergency function.

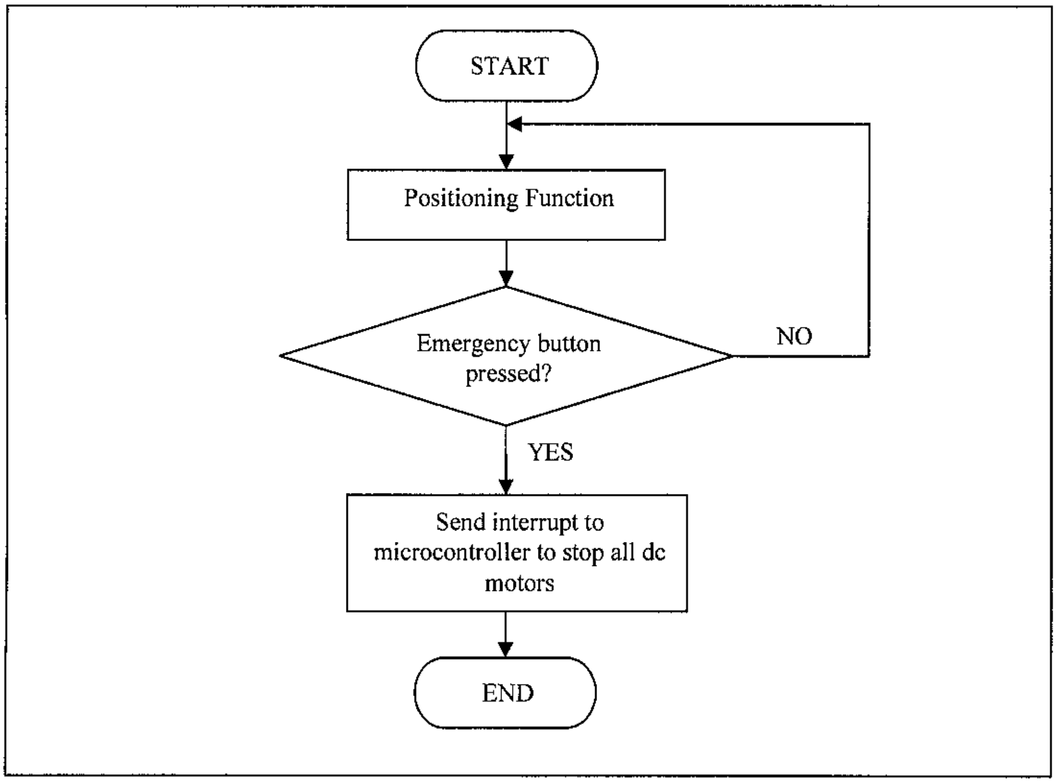


Figure 4.19 Emergency Function

9. DC MOTOR APPLICATION SOURCE CODE

The code implements a brush dc servomotor using PIC 16F877 microcontroller. The code also contains the process of controlling dc motor as discussed above.

Initially, the code will instruct the motor to rotate. After that, it receives encoder pulse signal from the motor for position calculation. Then, the result from position calculation will be transmitted to PC for displaying purpose through RS-232 serial communication.

This source code is compiled using PIC C Compiler version 2.7333 available at Microprocessor Lab. The generated source code is shown in Appendix I

10. USER INTERFACE

Instead of console, a HyperTerminal program is used to communicate between PC and microcontroller. HyperTerminal is used by applying the setting at COM1 to select serial port communication.

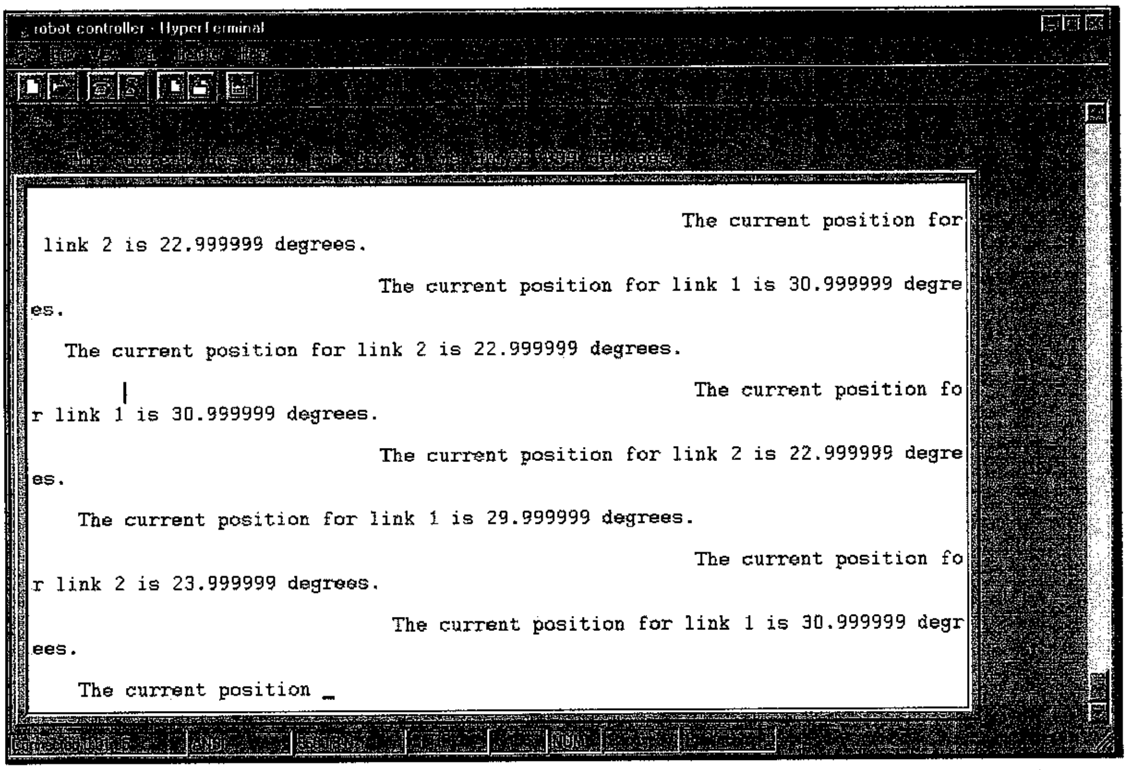


Figure 4.20 HyperTerminal Interface, Displaying the Current Position for The Manipulator: Link 1 is 30.99° While Link 2 is 22.99°.

CHAPTER 5

DISCUSSION

The development of robot controller and manipulator involves both electrical and mechanical skills. In general, the development of controller relates a lot with electrical skill while development of manipulator relates a lot with mechanical skill.

There are several aspects that play important role in determining the successful of manipulator design, which are stability, reliability, matching and accuracy.

The designed manipulator must be stable in the way that the base can support all three arms. And the lower arm is able to support correctly the upper arms. This condition can be achieved by implementing design carefully. The analysis is suggested to be more done at preliminary and detailed design stage by carefully analyze the full dimensional drawing. This is because any changes during design stage are easier to alter and involve lower cost compared in the construction or operation phase. If any of the design problems occur during construction phase, the design should be revisited and do necessary changes. Among the basic design strategies to ensure that the manipulator is stable are by allocating wide base area and lowering the centre of gravity at base.

Reliability in manipulator design means the manipulator can easily and properly operate. If the specified workspace is -90° to +100° (total 190°), the manipulator must achieve any position in this limit. The disturbance of achieving this limit might be improper arrangement of parts, for example improper placing of motor that disturb the movement of arm. This kind of disturbance can be eliminated by designing the parts' position properly including testing the model and improve the design if any problems arise. The manipulator also should move in such a way that having the lowest friction at joint and strong link or arm. This consideration can be achieved by implementing bearing at each joint and strong material is used. Note that, strong material is important to avoid links' deviation during its operation.

All parts in manipulator must be matched together to maximize its operability. The parts matching may include all parts are tightly placed at manipulator body, disallowing vibration between them. In case of vibration is allowed, it may disturb the rotation or revolution of manipulator. The most important thing is the gear must be matched in all of its application. These requirements may be achieved by properly applying screw or bolt and nut to tight the parts together and also use gears with same pitch only.

The final consideration in manipulator design is it must provide accuracy in positioning. The accuracy is very difficult to obtain since it involve a lot of criteria. But the most vital parts to achieve accuracy are the above-mentioned aspects that have been discussed earlier. Accuracy means how close for the manipulator to achieve the right position, which is if the controller sends the signal that want to go to coordinate (3, 4, 5), the manipulator can achieve that point. The most possible defector for the manipulator to achieve accuracy is the tightening and loosening process of belt during motor on and off. As the motor on, the belt is tightened while during motor off, the belt might loose. As the belt loose, the manipulator will slightly fall down from right coordinate after some manipulator's operation. This problem should be always reminded by applying belt tightening shaft for all belts used for the sake of minimizing this effect. This condition is also might be the advantage of using belt over chain since the belt can be vigorously tightened and chain might little loose during motor stop.

Another problem that might appear is during power off, which the motor also will off. Analytically, as the motor off, the motor is not holding arm anymore to a specific location. Thus, arm will fall down. If the power supply on, the arm then will up again. In order to remedy this problem, latching gearbox might be used to counter that effect, which latching gear box will restrict the gear or link to move during power off.

For controller, the determining aspects for the successful of controller are the goodness of microcontroller programming. The good programming will yield to the reliability of the program. The program should not mismatch or misinterpret between user interface (program at PC) and output signal established by microcontroller.

We could say that controller is a heart for manipulator's operation. Whatever things processed in heart is interpreted in the movement of manipulator. If the program has very little mismatch between the calculation of desired position and measured position, this difference will be accumulated throughout the manipulator's movement. Finally, let say after 50 movements or 60 movements, the error is quite significant. Thus, it is essential for that program to be equipped together with error calculation and compensation function.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

1. CONCLUSION

As the conclusion, the designing and fabricating process starts from study the existing design, followed by preliminary design, detailed design, construction and finally testing the constructed model.

At study stage, the study of robotic arms or robot manipulators of existing design is carried out. The functions are to find the idea to develop the design and setting up the best design parameters.

Then the process is followed by preliminary and detailed design. At this stage, the drawings and dimensions are established. The design also considers 3-DOF principle, selection of robot coordinate frame, workspace determination, forward and inverse kinematics and gearing design.

After the design is properly established and satisfied, the construction of robot manipulator is initiated. The construction process is started by selection of best material that posses light, strong and affordable characteristics. Then, construction process might be preceded by combining or joining all materials with specific dimension to be a robot manipulator. During construction stage, these skills are important such as cutting, drilling, grinding and combining the parts. During construction stage also, another activities are done such as developing microcontroller programming, developing the PC interface and generating relevant circuits.

The final stage is testing the manipulator. If there are any defects found, the design should be revisited, altered if necessary and executed on the model.

The controller designing stage is initiated by selection the best MCU in market. Then, the specification of this MCU must be understood for example its bandwidth, data memory capacity, program memory capacity and the most important things are pin layout and registers assignment.

The process of controller design is continued by understanding its program. There are two ways of developing source code, which are generated ourselves or alteration from the existing source code. Generating the source code ourselves consumes greater time than alteration the existing one. There might be nearly same source code in our resources, e.g book or internet and some alterations to suit our needs must be done.

After the program is generated and compiled with compiler results no error, the program is ready to be burned in PIC 16FF877. The program is testing again in real application, with our manipulator and the operation is observed. If the operation does not meet our design, the program or source code must be reviewed again.

2. RECOMMENDATION

Up to this stage, a robot manipulator and controller is completely designed and implemented. The application of robot manipulator covers only 3-DOF operation. It is restricted to the positioning function only.

It is recommended that for next stage, this robot manipulator is revamped with improved design and covers wide application, equipped together with manipulating function. Thus, the design will require high DOF, 6-DOF or higher. This is important to observe that better application could be implemented for example the process of lifting something, drawing, painting and several other suitable applications. The design also should be equipped with sensor system, for example it identifies which material to be picked up and more efficient user interface.

REFERENCES

[1] Lung-Wen Tsai, 1999, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Canada, John Wiley & Sons, Inc., pg 1.

[2] Gordon McComb, 2000, *The Robot Builder's Bonanza*, Martinsburg, McGraw-Hill, pg. ix.

[3] Saeed B. Niku, 2001, *Introduction to Robotics: Analysis, Systems, Applications*, New Jersey, Prentice Hall, Inc., pg. 2.

[4] Lung-Wen Tsai, 1999, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Canada, John Wiley & Sons, Inc., pg 2.

[5] Lung-Wen Tsai, 1999, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Canada, John Wiley & Sons, Inc., pg 46.

[6] Lung-Wen Tsai, 1999, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Canada, John Wiley & Sons, Inc., pg 25.

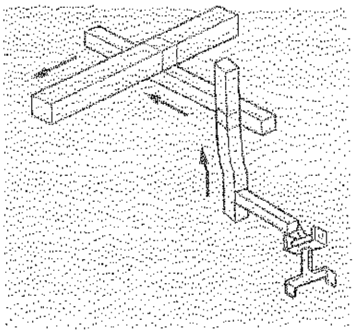
[7] Niku,, S., "Scheme for Active Positional Correction of Robot Arms," *Proceedings of the 5th International Conference on CAD/CAM, Robotics and Factories of Future*, Springer Verlag, pg. 590 – 593, 1991.

[8] Brady, M., J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*, MIT Press, Cambridge, Mass., 1982.

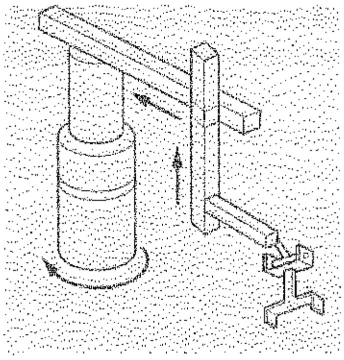
Appendix A

Types of Robot Coordinate Frame

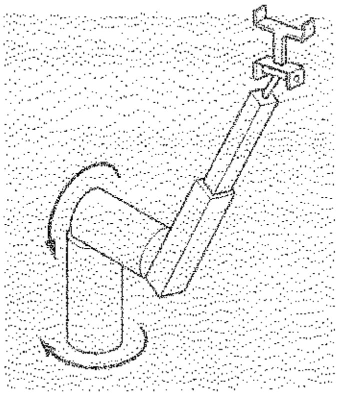
ROBOT COORDINATE FRAME



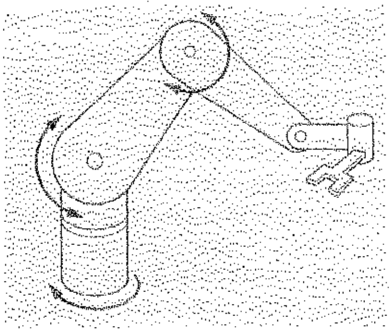
Cartesian
(x, y, z)



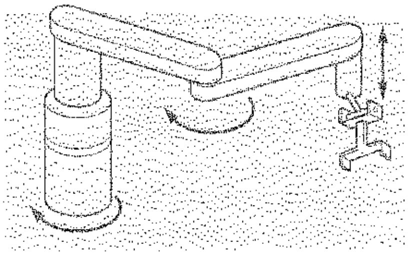
Cylindrical
(θ_1, y, z)



Spherical
(θ_1, θ_2)



Articulated
$(\theta_1, \theta_2, \theta_3)$



SCARA
(θ_1, θ_2, z)

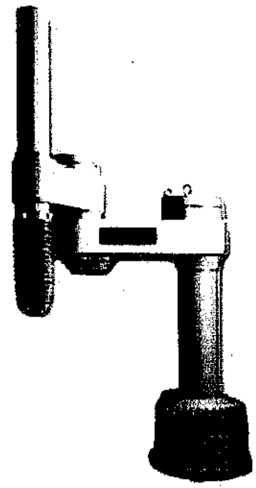
Appendix B

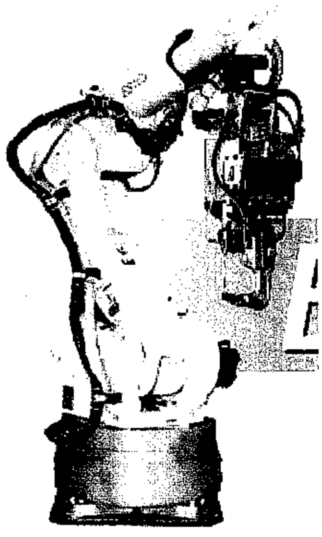
Types of Existing Robot Manipulator in Market

SAMPLE OF CURRENT ROBOTS IN MARKET

Kuka Robot from Kuka Roboter GmbH.

Adept-One Robot from Adept Technology, Inc.

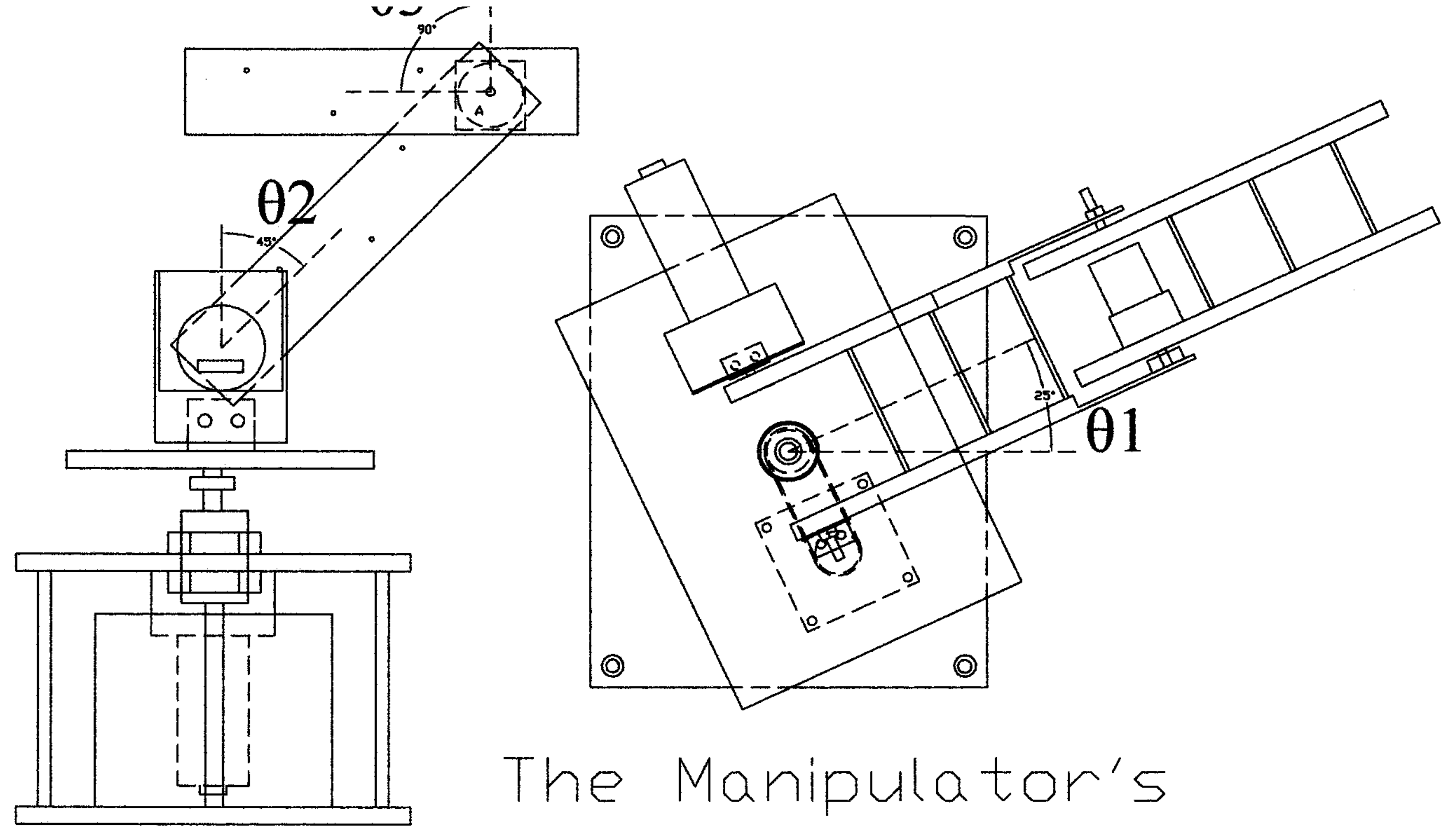




Motoman ES165 from Motoman Inc.

Appendix C

Revolution Joint Indication

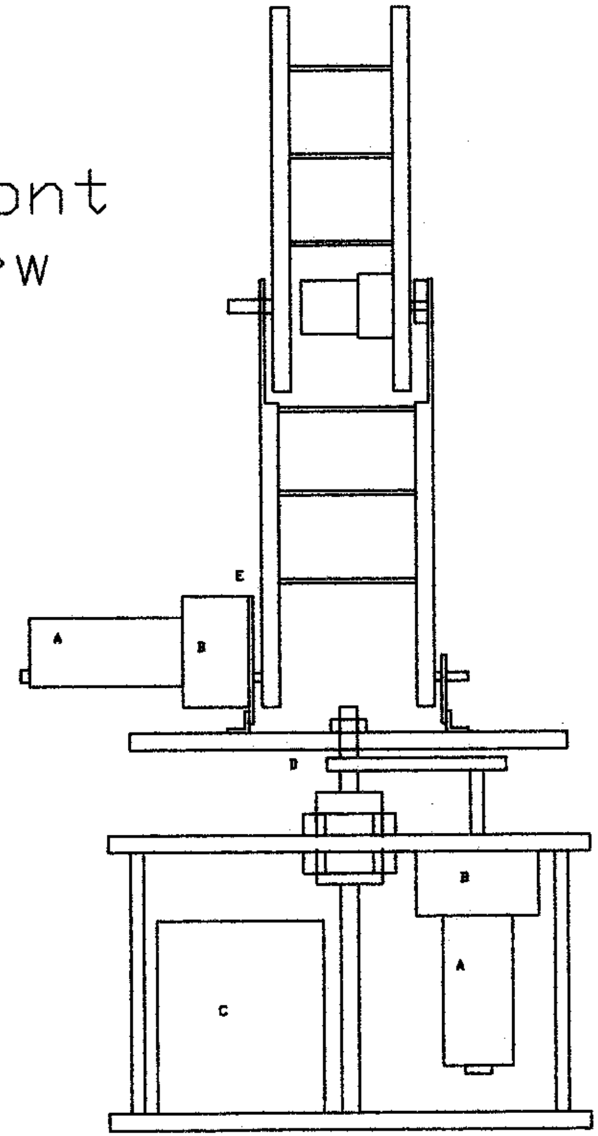


The Manipulator's
Angle Assignment

Appendix D

Detailed Drawing of Robot Manipulator

Front View

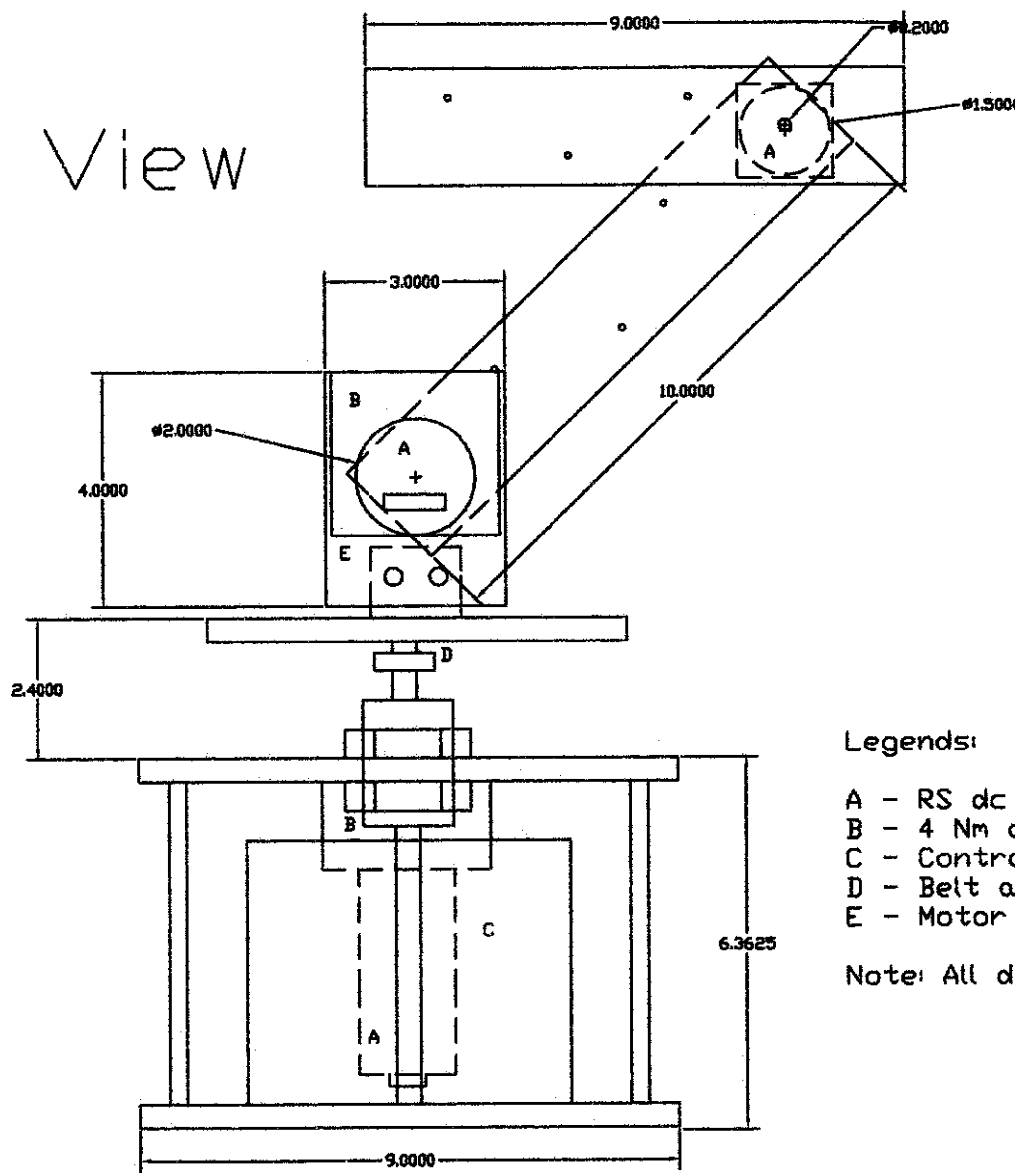


Legends:

A - RS dc motor with add-on encoder
B - 4 Nm adaptable gearbox
C - Controller box
D - Belt and gear
E - Motor mounting plate

Note: All dimensions are in inch

Side View

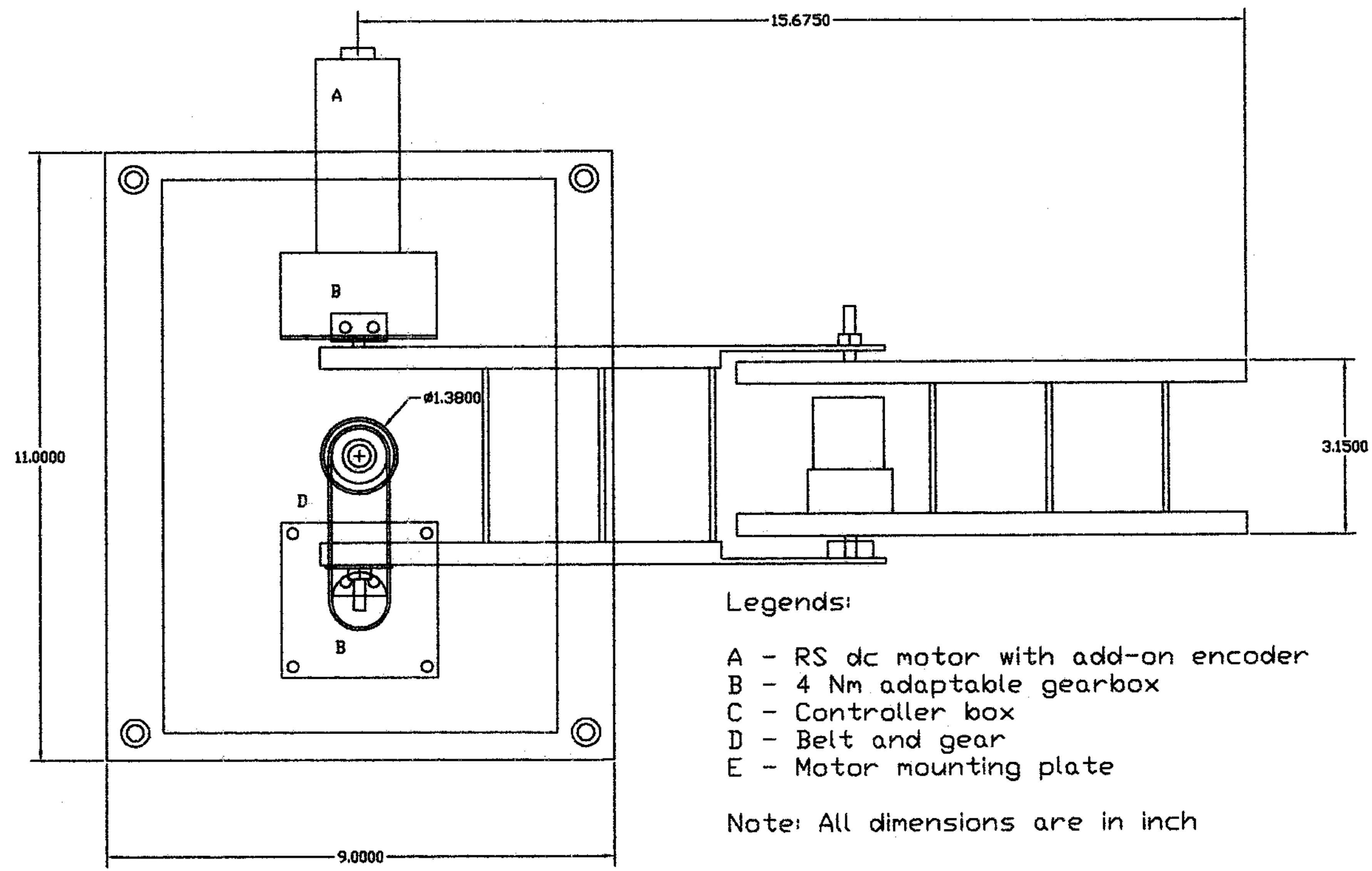


Legends:

A - RS dc motor with add-on encoder
B - 4 Nm adaptable gearbox
C - Controller box
D - Belt and gear
E - Motor mounting plate

Note: All dimensions are in inch

Top View



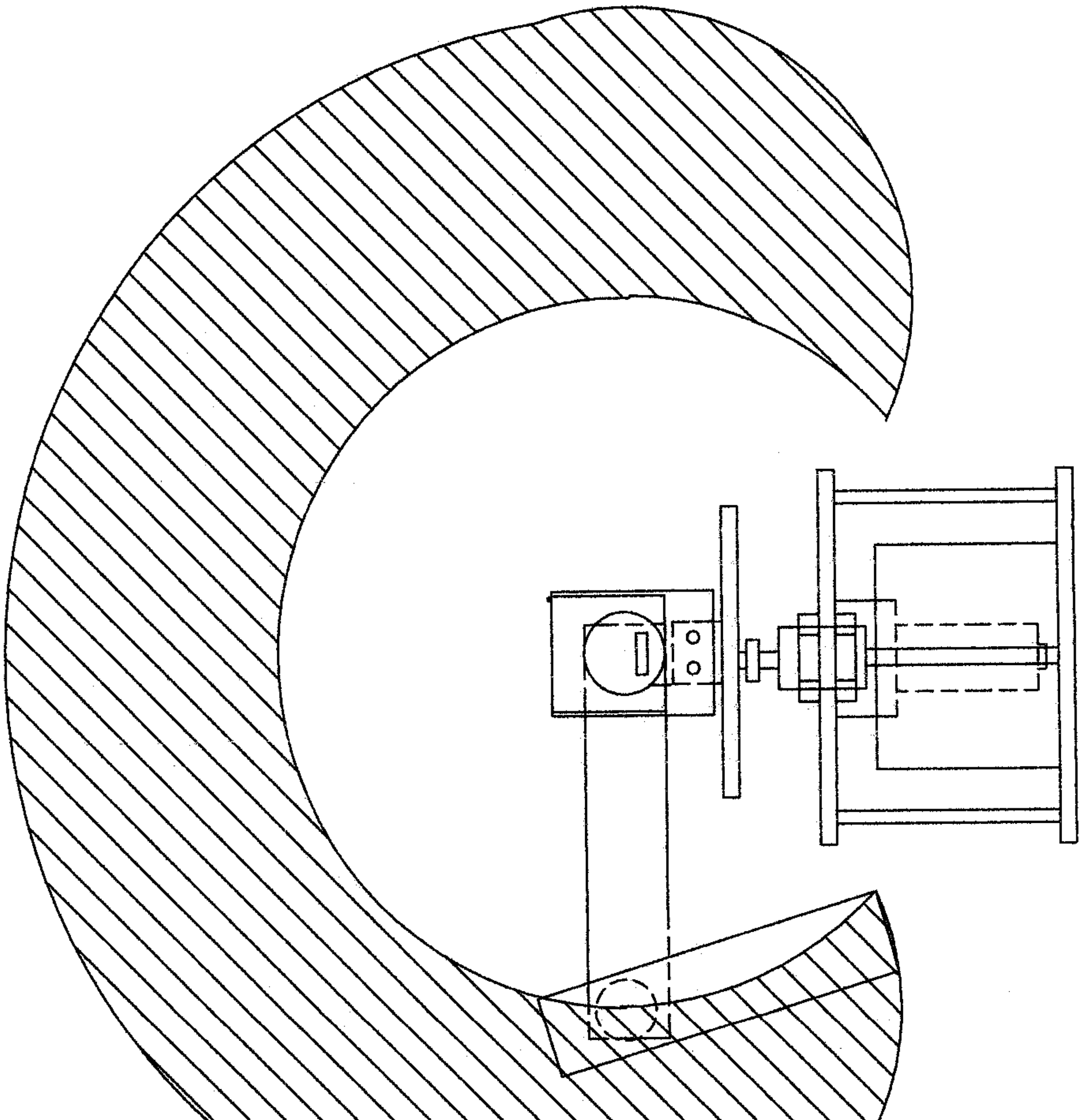
Legends:

A - RS dc motor with add-on encoder
B - 4 Nm adaptable gearbox
C - Controller box
D - Belt and gear
E - Motor mounting plate

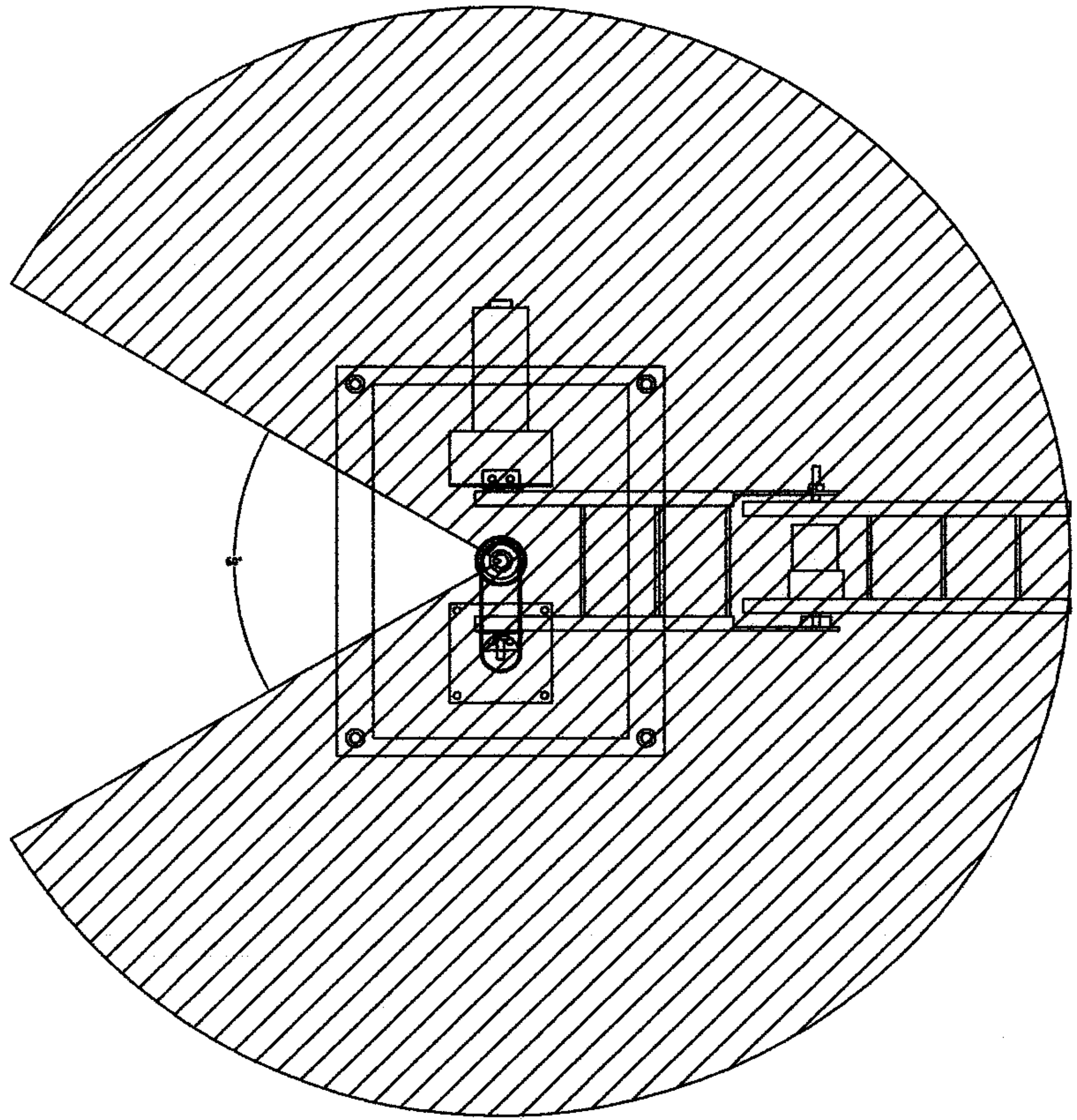
Note: All dimensions are in inch

Appendix E

Robot Manipulator's Workspace



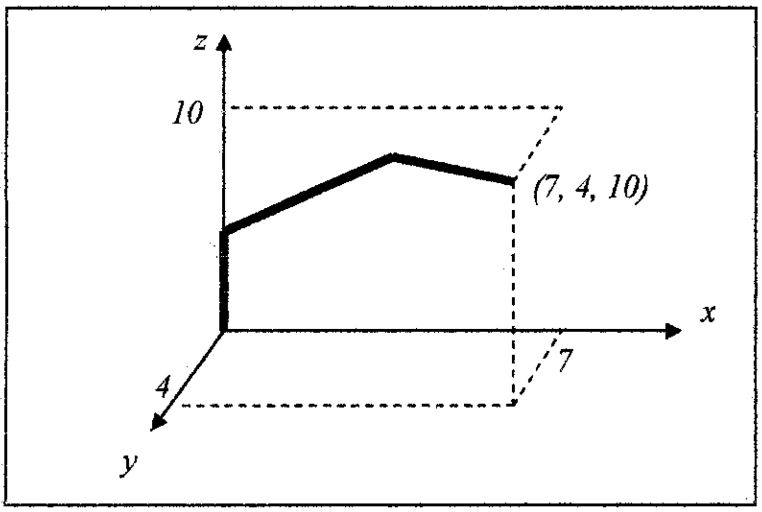
Top View Workspace



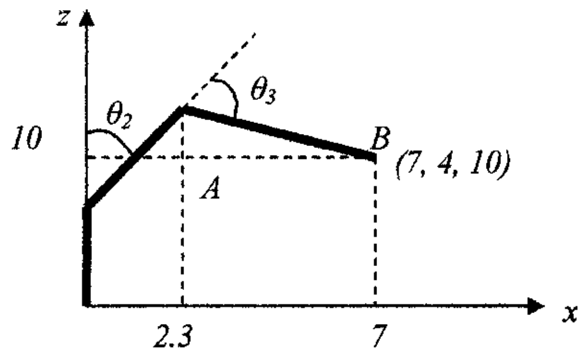
Appendix F

Detailed Calculation of Position Analysis

DETAILED CALCULATION OF POSITION ANALYSIS



Analyzing x- and z-axis



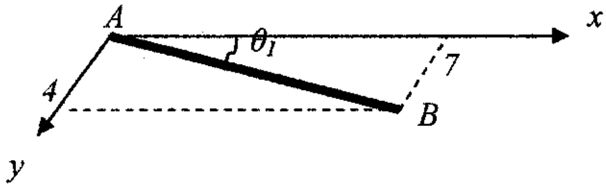
Setting up a parameter:

Length A-B = 4.7

$$\theta_2 = 90° - \cos^{-1}\left(\frac{2.3}{8}\right)$$
$$= 16.71°$$

$$\theta_3 = \cos^{-1}\left(\frac{2.3}{8}\right) + \left[90° - \sin^{-1}\left(\frac{4.7}{5}\right)\right]$$
$$= 93.24°$$

Analyzing x- and y-axis



$$\theta_1 = \tan^{-1}\left(\frac{4}{7}\right)$$
$$= 29.75°$$

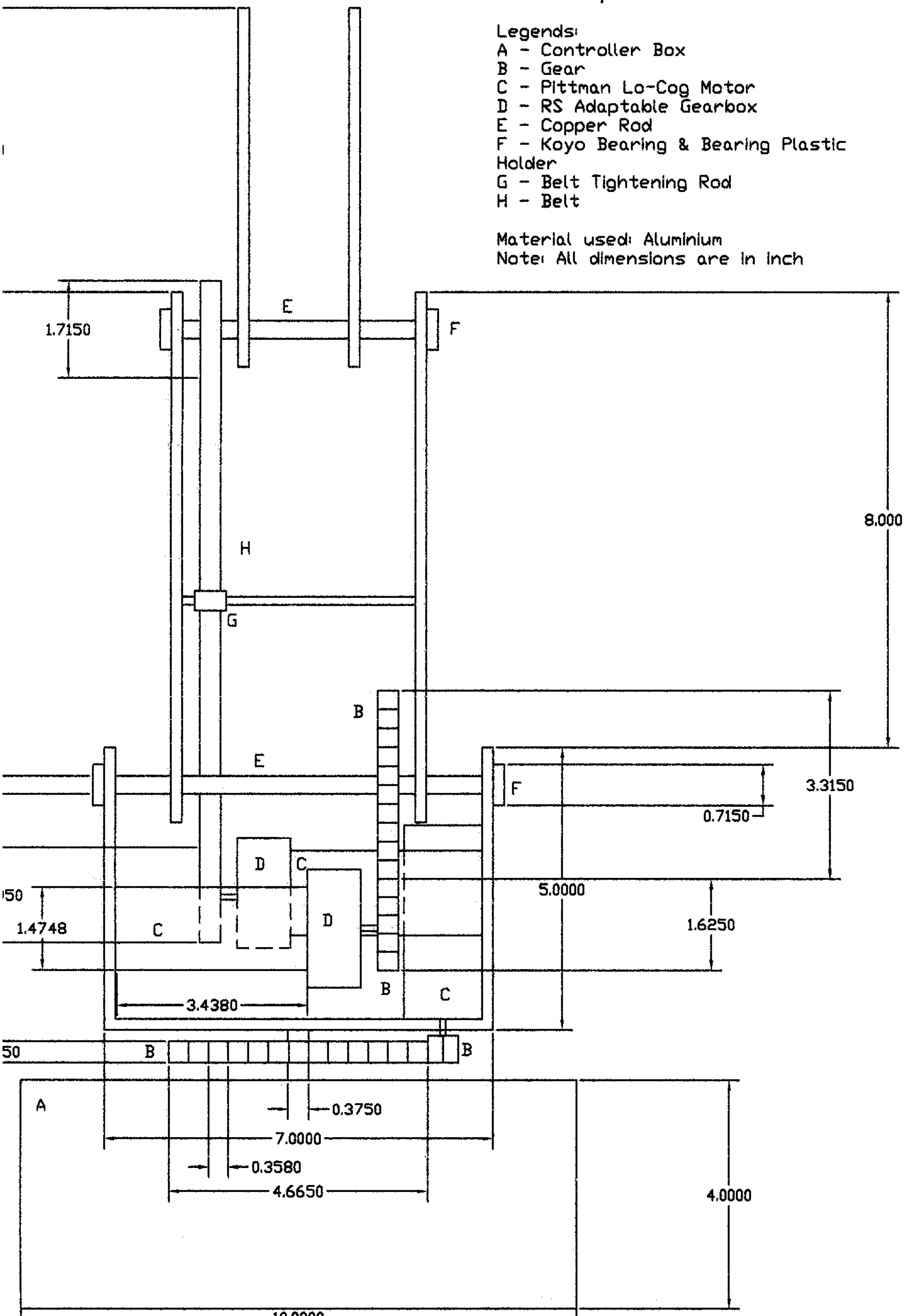
Appendix G

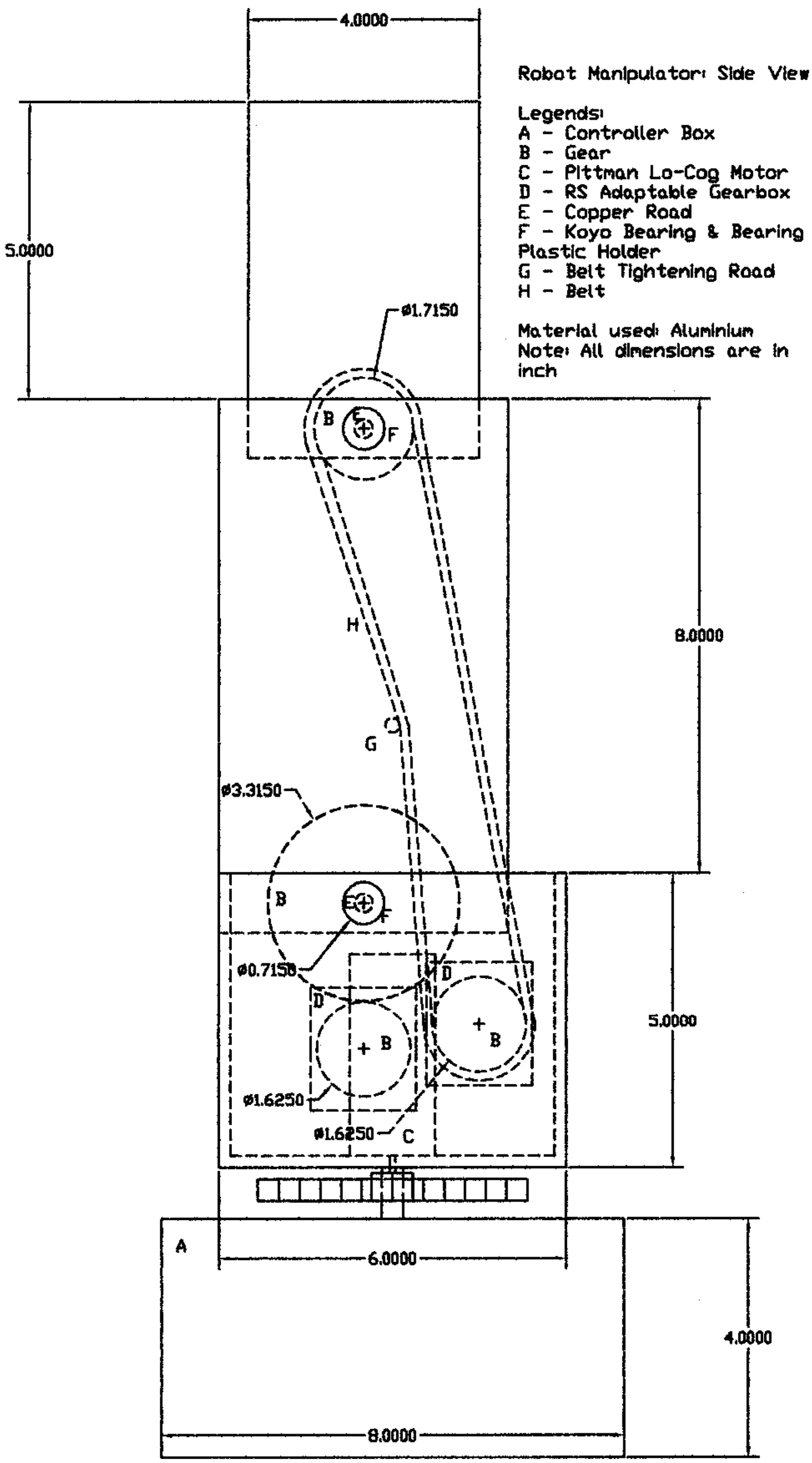
Detailed Drawing of First Manipulator Design

Robot Manipulator: Front View

Legends:
A - Controller Box
B - Gear
C - Pittman Lo-Cog Motor
D - RS Adaptable Gearbox
E - Copper Rod
F - Koyo Bearing & Bearing Plastic Holder
G - Belt Tightening Rod
H - Belt

Material used: Aluminium
Note: All dimensions are in inch




4.0000
5.0000
ø1.7150
B
E
F
H
G
8.0000
ø3.3150
B
E
F
ø0.7150
D
D
B
B
5.0000
ø1.6250
ø1.6250
C
A
6.0000
4.0000
8.0000
Robot Manipulator: Side View
Legends:
A - Controller Box
B - Gear
C - Pittman Lo-Cog Motor
D - RS Adaptable Gearbox
E - Copper Road
F - Koyo Bearing & Bearing Plastic Holder
G - Belt Tightening Road
H - Belt
Material used: Aluminium
Note: All dimensions are in inch

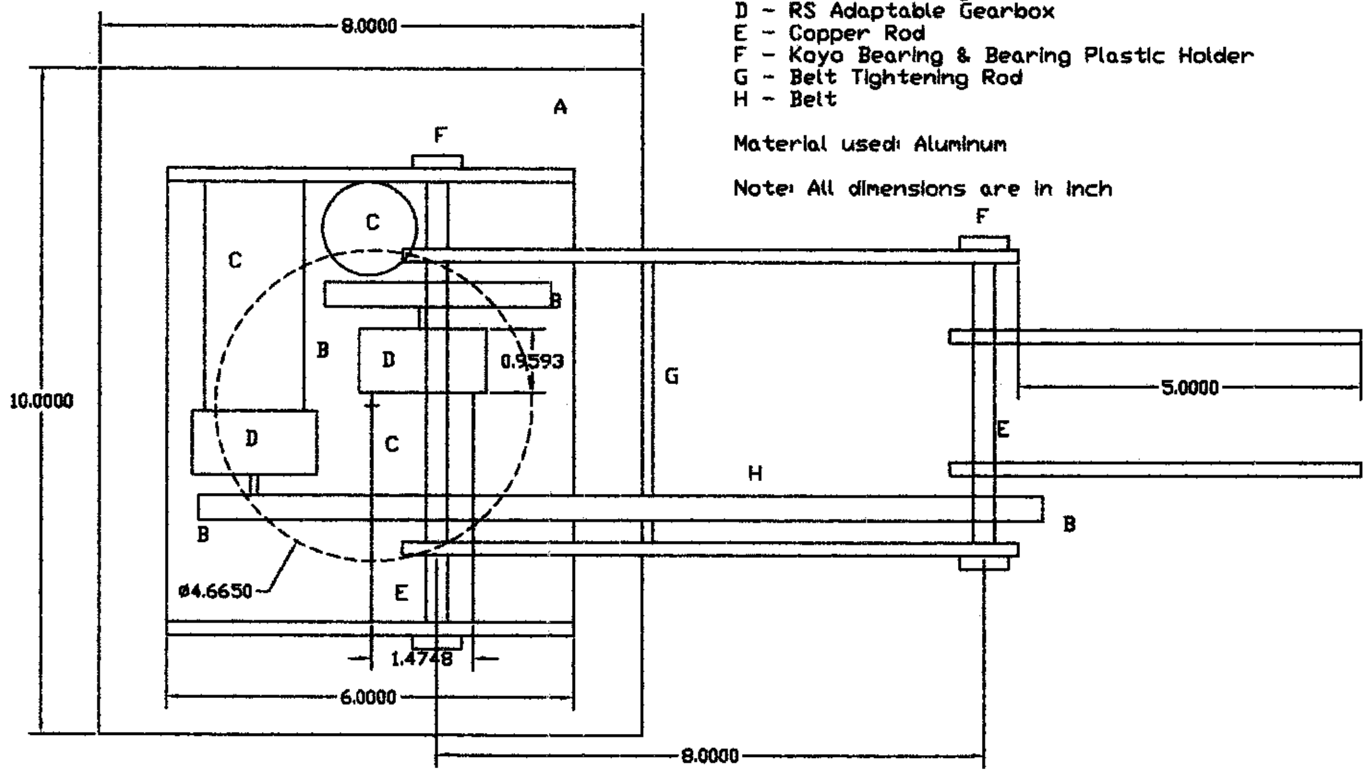
Robot Manipulator: Top View

Legends:

A - Controller Box
B - Gears
C - Pittman Lo-Cog Motor
D - RS Adaptable Gearbox
E - Copper Rod
F - Koyo Bearing & Bearing Plastic Holder
G - Belt Tightening Rod
H - Belt

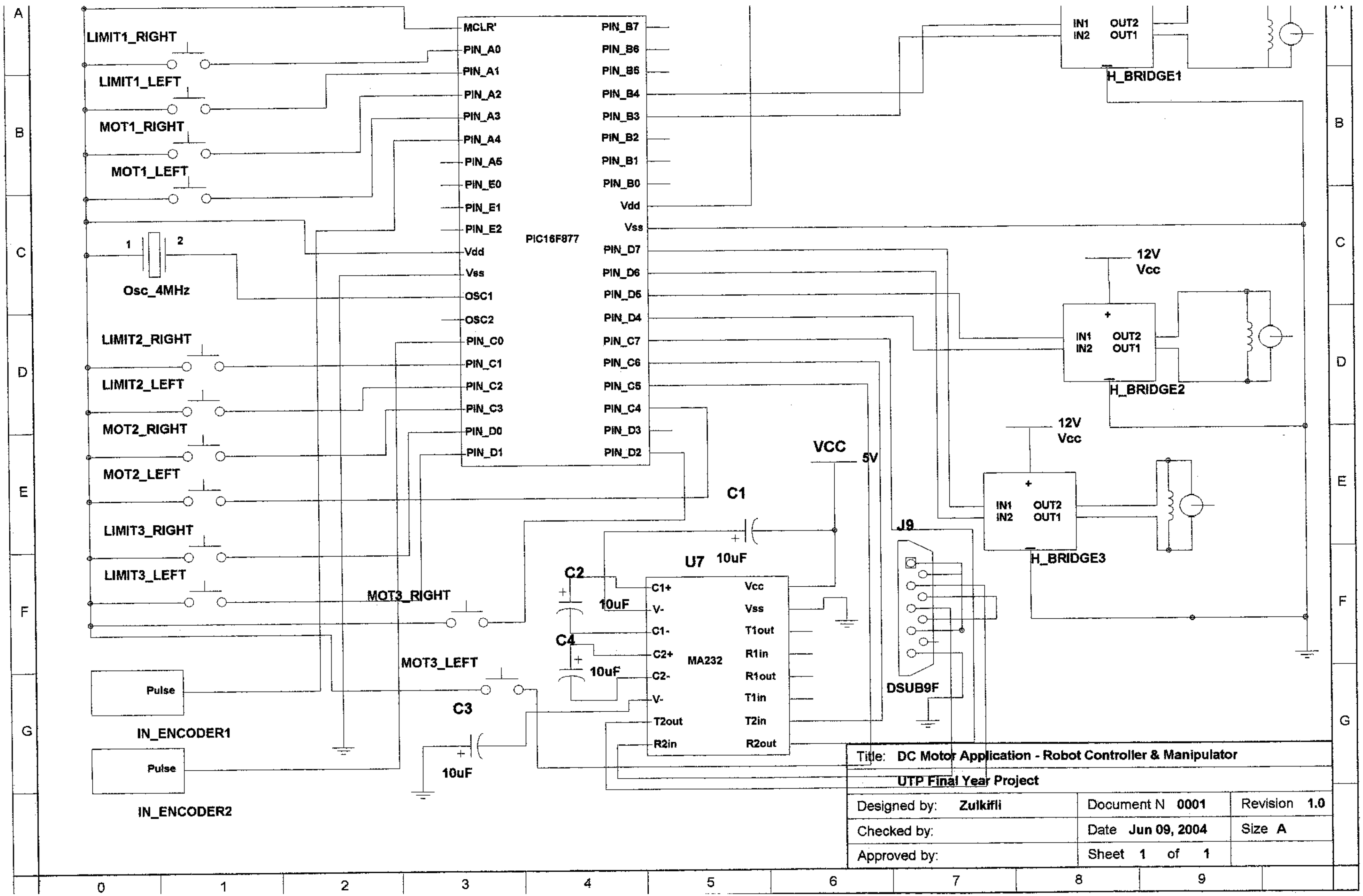
Material used: Aluminum

Note: All dimensions are in inch



Appendix H

DC Motor Application Schematic Diagram


LIMIT1_RIGHT
LIMIT1_LEFT
MOT1_RIGHT
MOT1_LEFT
Osc_4MHz
LIMIT2_RIGHT
LIMIT2_LEFT
MOT2_RIGHT
MOT2_LEFT
LIMIT3_RIGHT
LIMIT3_LEFT
MOT3_RIGHT
MOT3_LEFT
Pulse
IN_ENCODER1
Pulse
IN_ENCODER2
MCLR'
PIN_A0
PIN_A1
PIN_A2
PIN_A3
PIN_A4
PIN_A5
PIN_E0
PIN_E1
PIN_E2
Vdd
Vss
OSC1
OSC2
PIN_C0
PIN_C1
PIN_C2
PIN_C3
PIN_D0
PIN_D1
PIC16F877
PIN_B7
PIN_B6
PIN_B5
PIN_B4
PIN_B3
PIN_B2
PIN_B1
PIN_B0
Vdd
Vss
PIN_D7
PIN_D6
PIN_D5
PIN_D4
PIN_C7
PIN_C6
PIN_C5
PIN_C4
PIN_D3
PIN_D2
IN1 IN2 OUT2 OUT1
H_BRIDGE1
12V Vcc
IN1 IN2 OUT2 OUT1
H_BRIDGE2
12V Vcc
IN1 IN2 OUT2 OUT1
H_BRIDGE3
VCC 5V
C1
10uF
U7
C2
10uF
C4
10uF
C3
10uF
C1+
V-
C1-
C2+
C2-
V-
T2out
R2in
MA232
Vcc
Vss
T1out
R1in
R1out
T1in
T2in
R2out
J9
DSUB9F
Title: DC Motor Application - Robot Controller & Manipulator
UTP Final Year Project
Designed by: Zulkifli
Checked by:
Approved by:
Document N 0001
Date Jun 09, 2004
Sheet 1 of 1
Revision 1.0
Size A

Appendix I

Program Source Code

```
#include <16f877.h>
#fuses XT, NOPROTECT, NOWDT
#use delay(clock=4000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)

float movement_1();
float movement_2();
void reset();

//declaration of motor 1
#define LIMIT1_RIGHT PIN_A0
#define LIMIT1_LEFT  PIN_A1
#define MOT1_RIGHT   PIN_A2
#define MOT1_LEFT    PIN_A3
#define IN_ENCODER1  PIN_A4
#define H_BRIDGE1_1  PIN_B4
#define H_BRIDGE1_2  PIN_B5

//declaration of motor 2
#define LIMIT2_RIGHT PIN_C1
#define LIMIT2_LEFT  PIN_C2
#define MOT2_RIGHT   PIN_C3
#define MOT2_LEFT    PIN_C4
#define IN_ENCODER2  PIN_C0
#define H_BRIDGE2_1  PIN_D4
#define H_BRIDGE2_2  PIN_D5

//declaration of motor 3
#define MOT3_RIGHT   PIN_D2
#define MOT3_LEFT    PIN_C5
#define H_BRIDGE3_1  PIN_D6
#define H_BRIDGE3_2  PIN_D7

//general declaration
#define IN_RESET     PIN_A5

//main function---------------------------------------

main()
{
float position;

set_tris_a(0xff);
set_tris_b(0x00);
set_tris_c(0xff);
set_tris_d(0xf0);

printf("\nRobot Controller and Manipulator\n");
printf("\nCalculating the degree of rotation of robot manipulator\n");

// controlling motor 1
while (true)
{
if (MOT1_RIGHT)
   {
   if (!LIMIT1_RIGHT && !LIMIT1_LEFT)
      output_high(H_BRIDGE1_1);

   if (LIMIT1_RIGHT)
      output_low(H_BRIDGE1_1);

   if (LIMIT1_LEFT)
      output_high(H_BRIDGE1_1);
   }
else
   output_low(H_BRIDGE1_1);

if (MOT1_LEFT)
   {
   if (!LIMIT1_RIGHT && !LIMIT1_LEFT)
      output_high(H_BRIDGE1_2);

   if (LIMIT1_LEFT)
      output_low(H_BRIDGE1_2);

   if (LIMIT1_RIGHT)
      output_high(H_BRIDGE1_2);
```

```
    }
else
    output_low(H_BRIDGE1_2);

//controlling motor 2
if (MOT2_RIGHT)
    {
    if (!LIMIT2_RIGHT && !LIMIT2_LEFT)
        output_high(H_BRIDGE2_1);

    if (LIMIT2_RIGHT)
        output_low(H_BRIDGE2_1);

    if (LIMIT2_LEFT)
        output_high(H_BRIDGE2_1);
    }
else
    output_low(H_BRIDGE2_1);

if (MOT2_LEFT)
    {
    if (!LIMIT2_RIGHT && !LIMIT2_LEFT)
        output_high(H_BRIDGE2_2);

    if (LIMIT2_LEFT)
        output_low(H_BRIDGE2_2);

    if (LIMIT2_RIGHT)
        output_high(H_BRIDGE2_2);
    }
else
    output_low(H_BRIDGE2_2);

//controlling motor 3
if (MOT3_RIGHT)
    output_high(H_BRIDGE3_1);

if (MOT3_LEFT)
    output_high(H_BRIDGE3_2);

position_1=movement_1();
position_2=movement_2();

printf("\n The current position for link 1 is %f degrees.\n", position_1);
printf("\n The current position for link 2 is %f degrees.\n", position_2);

reset();
}
}

//Calculating the degree of rotation-------------------------

float movement_1()
{
float pos, count_pulse;

pos=0, count_pulse=0;

while (input(MOT1_RIGHT))
    {
        if (input(IN_ENCODER1))
        count_pulse = count_pulse + 1;
    }

while (input(MOT1_LEFT))
    {
        if (input(IN_ENCODER1))
        count_pulse = count_pulse - 1;
    }

    pos = count_pulse/2;                //2 pulses = 1 degree of rotation
    return pos;
}

float movement_2()
{
float pos, count_pulse;
```

```
pos=0, count_pulse=0;

while (input(MOT2_RIGHT))
   {
      if (input(IN_ENCODER2))
      count_pulse = count_pulse + 1;
   }

while (input(MOT2_LEFT))
   {
      if (input(IN_ENCODER2))
      count_pulse = count_pulse - 1;
   }

   pos = count_pulse/2;                //2 pulses = 1 degree of rotation
   return pos;
}

//Function for resetting----------------------------------

void reset(void)
{
int i, j;
while (IN_RESET)
   {
   if (!input(LIMIT1_RIGHT))
      output_high(H_BRIDGE1_1);

   for (i=300;i==0;i--)                //calculating the rotation pulse to initial state
      output_high(H_BRIDGE1_2);

   if (!input(LIMIT2_RIGHT))
      output_high(H_BRIDGE2_1);

   for (j=180;j==0;j--)
      output_high(H_BRIDGE2_2);
   }
}
```