ABSTRACT

HAND WRITING USING SCORBOT ARM

Jasvanth Satya Prabhakar Pakalapati, M.S.
Department of Electrical Engineering
Northern Illinois University, 2014
Peng-Yung Woo, Director

Robot arm is a robot manipulator with similar functions of a human arm. There are many research and industrial applications where robot arm is replacing humans. This thesis aims at analyzing and achieving the human-like behavior by robot arm manipulators. Writing skills would be a significant challenge in this research. This thesis aims at developing a robot arm which can help people in writing. A pen is mounted on the end of the robot's gripper and is programmed to write down the words that are instructed to it.

The patterns of handwriting are studied, and a strategy is developed to plan the trajectory of each letter. These paths are transformed into the vector form of joint angles and robot axis position in which order and speed of the movement can be developed to imitate human hand writing. The practicality of these methods is demonstrated by using the SCORBOT-ER-4u robot arm from Intelitek. It is a five-link arm similar to a human arm and has five degrees of freedom. It is a coordinate-based robotic system with encoder feedback. The system runs on syntax-based software SCORBASE, translated to robotic actuation by proprietary control hardware. Through this thesis, it has been proven that the algorithm developed can allow the robotic arm to write. This thesis could be a stepping stone for more opportunities in advanced robotic experiments at the Robotics and Intelligent Systems Laboratory.

NORTHERN ILLINOIS UNIVERSITY

DE KALB, ILLINOIS

DECEMBER 2014

HAND WRITING USING SCORBOT ARM

BY

JASVANTH SATYA PRABHAKAR PAKALAPATI

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

Thesis Director:

Dr. Peng-Yung Woo

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards all of those who helped me in successfully completing the thesis work. I would like to thank my advisor, Dr.Peng Yung Woo, for his guidance, support for procuring the required hardware, and his valuable suggestions and encouragement, which helped me complete this thesis within the stipulated time. Also, I would like to thank Dr. Donald S Zinger and Dr. Martin Kocanda for their valuable suggestions.

I would like to express my sincere thanks to Keith Mc, Doyle Maleche, Gerry O'Brien, Chris, most active members of the SCORBOT group on Yahoo! who helped me sort out mechanical and electrical issues related to the SCORBOT.

I would like to thank my family for their unconditional love, continuous support, enduring patience and inspiring words. Finally, I would like to thank my friends and everyone who has directly or indirectly helped me for their cooperation in completing the thesis.

DEDICATION

To my mom dad and brother, with love.

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview

Robots are designed to make human life better. Robots are replacing humans because they can work limitlessly and more efficiently. Today, universities, research institutes and companies are developing robots that can replicate functions as in humans. However, one of the primary functions that can be easily done by humans and is difficult for robots is writing. Currently, there are many ongoing research works that are targeted at making robots write. They can help physically disabled people in writing. The robotic arm can be compared to a human hand. It has a free rotating joint and a translational joint for the movement of the arm. An electric motor usually drives this arm movement. These actuators are controlled by a microcontroller or a microprocessor, usually programmable and made to perform a set of sequential tasks. These tasks are designed in a way that it can duplicate a human hand writing. This achievement is made possible by incorporating computational intelligence and integration of sensory feedback from the robot motors.

Many researchers have worked hard to study the mechanism and control of the robot arm to make it write. Michelman and Allen (1993) discuss how to control the compliant manipulation tasks with multi-fingered robots precisely. They focused their work on performing manipulation tasks on grasped objects such as using tools, writing, and sliding an object on a surface. Masui and Terano (1994) worked hard to make a robot write Kanji; a Japanese writing system adopted from logographic Chinese characters (“kanji,” 2014). They have used fuzzy logic system to achieve this design. The most essential features of Kanji are extracted and represented as fuzzy rules. Calligraphy requires very delicate operation of the writing brush. The results are effected sensitively by the touch, speed and control of the writing brush. Yossuf, Anuar and Fernandez (2005) developed an algorithm for robot writing using character segmentation process. The main idea is to store character information as segments, and the segment information can then be used by the robot to write. They applied this algorithm on a Mitsubishi RV-2AJ robot arm and made it write English characters and numbers.

1.2 Thesis Objective

In this thesis, an approach similar to the character segmentation algorithm Yossuf et al., is proposed. The core of the idea is to break down each character into small segments for which a robot path can be determined, calculate the position data for each of the small segments, and transform these data into vector form of robot joint angles and axis position. Also, this can design the optimal robot path for each character using these segments. These position data and robot path plan should provide sufficient information for the SCORBOT-ER-4u to write. Using the SCORBASE software, this data is uploaded into an individual position information table. The robot path planned for each character is now programmed into the controller. After analyzing the initial results, check for any errors and, if any, implement the respective countermeasures to compensate the errors.

1.3 A Note for the Reader and Research Successor

For the reader to get an overview of the work mentioned in this thesis, it is sufficient to read this document. Chapter 2 introduces the robot arm and its architecture. Chapter 3 gives information on the software used, and it explains how the robot arm is interfaced to the controller and how it can be programmed. Chapter 4 explains how the trajectory is planned for each character. Chapter 5 shows the results and analysis of the work done, and Chapter 6 concludes the thesis by mentioning limitations and future work.

A research successor who is interested in extending the thesis to control the robot arm with the existing setup or to change it as per his requirements and functionalities should go through additional sources mentioned below in addition to this document.

1. Sources from Intelitek (available in the lab)
   - SCORBOT-ER-4u User Manual: This explains the robot arm structure and specifications and gives details of its construction, components used (motors, encoders, microswitches, transmissions, and gripper), operation, workcell constraints, load capacities, etc. Access this manual from the following URL: http://elabz.com/wp-content/uploads/2010/04/ER4u_User_Manual.pdf
   - Controller-USB User Manual: This manual describes how to connect the robot arm to the controller. It describes the type of microprocessor or microcontroller embedded in the controller executing instructions interpreted by the software. Also, it illustrates the control mechanisms that the controller provides to control the robot arm through its architecture. In addition to these, it gives details of available connection interfaces on controller (like analog, digital, TTL, dry contacts, etc) to

connect external devices (like relays, motors, LEDs, buzzers, alarms, etc). Access this manual from the following URL: http://kurser.iha.dk/eit/i4prj4/ScorBot%20CD/Books/100341_D_Controller_USB(0305).PDF

- SCORBASE User Manual: This manual explains the SCORBASE installation on a computer and the connections to the Controller-USB for controlling the robot arm with the instructions, commands and their syntax. Access this manual from the following URL: http://intelitekdownloads.com/skills/skills_robot/scorbase-usb-v53.pdf
- RoboCell User Manual: This is a software from Intelitek for simulating a work environment (like objects to pick, workcell, positions, etc.) for the robot arm. This software can be integrated with SCORBASE to make the robot arm physically perform movements defined in the simulation. Access this manual from the following URL: http://www.intelitekdownloads.com/Manuals/Robots/100346-b-RoboCell-usb-v47(0210).pdf

2. Sources from Eshed Robotec (available in the lab)

- SCORBOT-ER-Vplus User's Manual: This is a similar robot arm from Eshed Robotec (now called Intelitek) older than the SCORBOT-ER-4u. This manual gives same details about the robot arm as mentioned in SCORBOT-ER-4u User Manual. Access this manual from the following URL:

  http://www.intelitekdownloads.com/Manuals/Robots/ER_V_plus_manual_100016.pdf

- Controller-A User's Manual: This is a controller similar to the Controller-USB. The computer should be connected to this over a serial communication line instead of USB like for Controller-USB. It also explains about the controller architecture similar to the Controller-USB User Manual. Access this manual from the following URL: http://www.intelitekdownloads.com/Manuals/Robots/ER_V_plus_manual_100016.pdf
- ACL User's Manual: This is DOS-based software older than SCORBASE for SCORBOT-ER-Vplus. The ACL (Advanced Control Language) manual explains set of commands and their syntax to control the robot arm. Access this manual from the following URL: http://elabz.com/wp-content/uploads/2010/04/100083-a-ACLV1.43-F44-Ctrl-A.pdf
- SCORBASE User's Manual: This manual explains that this software uses ACL commands to program and control the robot arm. This is not similar to the SCORBASE User Manual provided by Intelitek. Access this manual from the following URL: http://www.theoldrobots.com/book45/scorbase5.pdf
- RoboCell User's Manual: This manual is similar to RoboCell User Manual from Intelitek. Access this manual from the following URL: http://www.intelitekdownloads.com/Manuals/Robots/100166-f-RoboCell-24pc579-v3x.pdf

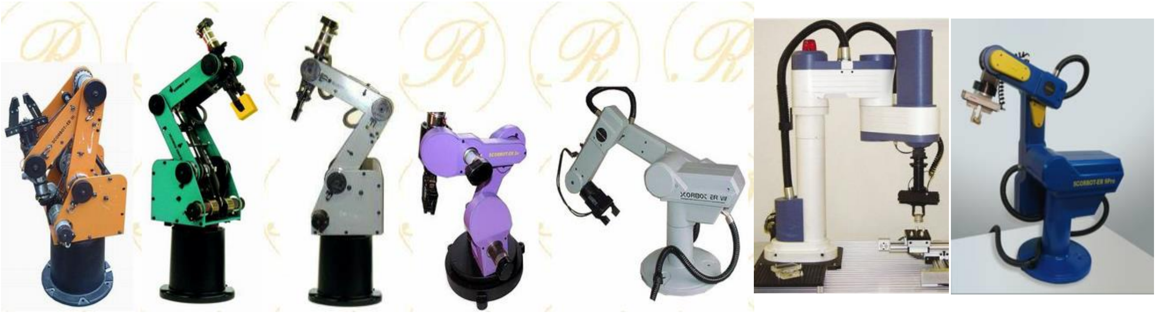
CHAPTER 2

THE ROBOT ARM

2.1 SCORBOT-ER-4u

SCORBOT is a family of robotic manipulators from Eshed Robotec, now called Intelitek. These robotic arms are equipped with a microprocessor or microcontroller-based controller, depending on the type of the robot. The controller when interfaced with a computer installed with the software can receive a set of commands or instructions and communicate with the robot using different channels, like serial, parallel, USB, etc., depending on the controller type, and send signals to activate the robot motors accordingly. Figure 1 shows some of the SCORBOT types available and some of them are listed below. The SCORBOT-ER-4u is used in this thesis.

- SCORBOT-ER-III is shown in Figure 1 (i).
- SCORBOT-ER-4u is shown in Figure 1 (ii).
- SCORBOT-ER-5 is shown in Figure 1 (iii).
- SCORBOT-ER-2 is shown in Figure 1 (iv).
- SCORBOT-ER-VII is shown in Figure 1 (v).
- SCORBOT-ER I4 is shown in Figure 1 (vi).
- SCORBOT-ER 9 Pro is shown in Figure 1 (vii).



(i) (ii) (iii) (iv) (v) (vi) (vii)

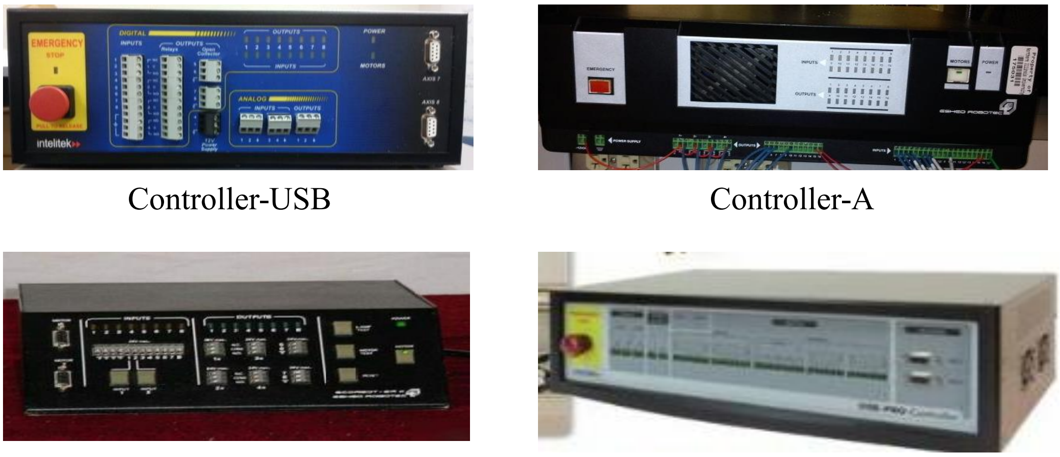
Figure 1: Some of the types of SCORBOTs available (The Scorbot Robotic Arms, 2010).

The complete SCORBOT package consists of the following parts (Eshed Robotec, 1988):

- The Robot Arm
- Controller
- Software
- Teach Pendant (optional)
- Conveyers (optional)
- Additional Motor Axes (optional)

2.2 Controller and Software

Intelitek ships the SCORBOT with its controller and software depending on the type of robot arms. Figure 2 shows different types of controllers available to control the SCORBOTs. Table 1 provides information on software type, environment and functions that come in the package. Table 2 provides information on some of the controllers compatible with some of the SCORBOT types available.



Controller-USB

Controller-A

Controller-B

Controller-USB PRO

Figure 2: Different types of controllers for SCORBOT.

Table 1

Software and Its Functions Available for the SCORBOTs (Intelitek 2005, 2006)

Software	Environment	Functions
ACL (Advanced Control Language) for SCORBOT-ER V plus	DOS-based program editor environment available till Windows 98	Uses ACL commands to control the robot axes. All the commands have to be sent through an ATS (Advanced Terminal Software) editor.
SCORBASE for SCORBOT-ER V plus, SCORBOT-ER-4u	DOS-based or an IDE (integrated development environment)-like environment.	Either uses ACL commands or high-level commands and compatible with high-level languages like C++, VisualBasic to control the robot axes.
RoboCell for SCORBOT-ER V plus, SCORBOT-ER-4u	3D simulation environment	An interactive graphic software package which provides simulation of robot and other devices in the workcell.

Table 2

Some SCORBOT and Controller Types Available

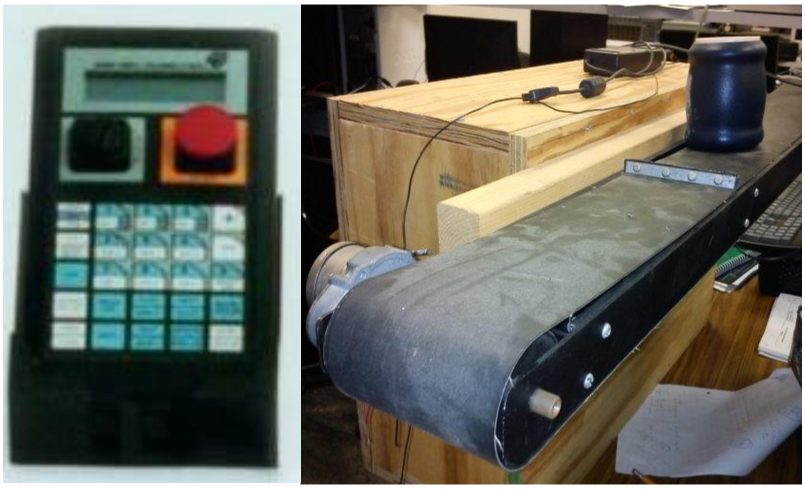
SCORBOT Type	Controller Type	Microprocessor or Microcontroller
SCORBOT-ER-4u	Controller-USB	NEC V853
SCORBOT-ER V plus	Controller-A	Motorola 68010
SCORBOT-ER 4PC	Controller-B	Motorola 68020 Motorola FPU – MC68881
SCORBOT-ER 9 Pro SCORBOT-ER 14 Pro	Controller-USB Pro	ARM7 TDMI

2.3 Teach Pendant, Conveyer, Additional Motor Axes

These are optional equipment that comes with the robot arm either to control the robot arm or to create additional working environment to the robot workcell. Each one is briefed up below.

- A teach pendant shown in Figure 3 controls the robot motors in manual mode and is used instantaneously to check robot motors and the joint and link movements.
- The conveyor shown in Figure 3 is driven with additional motor axes interfaced to the controller to create an industrial environment for the robot arm to pick or place objects.

- Additional motor axes can be interfaced to the controller to drive external devices through the software.



Teach Pendant Conveyer

Figure 3: Teach pendant and conveyor.

2.4 Structure

The SCORBOT-ER-4u shown in Figure 4 is a versatile and reliable system for industrial robotics training and education. The Scorbot-ER4u robot arm can be mounted on a tabletop, pedestal or linear slidebase. It has five revolute joints. With gripper attached, the robot has six degrees of freedom. Figure 5 shows the different parts on the robot arm. Its body is divided into links and joints as shown in Figures 6 and 7, with gripper attached to the end of the forearm, and their movements are described in Table 3. For moving the links and joints of the robot, DC motors, different types of arrangements involving gears, timing belts, pulleys and lead screw are used which are explained in detail in the following sections of this chapter. Encoders fixed on the motors and microswitches provide feedback to find the position, angle and speed of the links and joints within the robot's workspace. Table 4 provides complete specification details regarding SCORBOT-ER-4u. The main important parts of robot are shown below (Eshed Robotec, 1988; Intelitek, 2001).

- Motors
- Encoders
- Microswitch
- Transmissions
- Gripper

Table 3

Joints and Its Movements of SCORBOT-ER-4u (Intelitek, 2001)

Axis	Joint Name	Motion	Motor No.
1	Base	Rotates Body	1
2	Shoulder	Raises and Lowers the Upper Arm	2
3	Elbow	Raises and lowers the Forearm	3
4	Wrist Pitch	Raises and lowers the end effector (gripper)	4 + 5
5	Wrist Roll	Rotates the end effector (gripper)	4 + 5
Gripper	Gripper	Opens and closes the end effector (gripper)	6

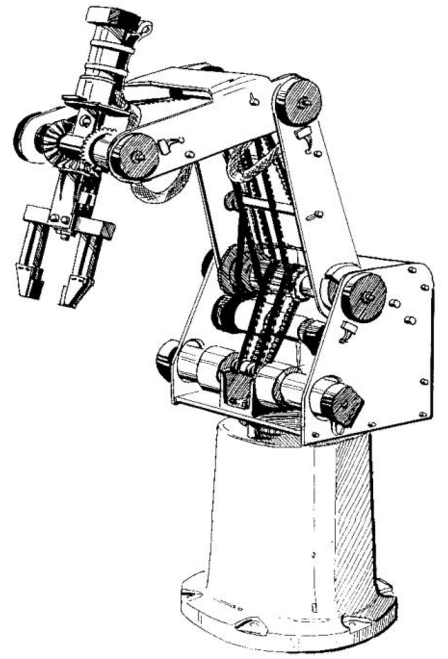


Figure 4: Robot arm construction (Intelitek, 2001).

Table 4

SCORBOT-ER 4u Specifications (Intelitek, 2001).

Mechanical Structure	Vertical articulated
Number of Axes	5 axes plus servo gripper
Axis Movement Axis 1: Base rotation Axis 2: Shoulder rotation Axis 3: Elbow rotation Axis 4: Wrist pitch Axis 5: Wrist roll	310° +130° / –35° ±130° ±130° Unlimited (mechanically); ±570° (electrically)
Maximum Operating Radius	610 mm (24.4")

(continued on following page)

Table 4. continued.

End Effector	DC servo gripper, with optical encoder, parallel finger motion; Measurement of object’s size by means of gripper sensor and software.
Maximum Gripper Opening	75 mm (3") without rubber pads 65 mm (2.6") with rubber pads
Homing	Fixed position on each axis, found by means of microswitches
Feedback	Optical encoder on each axis
Actuators	12 VDC servo motors
Motor Capacity (axes 1–6)	15 oz. in Peak Torque (stall) 70 W Power for Peak Torque
Gear Ratios	Motors 1, 2, 3: 127.1:1 Motors 4, 5: 65.5:1 Motor 6 (gripper) 19.5:1
Transmission	Gears, timing belts, lead screw
Maximum Payload	1 kg (2.2 lb), including gripper
Position Repeatability	±0.18 mm (0.007") at TCP (tip of gripper)
Weight	10.8 kg (23.8 lb)
Maximum Path Velocity	600 mm/sec (23.6"/sec)
Ambient Operating Temperature	2°–40°C (36°–104°F)

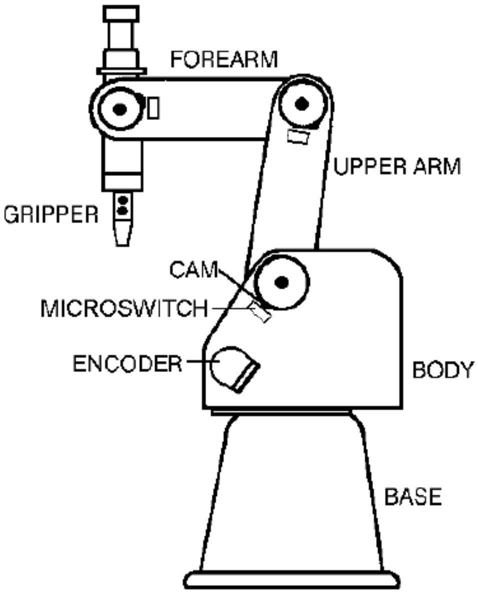


Figure 5: Robot arm parts (Intelitek, 2001)

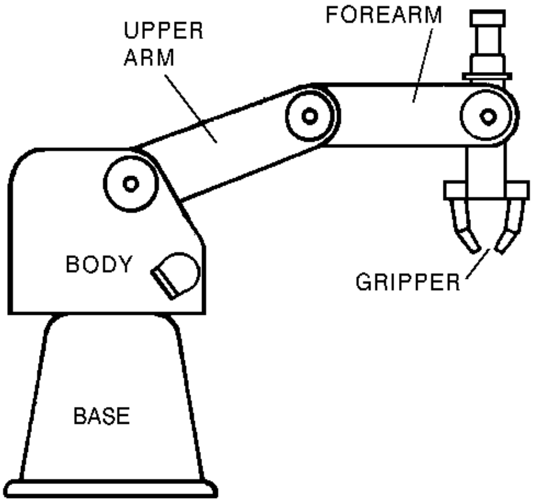


Figure 6: Robot arm links (Intelitek, 2001)

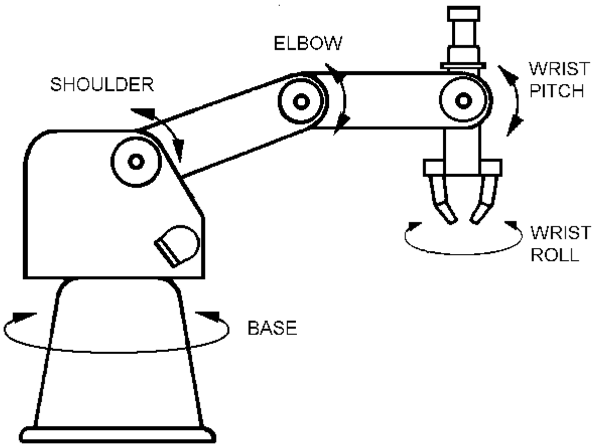


Figure 7: Robot arm joints (Intelitek, 2001).

2.4.1 Work Envelope

The length of the links and the degree of rotation of the joints determine the robot's work envelope. Figures 8 and 9 show the dimensions and reach of the SCORBOT-ER-4u. The base of the robot is normally fixed to a stationary work surface. It may, however, be attached to a slidebase, resulting in an extended working range (Intelitek, 2001).

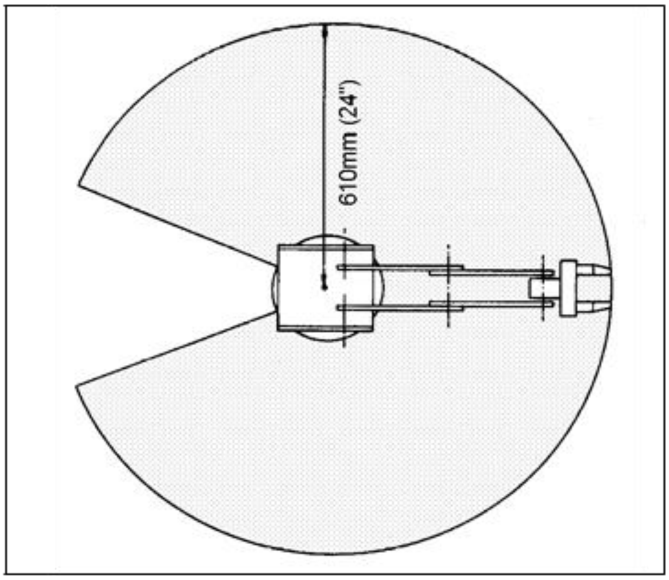


Figure 8: Operating range (top view) (Intelitek, 2001).

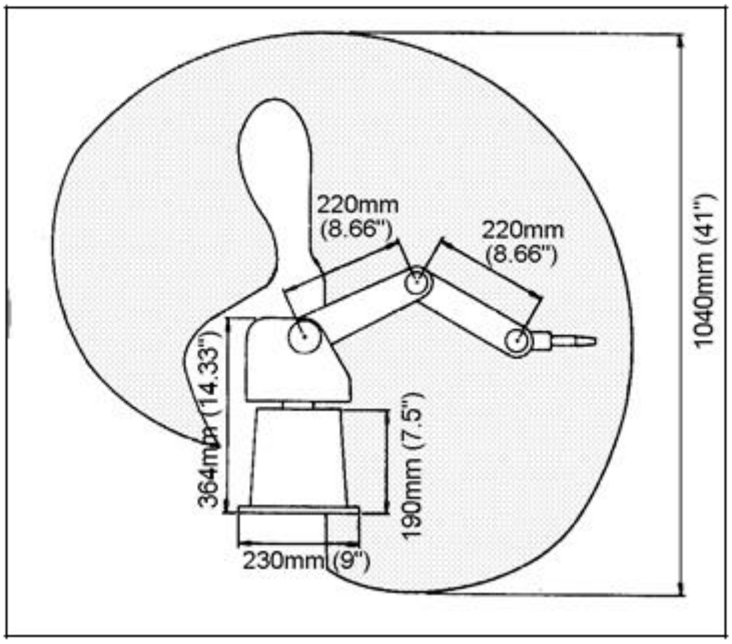


Figure 9: Operating range (side view) (Intelitek, 2001).

2.4.2 Motors

The robot’s five axes and gripper are operated by DC servo motors shown in Figure 10. The direction of motor revolution is determined by the polarity of the operating voltage: positive DC voltage turns the motor in one direction while negative DC voltage turns it in the opposite direction. Each motor is fitted with an encoder for closed-loop control. The motors are from Pittman MFG, operate at 12V DC, nominal and draw up to 3A standard and 7A peak (Eshed Robotec, 1988; Intelitek, 2001).

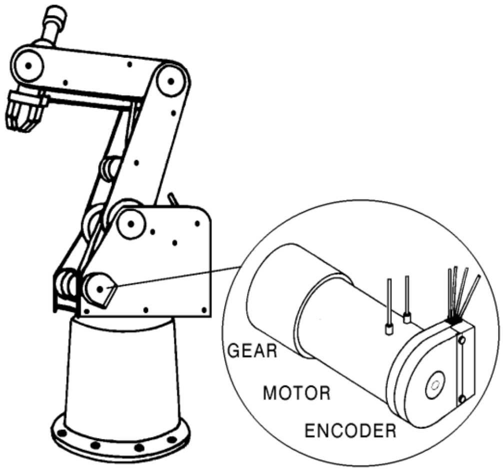


Figure 10: Motor (Intelitek, 2001).

2.4.3 Encoders

The location and movement of each axis is measured by an electro-optical encoder, shown in Figure 11, attached to the shaft of the motor that drives the axis. When the robot axis moves, the encoder generates a series of alternating high and low electrical signals. The number of signals is proportional to the amount of axis motion. There are four wires going through the encoder: the voltage and ground wires and two direction signal wires P_o and P_1. The direction is determined by checking either P_o pulse is leading P_1 pulse or the other case. The sequence of the signals indicates the direction of movement. The controller reads these signals and determines the extent and direction of axis movement (Eshed Robotec, 1988; Intelitek, 2001).

.

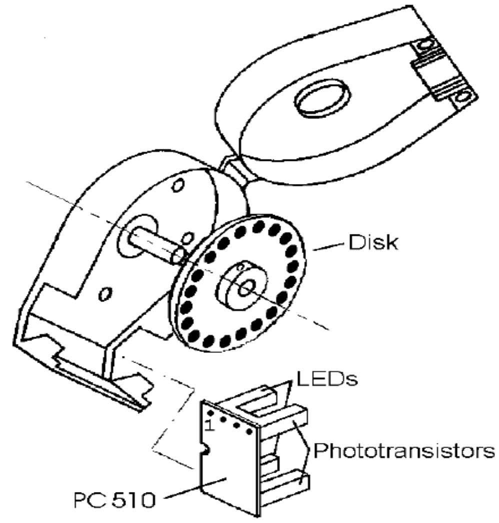


Figure 11: Encoder (Intelitek, 2001).

2.4.4 Microswitch

Five microswitches are fitted onto the frame of the robot arm. When the robot assumes the position shown in Figure 12, in which the microswitch for each joint is depressed (by means of a cam), this predetermined position is known as home. This is the point of reference for robot operation. Whenever the system is turned on, the robot should be sent to this position by means of a software homing routine (Eshed Robotec, 1988; Intelitek 2001).

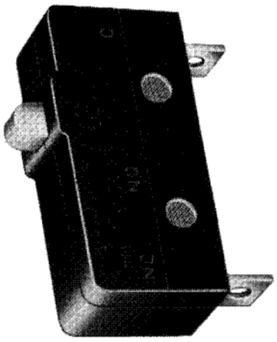


Figure 12: Microswitch (Intelitek, 2001).

2.4.5 Transmissions

Several kinds of transmissions are used to move the links of the robot arm as shown in Figure 13. Spur gears move the base and shoulder axes (Eshed Robotec, 1988; Intelitek, 2001).

- Pulleys and timing belts move the elbow axis.
- Pulleys, timing belts and a bevel gear differential unit at the end of the arm move the wrist pitch and roll axes.
- A lead screw transmission opens and closes the gripper.

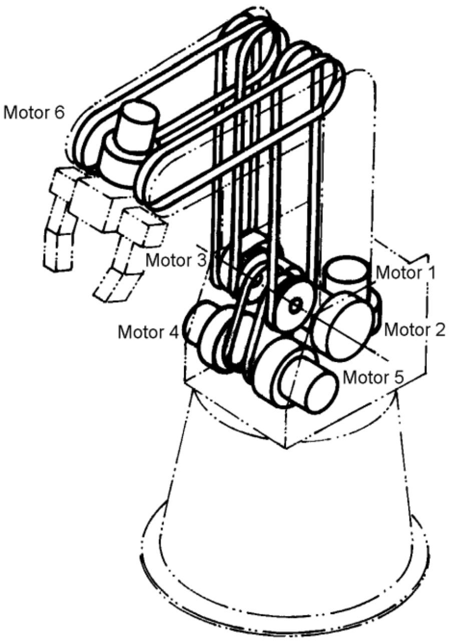


Figure 13: Transmissions (Intelitek, 2001).

2.4.6 Gripper

The SCORBOT-ER-4u consists of a jaw gripper, as shown in Figure 14, fitted with rubber pads. These pads can be removed to allow the attachment of other end-effector devices, like suction pads, electromagnets, etc. Three bevel gears form a differential gear train which moves the wrist joint. When motors 4 and 5 are driven in same direction, the wrist pitch moves up and down, and when they are driven in the opposite directions, the wrist rolls clockwise and counterclockwise. A leadscrew coupled directly to motor 6 causes the gripper to open and close (Eshed Robotec, 1988; Intelitek, 2001).

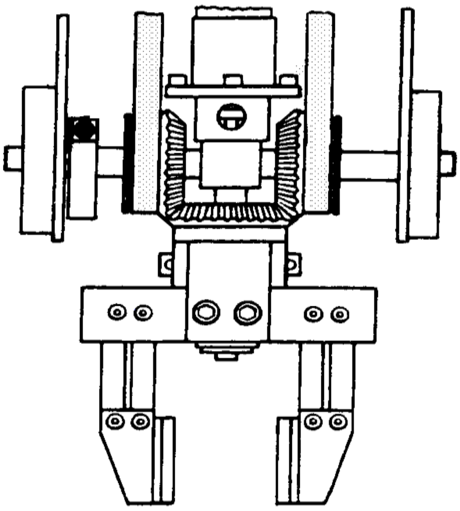


Figure 14: Gripper (Intelitek, 2001).

2.5 Robot Wiring

The robot is connected to the controller by means of a cable which runs from the robot base to the D50 connector shown in Figure 15. The leads from the five motors on the robot body and their encoders are connected directly to the D50 connector on the robot cable. The leads from the gripper motor and the microswitches on the arm reach the D50 connector via a square 12-pin Molex connector in the base of the robot; these leads are particularly flexible and resistant to breakage, even after extensive movement of the robot arm. Table 5 details the wiring for the various electrical components in the robot (Eshed Robotec, 1988; Intelitek, 2001).

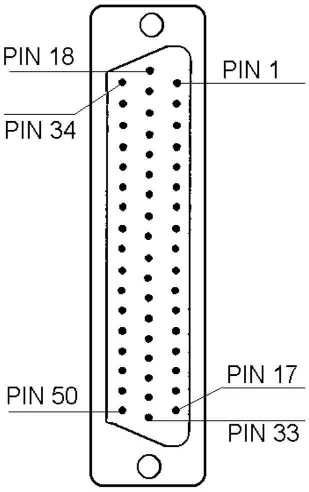


Figure 15: Robot's D50 connector (Intelitek, 2001).

Table 5

SCORBOT-ER-4u Wiring to the D50 Connector (Eshed Robotec, 1988; Intelitek, 2001).

	Robot Arm Signal				Lead to Molex 12-pin Connector		Lead to D 50 Connector	
Axis	Motor	Encoder	Pad #	Microsw.	Color	Pin #	Color	Pin #
1	+ -						white gray/green	50 17
2	+ -						white white/green	49 16
3	+ -						white orange/brown	48 15
4	+ -						white orange/green	47 14
5	+ -						white orange/gray	46 13
Gripper	+ -				gray yellow	8 7	white orange/blue	45 12

(continued on following page)

Table 5. Continued.

1		GND P1 VLED P0	1 4 2 3				white white/Gray yellow brown	33* 5 11 2
2		GND P1 VLED P0	1 4 2 3				white white/orange yellow gray	32* 21 27 1
3		GND P1 VLED P0	1 4 2 3				white brown/blue yellow green	31* 4 10 36
4		GND P1 VLED P0	1 4 2 3				white green/brown yellow orange	30* 20 26 35
5		GND P1 VLED P0	1 4 2 3				white green/blue yellow blue	29* 3 9 18
Gripper		GND P1 VLED P0	1 3 2 4		black green yellow brown	12 11 10 9	white gray/blue white white/blue	28* 19 25 34
1				GND MS			white brown	33* 23
2				GND MS			white gray	32* 7
3				GND MS	white white	1 2	white orange	31* 24
4				GND MS	blue blue	3 4	white green	30* 8
5				GND MS	orange orange	5 6	white blue	29* 6
Gripper				no connection			white brown/gray	28* 22

2.6 PWM Operation

The power unit in the controller drives the motors by switching ±24V to it at 15 kHz, according to the input pulse. The motor cannot react to this high frequency of switching and is therefore affected by only the average value of the voltage. This method of controlling the time during which current flows through the motor, rather than controlling the value of the current, is known as PWM (pulse width modulation) control as shown in Figure 16. Once per cycle the processor reads the encoder’s count and calculates the motor’s position and speed (rate of encoder counts). The processor then compares the actual (output) position and speed values with the desired (input) ones, determines the error values and takes the necessary action to cancel them. The entire control cycle takes 10ms. The processor calculates the command position and speed once per cycle. It outputs a digital value to the DAC unit in the range of ±5000. The analog unit creates a series of pulses, resulting in an average voltage value proportional to the DAC input and supplies it to the power unit for PWM generation. The whole system is shown in Figure 17. (Eshed Robotec, 1988).

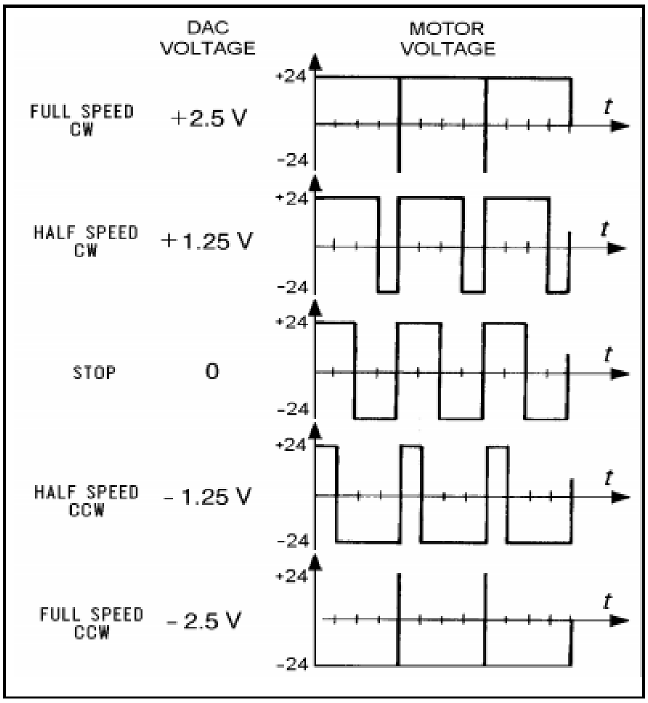


Figure 16: Robot controller control signals, PWM (Eshed Robotec, 1988).

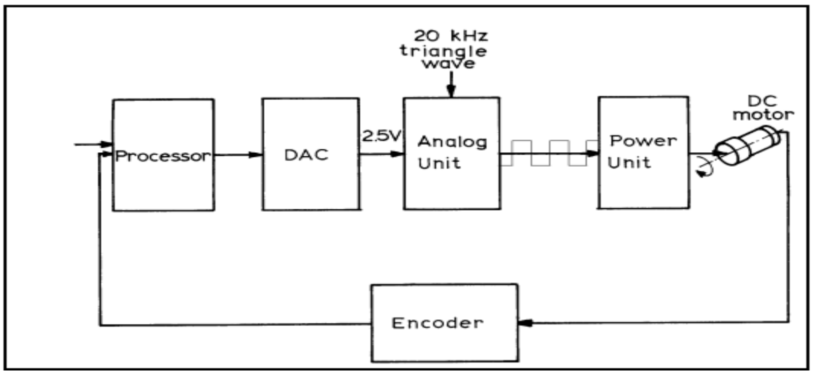


Figure 17: Robot controller control loop (Eshed Robotec, 1988).

CHAPTER 3

SCORBASE SOFTWARE

3.1 Introduction to SCORBASE

SCORBASE for SCORBOT-ER-4u is a robotics control software package for robot programming and operation. SCORBASE provides numerous capabilities (Intelitek, 2006):

- Communication with the robot controller over USB channel.
- Control and real-time status display of five robot axes and gripper.
- Position definition and display as well as manual robot movement in reference to joint coordinate system (encoder units).
- The Cartesian coordinate system (X, Y, Z pitch and roll) is also used.
- Robot movement definition as Go to Position, Go Linear, or Go Circular, with ten active speed settings. (Availability depends on experience level setting.)
- Variable programming, in three levels of complexity, to moderate the learning curve. This makes it possible for beginners to start at a lower level and advance through the levels, as they become more skilled in robotics programming.
- SCORBASE can be installed as part of RoboCell, an interactive graphic software package, which provides simulation of the robot and other devices in the workcell.

3.2 Overview

SCORBASE provides user-friendly interface and visible robotic program execution. The software has several operational levels which can be easily used by both learner and advanced users. It has a predefined command set and different levels like L1, L2 and Pro. They will have access to these command sets based on their level of programming. Let us see more details about setting up the software. The SCORBASE offers the following options for system configuration (Intelitek, 2006):

- Window layout options.
- Experience level.
- Hardware setup.

3.2.1 Window Layout Options

The display options enable optimal usage of the screen area. SCORBASE offers five basic display options and a wide range of dialog bars and windows through which the user can see and change system data. The different window screens available are given below and we will be mostly using the Teach and Edit screen (Intelitek, 2006).

- Teach and Edit
- Run Screen
- Project Screen
- Open CIM Screen
- User Screen

3.2.2 Experience Level

SCORBASE offers three experience levels and are also shown in Table 6:

- Introductory (Level 1)
- Advanced (Level 2)
- Professional (Pro Level)

A higher level offers more commands and tools. Levels can be selected from the Tool bar or from the Options menu (Intelitek, 2006).

Table 6

Different Levels and their respective functions (Intelitek, 2006).

Level 1	Displays list of commands and options at introductory level. Commands related to Level 2 and Pro are disabled.
Level 2	Displays list of commands and options at advanced level. Commands related to Pro are disabled.
Pro	Displays list of all commands and options.

3.2.3 Hardware setup

SCORBASE allows you to define the devices that are connected and operated by the controller as Axes 7 and 8. The Hardware Setup option also enables you to work with a different robot from the one you selected during installation. Table 7 lists the peripherals that can be connected to the USB controller (Intelitek, 2006).

Table 7

List of all peripherals available (Intelitek, 2006).

Catalog number	Description
1009	Rotary Table, 24V
1010	Conveyor Belt (gray), 24V
1020	1.0m Linear Slidebase, belt-drive, 24V
1021	1.8m Linear Slidebase, belt-drive, 24V
1013	Linear Table 0.3m, 24V
1014	XY-Table, 24V
1234	Motor Kit 24V

3.3 Homing and Control

The robot and peripheral axes location are monitored and controlled using encoders. To initialize the encoders and to obtain repeatable performance, the axes must first reach a predefined position known as hard home. All recorded positions and movements refer to the hard home position. The homing procedure finds the hard home for the selected axes. SCORBASE offers Search Home command relating to the home position as shown in Figure 18. During Search Home, each axis is homed separately. The controller activates the currently homed motor axis, until its microswitch is pressed. Then the controller initializes the axis encoder counter and turns to home the next axis. After all configured axes are homed, the homing procedure ends. The Search Home command is available in three different levels (Intelitek, 2006):

- Go Home All (applies for all active axes)

- Go Home Robot (applies for the robot)
- Go Home Peripherals (applies for axes 7 & 8)

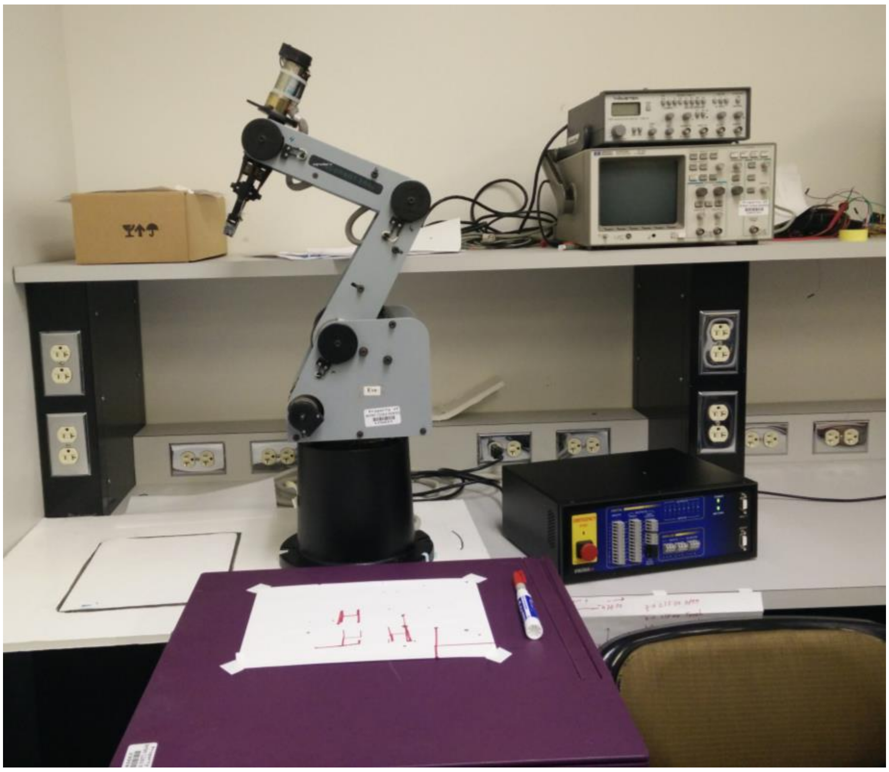


Figure 18: A picture of robot home position.

3.4 Position Definition

Every SCORBASE project includes a set of pre-defined positions and a program that sends the robot from one position to the other. Prior to running a program, all the positions used in that program should be defined. SCORBASE offers various tools to define and store positions that will be used in the programs. The following SCORBASE tools are used in the position definition process (Intelitek, 2006):

- Manual Movement Dialog Box
- Teach Positions Dialog Box

- Robot Movement Dialog Box
- Position Data Dialog Bars
- Positions Window

Defining a position in SCORBASE can be done by using either the joint or Cartesian coordinate systems. In both systems, a robot position is defined using five parameters derived from the data supplied by the five axes encoders. An encoder is an angular movement sensor attached to the axes motor. A peripheral position is always defined using one variable that stores the sensor output (encoder value) of that position. In this thesis the Cartesian coordinate system is been used. A robot position in Cartesian (or XYZ) coordinates is defined by these parameters: The distance of the robot's tool center point (TCP) from the point of origin (the center bottom of the robot base); Figure 19, along the three axes that describe three-dimensional space (X, Y, Z) and the pitch (P) and roll (R) angles of the gripper, specified in angular units. For example, after homing, the robot position in Cartesian coordinates is defined as (Intelitek, 2006):

- X = (~169) [mm]
- Y = (0) [mm]
- Z = (~503) [mm]
- Pitch = (~-63°)
- Roll = (0°)

SCORBASE offers two methods of defining a robot or peripheral position: absolute and relative. The two methods are applicable in Cartesian and in joint coordinates. An absolute position is defined using all five robot position parameters. If the joint coordinate system is used, the robot position is defined using the base, shoulder, elbow, pitch, and roll angles. In the XYZ coordinate system, a position is defined using the X, Y, Z values in millimeters and the pitch and roll angles in degrees. An absolute position is usually a fixed position in world space whereas a relative position is a position whose coordinates are defined as an offset from a reference position coordinates. If the coordinates of the reference position change, the relative position moves along with it, maintaining the same offset. A relative position can be defined in either Cartesian or joint coordinate values. A position can also be defined as relative to current. In this case, the relative position is calculated as an offset from the coordinates of the current robot position.

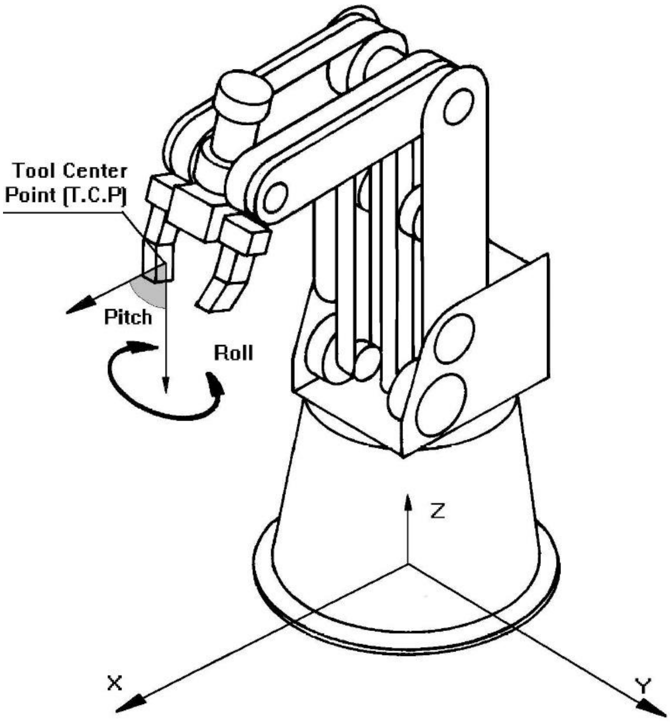


Figure 19: Tool center point in Cartesian coordinate system (Intelitek, 2006).

3.5 Command Set

SCORBASE commands (Table 8) are organized in a command tree, which is displayed in the workspace window. SCORBASE only displays the commands which are available in the currently set experience level. At the Introductory level (Level 1), only basic commands appear in the command tree. At the Advanced level (Level 2), the number of commands is increased. At the Professional level (Pro Level), all commands are accessible. SCORBASE commands are grouped into these categories (Intelitek, 2006):

- Axis and Control
- Program Flow
- Inputs and Outputs
- Advanced - By default, this category is hidden. Parameter commands are displayed.
- Vision - By default, this category is hidden. ViewFlex commands are displayed.

Double click on the desired command and enter the required fields and hit enter. Then the program code will be automatically added to the script.

Table 8

Commands set and description in SCORBASE (Intelitek, 2006)

Command	Levels	Description
OG Open Gripper	1,2,Pro	Fully opens the gripper.
CG Close Gripper	1,2,Pro	Fully closes the gripper (on itself, or on a grasped Object).
GP Go to Position	1,2,Pro	The Go to Position command sends the robot to a recorded position, in the shortest time, using Point to Point (P to P) control.
GL Go Linear to Position #_Speed	2,Pro	Sends the robot's TCP (tool center point) from its current position to the target position, along a linear path (straight line). The linear motion applies only to the robot axes.
GC Go Circular to Position #_ Speed	2,Pro	Sends the robot's TCP in a circular path to the target position, via the position specified in the Via Position field. The circular motion applies only to the robot
JA Jaw	2,Pro	Moves the gripper's jaw to the specified span
IL If Limit Switch < # > on jump to <Label>	2,Pro	The IL is a conditional jump command. It causes program execution to jump to the line that contains the specified Label, if the selected axis micro switch is pressed (On).
RP Record Position #_	Pro	When the Record Position command is executed (during program execution), the controller records the current position data to the specified position.
SA Set Axis #_ ... (to Zero)	2,Pro	Initializes (sets to 0) the encoder count of the selected axis
SC Start Conveyor	Pro	Starts the conveyor, as a speed-controlled conveyor. Movement of the conveyor will continue until a Stop Conveyor (ST) command is encountered

(continued on following page)

Table 8. Continued.

ST Stop Conveyor	Pro	Stops the continuous motion of a conveyor that was initiated by a Start Conveyor (SC) command.
SG Set Variable to Gripper Sensor	Pro	Assigns the value of the gripper opening (in mm) to a variable.
TJ Teach Position By Joints	Pro	Teaches the position as defined by the absolute positions of the joints
RJ Teach Position By Joints Relative to Position	Pro	Teaches the position as defined by the positions of the joints relative to another defined position
TX Teach Position by XYZ	Pro	Teaches the position as defined by the absolute Cartesian coordinate of the TCP (Tool Center Point).
RX Teach Position by XYZ Relative to Position	Pro	Teaches the position as defined by the Cartesian coordinate of the TCP (Tool Center Point) relative to another defined position
SJ Set Variable to Joint Position		Allows you to set a variable to the value of a specified joint at a specified position number
SX Set Variable to Cartesian Position		Allows you to set a variable to the value of a specified Cartesian coordinate at a specified position number
WT Wait (10ths of second)	1,2,Pro	Halts program execution for a time specified in tenths of a second
JU Jump to	1,2,Pro	This unconditional jump command causes the program pointer to jump to the line that contains the specified Label
RE Remark	2,Pro	Allows insertion of a comment line for explanation and documentation into the program
RB Ring Bell	2,Pro	When executed, this command produces a beep, using the computer’s internal loudspeaker
RT Reset Timer	Pro	SCORBASE uses a timer that measures time in units of tenths of a second. The timer starts operating when SCORBASE is opened. The Reset Timer command resets the value of the SCORBASE timer to 0.

(continued on following page)

Table 8. Continued.

SV Set Variable to Computation	1,2,Pro	Allows you to assign a value, or an expression (result of a specific computation), to a variable
VT Set Variable to Timer	Pro	Allows you to assign the current value of SCORBASE timer to a variable.
IF If Jump to	2,Pro	A conditional branch command, which is used to determine the program flow in relation to the value of the variables
SS Set Subroutine	2,Pro	Creates a subroutine. You can program up to 64 subroutines in one program.
RS Return from Subroutine	2,Pro	Marks the end of a subroutine. At run time, this command terminates the execution of the subroutine, and the program resumes execution at the line that follows the Call Subroutine command
CS Call Subroutine	2,Pro	Activates the specified subroutine
LA Label	1,2,Pro	Marks a line in the program that is referenced by a Jump command.
PS Print to Screen & Log	2,Pro	Instructs SCORBASE to print data containing strings, messages and variable values to a log file, or to the message window, or to both.
II If Input # On/Off Jump	1,2,Pro	Causes the program to jump to a label or call a subroutine, if the state of the tested digital input matches the status specified (On or Off).
OI On Input Interrupt # On/Off	Pro	Sets the condition for an input interrupt service. The service (Call Subroutine or Run Subroutine) will be performed whenever the condition (input status) is satisfied, regardless of the current program pointer position
DI Disable Interrupt #	Pro	Causes the specified input interrupt to become inactive.
EI Enable Interrupt #	Pro	Causes the specified input interrupt to become active.

(continued on following page)

Table 8. Continued.

ON Turn On Output #	1,2,Pro	Sets the state of the specified digital output On
OF Turn Off Output #	1,2,Pro	Sets the state of the specified digital output Off
AO Set Analog Output #	Pro	Sets the state of the specified analog output.
AI Set Variable to Analog Input #	Pro	Sets the value of the specified analog input to a variable

CHAPTER 4

IMPLEMENTATION OF PROPOSED IDEA

4.1 Planning the Process

In order to implement the idea, we need to understand the methodology of writing characters. Alphabets are mainly comprised of two types of segments, mainly a straight line and a curve. Many combinations of these segments can be designed using different sizes or type, which in turn create different characters. This information has a significant role on our goal to enable robots to write. As we discussed, characters are only made of straight lines and curves, any robot that can draw straight lines and curves has the ability to write characters. But the essential part of making a robot properly write the characters is a proper guideline on the type and size of the segments that need to be drawn for a particular character. This is where the implementation plays its role. Each character is divided into segments and each of these segments is stored separately as a single movement. When a character needs to be written, the respective movements are called simultaneously, and thus developing the requested character. The main important aspects in implementation of this work are as follows:

- Segmentation of the characters
- Storing the segments
- Using the stores segments

4.2 Segmentation of Characters

Character segmentation splits a character into segments that involve straight lines and curves. By doing so, we are dividing a character into components that can be drawn by a robotic arm. For example, if we take sample character “M”, it can be divided into four straight lines. If we select letter “D”, we have a straight line and a curve segment. Similarly, by applying this to all characters we can divide each character into segments. Table 9 gives detailed information of how the characters are segmented. This thesis is limited to write all the alphabetical letters only.

Table 9

Character Segmentation Information

Character	Segment 1	Segment 2	Segment 3	Segment 4
A	/	—	\	
B	│	ɔ	ɔ	
C	C			
D	│	Ɔ		

(continued on following page)

Table 9. Continued.

E	│	‾	—	_
F	│	‾	—	
G	C	│	—	
H	│	—	│	
I	│	‾	_	
J	│	◡	‾	
K	│	/	\	
L	│	_		
M	│	/	\	│
N	│	\	│	
O	C	Ɔ		

(continued on following page)

Table 9. Continued.

P	|	)		
Q	C	Ɔ	\	
R	|	)	\	
S	(	)		
T	|	—		
U	|	◡	|	
W	\	/	\	/
X	\	/		
Y	\	/	|	
Z	—	/	_	

4.3 Storing of Segments

Once the characters are segmented, the next step is to figure out how to store these segments in a way that the robot understands them. There are only two types of segments, a line and a curve. The line has only two points that need to be stored: they are starting and ending points. A curve has three points to be stored, they are start point, end point and a peak of the curve point. Assuming that the robot is going to write in 2-dimensional horizontal flat surfaces, the segment position can be represented in Cartesian coordinates (X, Y). The position data that is the (X, Y) points that will be stored can be interpreted in few different ways. Depending on the need, one or all the different ways have been used in storing the segments. They are as follows:

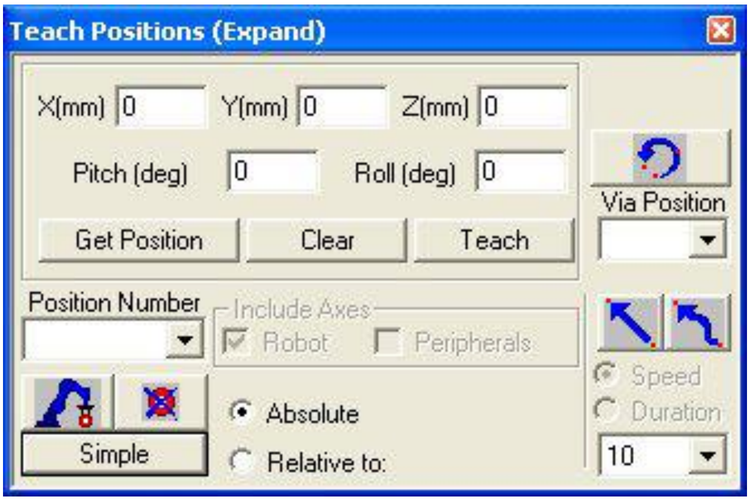
- The actual coordinates that the robotic arm is going to use.
- Displacement from a particular (X, Y) coordinate.
- Displacement from the current position of the robotic arm.

The database proposed here is a position information table available in the SCORBASE software. This table mainly consists of all the position information regarding all axes and types of position whether it is absolute or relative. Data can be uploaded to this table using Teach Position dialog box available in SCORBASE software. Figure 20 shows the structure of a sample teach positions dialog box. The Simple/Expanded Teach Positions dialog box enables the following:

- Teaching positions (in Cartesian coordinates).
- Recording positions (in joint coordinates).
- Sending the axes to the recorded positions (when program is not running).

- Go to Position.
- Go Linear.
- Go Circular.

Using this dialog box, we can command the robot to record its present position and go to a specific position by entering the position values in the respective columns. We can also choose between absolute or relative position definitions using this dialog box. Table 10 shows all the available options in the teach positions dialog box.



Teach Positions (Expand)

X(mm) 0 Y(mm) 0 Z(mm) 0

Pitch (deg) 0 Roll (deg) 0

☑ Robot □ Peripherals

● Absolute ○ Relative to:

● Speed ○ Duration 10

Figure 20: Teach Positions (Expand) dialog box.

Table 10

All available options in Teach Positions Dialog Box (Intelitek, 2006)

X(mm), Y(mm) Z(mm), Pitch(deg), Roll(deg)	Fields for displaying or changing the Cartesian coordinates of the selected position.
Get Position	Displays the Cartesian coordinates of the selected position.
Clear	Clears all position coordinate fields. The position data does not change.

(continued on following page)

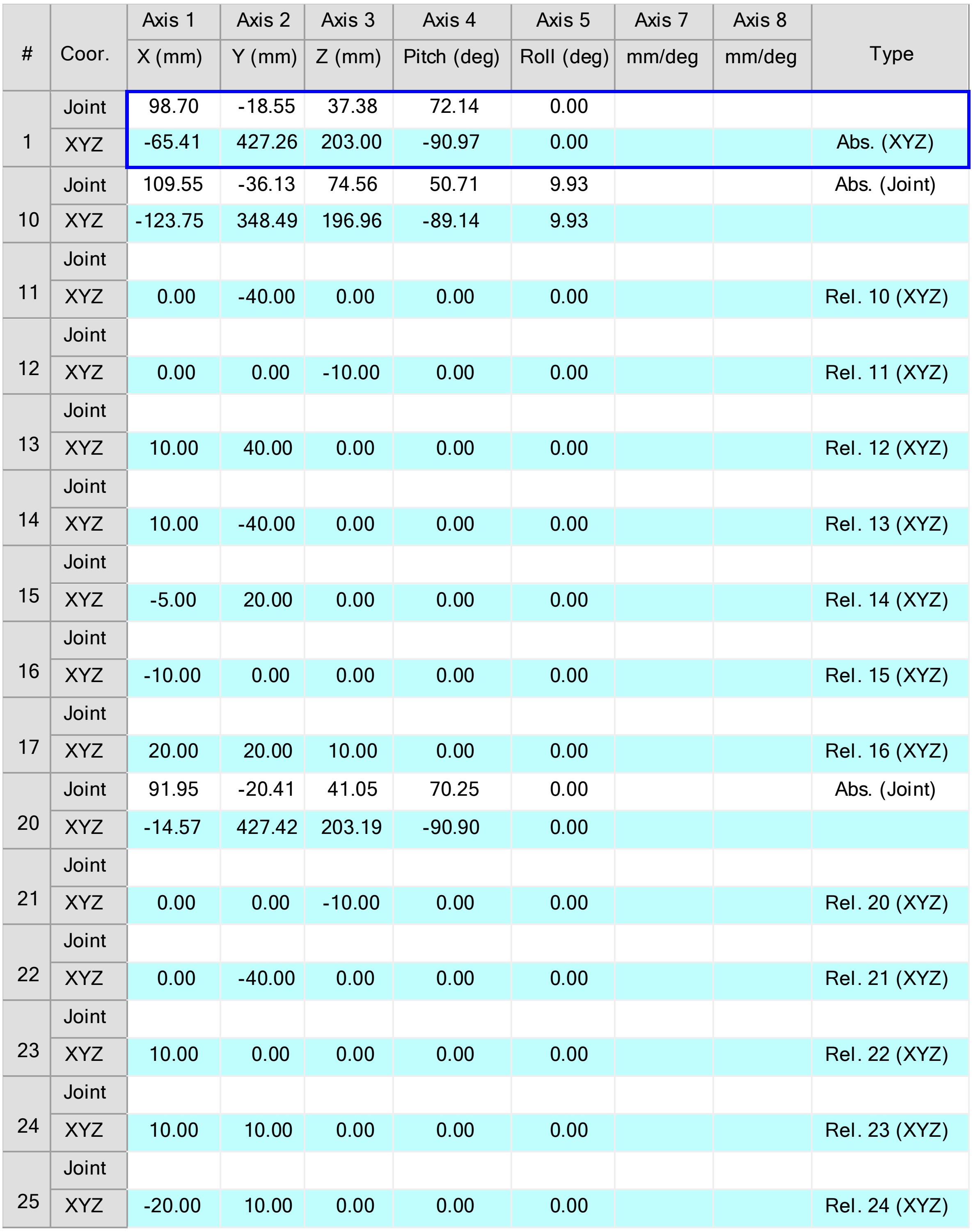
Table 10. Continued.

Teach	Teaches position using the Cartesian coordinate system.
Go Circular	Executes the Go Circular to Position command. This command sends the robot in a circular path to the target position (in the position number field) via the position specified in the Via position field. The circular motion applies only to the robot.
Via position	Selects the intermediate position through which the Go Circular movement passes.
Go Linear to Position	Executes the Go Linear to Position command. Sends the axes in a straight line to the selected position.
Simple	Toggles to the Teach Position (Simple) dialog box.

Now using the segment information table and the teach positions dialog box, we can store the position coordinates of each segment. Each segment must be properly planned and placed so that the robot has the shortest and optimized path. Each character is planned to have 20mm of width, 40mm of height and 5mm of space between each character. So each segment is placed exactly to have the proposed width and height. Some of the characters are planned to write from top to bottom and some of them bottom to top based on the optimum path possible. Table 11 shows the position information of all the segments of all characters. In the table, the segments of first letter “A” is represented from position number 10 to 17 and segments of the second letter “B” is stored from position number 20 to 29 and so on.

Table 11

Position Information Table



#	Coor.	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 7	Axis 8	Type
		X (mm)	Y (mm)	Z (mm)	Pitch (deg)	Roll (deg)	mm/deg	mm/deg	
1	Joint	98.70	-18.55	37.38	72.14	0.00			
	XYZ	-65.41	427.26	203.00	-90.97	0.00			Abs. (XYZ)
10	Joint	109.55	-36.13	74.56	50.71	9.93			Abs. (Joint)
	XYZ	-123.75	348.49	196.96	-89.14	9.93			
11	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 10 (XYZ)
12	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 11 (XYZ)
13	Joint								
	XYZ	10.00	40.00	0.00	0.00	0.00			Rel. 12 (XYZ)
14	Joint								
	XYZ	10.00	-40.00	0.00	0.00	0.00			Rel. 13 (XYZ)
15	Joint								
	XYZ	-5.00	20.00	0.00	0.00	0.00			Rel. 14 (XYZ)
16	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 15 (XYZ)
17	Joint								
	XYZ	20.00	20.00	10.00	0.00	0.00			Rel. 16 (XYZ)
20	Joint	91.95	-20.41	41.05	70.25	0.00			Abs. (Joint)
	XYZ	-14.57	427.42	203.19	-90.90	0.00			
21	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 20 (XYZ)
22	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 21 (XYZ)
23	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 22 (XYZ)
24	Joint								
	XYZ	10.00	10.00	0.00	0.00	0.00			Rel. 23 (XYZ)
25	Joint								
	XYZ	-20.00	10.00	0.00	0.00	0.00			Rel. 24 (XYZ)

(continued on following page)

Table 11. Continued.

26	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 25 (XYZ)
27	Joint								
	XYZ	10.00	10.00	0.00	0.00	0.00			Rel. 26 (XYZ)
28	Joint								
	XYZ	-20.00	10.00	0.00	0.00	0.00			Rel. 27 (XYZ)
29	Joint								
	XYZ	25.00	0.00	10.00	0.00	0.00			Rel. 28 (XYZ)
30	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
31	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 30 (XYZ)
32	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 31 (XYZ)
33	Joint								
	XYZ	-20.00	-20.00	0.00	0.00	0.00			Rel. 32 (XYZ)
34	Joint								
	XYZ	20.00	-20.00	0.00	0.00	0.00			Rel. 33 (XYZ)
35	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 34 (XYZ)
40	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
41	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 40 (XYZ)
42	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 41 (XYZ)
43	Joint								
	XYZ	20.00	20.00	0.00	0.00	0.00			Rel. 42 (XYZ)
44	Joint								
	XYZ	-20.00	20.00	0.00	0.00	0.00			Rel. 43 (XYZ)
45	Joint								
	XYZ	25.00	20.00	10.00	0.00	0.00			Rel. 44 (XYZ)
50	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			

(continued on following page)

Table 11. Continued.

51	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 50 (XYZ)
52	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 51 (XYZ)
53	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 52 (XYZ)
54	Joint								
	XYZ	0.00	20.00	10.00	0.00	0.00			Rel. 53 (XYZ)
55	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 54 (XYZ)
56	Joint								
	XYZ	-20.00	0.00	0.00	0.00	0.00			Rel. 55 (XYZ)
57	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 56 (XYZ)
58	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 57 (XYZ)
59	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 58 (XYZ)
60	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
61	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 60 (XYZ)
62	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 61 (XYZ)
63	Joint								
	XYZ	-20.00	0.00	0.00	0.00	0.00			Rel. 62 (XYZ)
64	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 63 (XYZ)
65	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 64 (XYZ)
66	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 65 (XYZ)
67	Joint								
	XYZ	5.00	20.00	10.00	0.00	0.00			Rel. 66 (XYZ)

(continued on following page)

Table 11. Continued.

70	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
71	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 70 (XYZ)
72	Joint								
	XYZ	-10.00	-20.00	0.00	0.00	0.00			Rel. 71 (XYZ)
73	Joint								
	XYZ	10.00	-20.00	-10.00	0.00	0.00			Rel. 72 (XYZ)
74	Joint								
	XYZ	5.00	5.00	0.00	0.00	0.00			Rel. 73 (XYZ)
75	Joint								
	XYZ	-5.00	5.00	0.00	0.00	0.00			Rel. 74 (XYZ)
76	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 75 (XYZ)
77	Joint								
	XYZ	0.00	-10.00	0.00	0.00	0.00			Rel. 76 (XYZ)
78	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 77 (XYZ)
80	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
81	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 80 (XYZ)
82	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 81 (XYZ)
83	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 82 (XYZ)
84	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 83 (XYZ)
85	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 84 (XYZ)
86	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 85 (XYZ)
87	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 86 (XYZ)

(continued on following page)

Table 11. Continued.

90	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
91	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 90 (XYZ)
92	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 91 (XYZ)
93	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 92 (XYZ)
94	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 93 (XYZ)
95	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 94 (XYZ)
96	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 95 (XYZ)
97	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 96 (XYZ)
100	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
101	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 100 (XYZ)
102	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 101 (XYZ)
103	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 102 (XYZ)
104	Joint								
	XYZ	0.00	-30.00	0.00	0.00	0.00			Rel. 103 (XYZ)
105	Joint								
	XYZ	-5.00	-10.00	0.00	0.00	0.00			Rel. 104 (XYZ)
106	Joint								
	XYZ	-5.00	10.00	0.00	0.00	0.00			Rel. 105 (XYZ)
107	Joint								
	XYZ	25.00	30.00	10.00	0.00	0.00			Rel. 106 (XYZ)
110	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			

(continued on following page)

Table 11. Continued.

111	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 110 (XYZ)
112	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 111 (XYZ)
113	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 112 (XYZ)
114	Joint								
	XYZ	20.00	20.00	0.00	0.00	0.00			Rel. 113 (XYZ)
115	Joint								
	XYZ	-20.00	-20.00	0.00	0.00	0.00			Rel. 114 (XYZ)
116	Joint								
	XYZ	20.00	-20.00	0.00	0.00	0.00			Rel. 115 (XYZ)
117	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 116 (XYZ)
120	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
121	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 120 (XYZ)
122	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 121 (XYZ)
123	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 122 (XYZ)
124	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 123 (XYZ)
130	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
131	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 130 (XYZ)
132	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 131 (XYZ)
133	Joint								
	XYZ	0.00	40.00	0.00	0.00	0.00			Rel. 132 (XYZ)
134	Joint								
	XYZ	10.00	-10.00	0.00	0.00	0.00			Rel. 133 (XYZ)

(continued on following page)

Table 11. Continued.

135	Joint								
	XYZ	10.00	10.00	0.00	0.00	0.00			Rel. 134 (XYZ)
136	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 135 (XYZ)
137	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 136 (XYZ)
140	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
141	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 140 (XYZ)
142	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 141 (XYZ)
143	Joint								
	XYZ	0.00	40.00	0.00	0.00	0.00			Rel. 142 (XYZ)
144	Joint								
	XYZ	20.00	-40.00	0.00	0.00	0.00			Rel. 143 (XYZ)
145	Joint								
	XYZ	0.00	40.00	0.00	0.00	0.00			Rel. 144 (XYZ)
146	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 145 (XYZ)
150	Joint	88.56	-20.49	41.16	70.23	0.00			Abs. (Joint)
	XYZ	10.78	427.39	203.38	-90.90	0.00			
151	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 150 (XYZ)
152	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 151 (XYZ)
153	Joint								
	XYZ	-10.00	-20.00	0.00	0.00	0.00			Rel. 152 (XYZ)
154	Joint								
	XYZ	10.00	-20.00	0.00	0.00	0.00			Rel. 153 (XYZ)
155	Joint								
	XYZ	10.00	20.00	0.00	0.00	0.00			Rel. 154 (XYZ)
156	Joint								
	XYZ	-10.00	20.00	0.00	0.00	0.00			Rel. 155 (XYZ)

(continued on following page)

Table 11. Continued.

157	Joint								
	XYZ	15.00	0.00	10.00	0.00	0.00			Rel. 156 (XYZ)
160	Joint	95.49	-21.72	45.41	66.31	91.11			Abs. (Joint)
	XYZ	-40.54	421.75	196.98	-90.00	91.11			
161	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 160 (XYZ)
162	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 161 (XYZ)
163	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 162 (XYZ)
164	Joint								
	XYZ	20.00	10.00	0.00	0.00	0.00			Rel. 163 (XYZ)
165	Joint								
	XYZ	-20.00	10.00	0.00	0.00	0.00			Rel. 164 (XYZ)
166	Joint								
	XYZ	25.00	0.00	10.00	0.00	0.00			Rel. 165 (XYZ)
167	Joint								
	XYZ	5.00	40.00	10.00	0.00	0.00			Rel. 179 (XYZ)
170	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
171	Joint								
	XYZ	10.00	0.00	0.00	0.00	0.00			Rel. 170 (XYZ)
172	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 171 (XYZ)
173	Joint								
	XYZ	-10.00	-20.00	0.00	0.00	0.00			Rel. 172 (XYZ)
174	Joint								
	XYZ	10.00	-20.00	0.00	0.00	0.00			Rel. 173 (XYZ)
175	Joint								
	XYZ	10.00	20.00	0.00	0.00	0.00			Rel. 174 (XYZ)
176	Joint								
	XYZ	-10.00	20.00	0.00	0.00	0.00			Rel. 175 (XYZ)
177	Joint								
	XYZ	0.00	-30.00	10.00	0.00	0.00			Rel. 176 (XYZ)

(continued on following page)

Table 11. Continued.

178	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 177 (XYZ)
179	Joint								
	XYZ	10.00	-10.00	0.00	0.00	0.00			Rel. 178 (XYZ)
180	Joint	98.70	-18.55	37.38	72.14	0.00			Abs. (Joint)
	XYZ	-65.36	427.27	203.02	-90.97	0.00			
181	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 180 (XYZ)
182	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 181 (XYZ)
183	Joint								
	XYZ	0.00	20.00	0.00	0.00	0.00			Rel. 182 (XYZ)
184	Joint								
	XYZ	20.00	-20.00	0.00	0.00	0.00			Rel. 183 (XYZ)
185	Joint								
	XYZ	-20.00	20.00	0.00	0.00	0.00			Rel. 184 (XYZ)
186	Joint								
	XYZ	20.00	10.00	0.00	0.00	0.00			Rel. 185 (XYZ)
187	Joint								
	XYZ	-20.00	10.00	0.00	0.00	0.00			Rel. 186 (XYZ)
188	Joint								
	XYZ	25.00	0.00	10.00	0.00	0.00			Rel. 187 (XYZ)
190	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
191	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 190 (XYZ)
192	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 191 (XYZ)
193	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 192 (XYZ)
194	Joint								
	XYZ	-10.00	-10.00	0.00	0.00	0.00			Rel. 193 (XYZ)
195	Joint								
	XYZ	10.00	-10.00	0.00	0.00	0.00			Rel. 194 (XYZ)

(continued on following page)

Table 11. Continued.

196	Joint								
	XYZ	10.00	-10.00	0.00	0.00	0.00			Rel. 195 (XYZ)
197	Joint								
	XYZ	-10.00	-10.00	0.00	0.00	0.00			Rel. 196 (XYZ)
198	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 197 (XYZ)
199	Joint								
	XYZ	25.00	40.00	10.00	0.00	0.00			Rel. 198 (XYZ)
200	Joint	85.18	-19.99	40.11	70.77	0.00			Abs. (Joint)
	XYZ	36.05	427.41	203.54	-90.90	0.00			
201	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 200 (XYZ)
202	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 201 (XYZ)
203	Joint								
	XYZ	-10.00	0.00	0.00	0.00	0.00			Rel. 202 (XYZ)
204	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 203 (XYZ)
205	Joint								
	XYZ	15.00	40.00	10.00	0.00	0.00			Rel. 204 (XYZ)
210	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
211	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 210 (XYZ)
212	Joint								
	XYZ	0.00	-30.00	0.00	0.00	0.00			Rel. 211 (XYZ)
213	Joint								
	XYZ	10.00	-10.00	0.00	0.00	0.00			Rel. 212 (XYZ)
214	Joint								
	XYZ	10.00	10.00	0.00	0.00	0.00			Rel. 213 (XYZ)
215	Joint								
	XYZ	0.00	30.00	0.00	0.00	0.00			Rel. 214 (XYZ)
216	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 215 (XYZ)

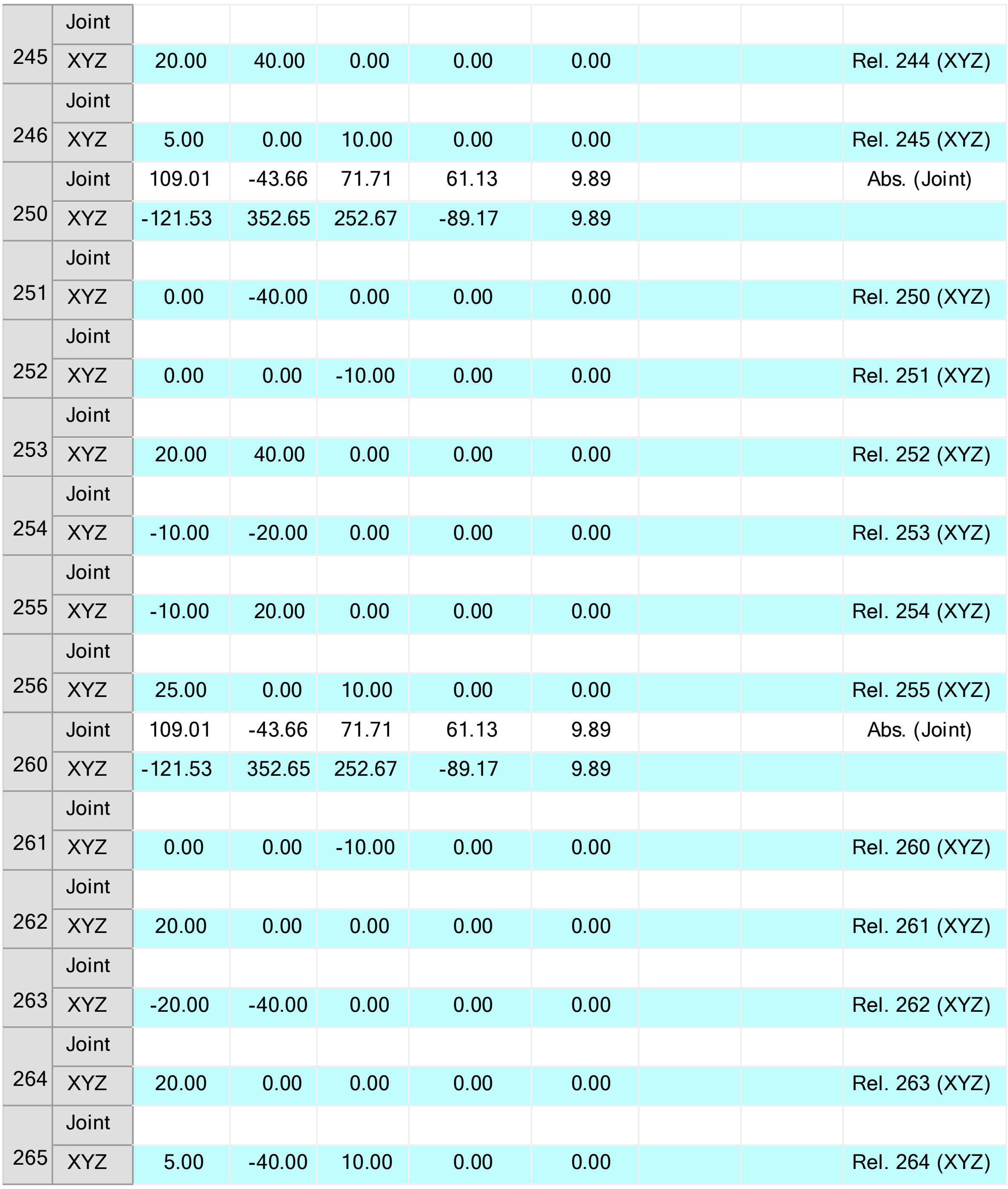
(continued on following page)

Table 11. Continued.

220	Joint	92.05	-22.31	46.53	65.74	91.08			Abs. (Joint)
	XYZ	-15.11	421.82	197.19	-89.96	91.08			
221	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 220 (XYZ)
222	Joint								
	XYZ	10.00	-40.00	0.00	0.00	0.00			Rel. 221 (XYZ)
223	Joint								
	XYZ	10.00	40.00	0.00	0.00	0.00			Rel. 222 (XYZ)
224	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 223 (XYZ)
230	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
231	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 230 (XYZ)
232	Joint								
	XYZ	5.00	-40.00	0.00	0.00	0.00			Rel. 231 (XYZ)
233	Joint								
	XYZ	5.00	40.00	0.00	0.00	0.00			Rel. 232 (XYZ)
234	Joint								
	XYZ	5.00	-40.00	0.00	0.00	0.00			Rel. 233 (XYZ)
235	Joint								
	XYZ	5.00	40.00	0.00	0.00	0.00			Rel. 234 (XYZ)
236	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 235 (XYZ)
240	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
241	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 240 (XYZ)
242	Joint								
	XYZ	20.00	-40.00	0.00	0.00	0.00			Rel. 241 (XYZ)
243	Joint								
	XYZ	-20.00	0.00	10.00	0.00	0.00			Rel. 242 (XYZ)
244	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 243 (XYZ)

(continued on following page)

Table 11. Continued.



245	Joint								
	XYZ	20.00	40.00	0.00	0.00	0.00			Rel. 244 (XYZ)
246	Joint								
	XYZ	5.00	0.00	10.00	0.00	0.00			Rel. 245 (XYZ)
250	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
251	Joint								
	XYZ	0.00	-40.00	0.00	0.00	0.00			Rel. 250 (XYZ)
252	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 251 (XYZ)
253	Joint								
	XYZ	20.00	40.00	0.00	0.00	0.00			Rel. 252 (XYZ)
254	Joint								
	XYZ	-10.00	-20.00	0.00	0.00	0.00			Rel. 253 (XYZ)
255	Joint								
	XYZ	-10.00	20.00	0.00	0.00	0.00			Rel. 254 (XYZ)
256	Joint								
	XYZ	25.00	0.00	10.00	0.00	0.00			Rel. 255 (XYZ)
260	Joint	109.01	-43.66	71.71	61.13	9.89			Abs. (Joint)
	XYZ	-121.53	352.65	252.67	-89.17	9.89			
261	Joint								
	XYZ	0.00	0.00	-10.00	0.00	0.00			Rel. 260 (XYZ)
262	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 261 (XYZ)
263	Joint								
	XYZ	-20.00	-40.00	0.00	0.00	0.00			Rel. 262 (XYZ)
264	Joint								
	XYZ	20.00	0.00	0.00	0.00	0.00			Rel. 263 (XYZ)
265	Joint								
	XYZ	5.00	-40.00	10.00	0.00	0.00			Rel. 264 (XYZ)

4.4 Using the Stored Segments

In order to allow the robot arm to write, a computer program needs to be written. The program needs to figure out the character that needs to be written, get all the coordinates from the position information table for the corresponding character and send the appropriate commands to the robot which in turn completes the character. The first task can be done by getting an input from a user. That can be an input device or through input/output console. But the software is limited by both the functionalities. So the program is modified to give the commands at the starting of the program code before executing it. The second task can be done by searching the position information table for all the data needed for that specific character. The SCORBASE software sends all the information to the controller to which it generates a sequence of commands that are understandable by the robot. For the third task, once the program has started executing, the controller will automatically send appropriate commands to the robot to move corresponding to the code. The controller detects the postion to which the robot needs to be moved and it sends the respective commands to the motors, and thus moving the robot to that location. Figure 21 shows the experimental setup of the robot writing system. The main blocks like software, controller, robot and the writing surface are highlighted. A sample program code is shown after the figure. The program is written for the word “ROBOT.” The letters are given at the start of the program.

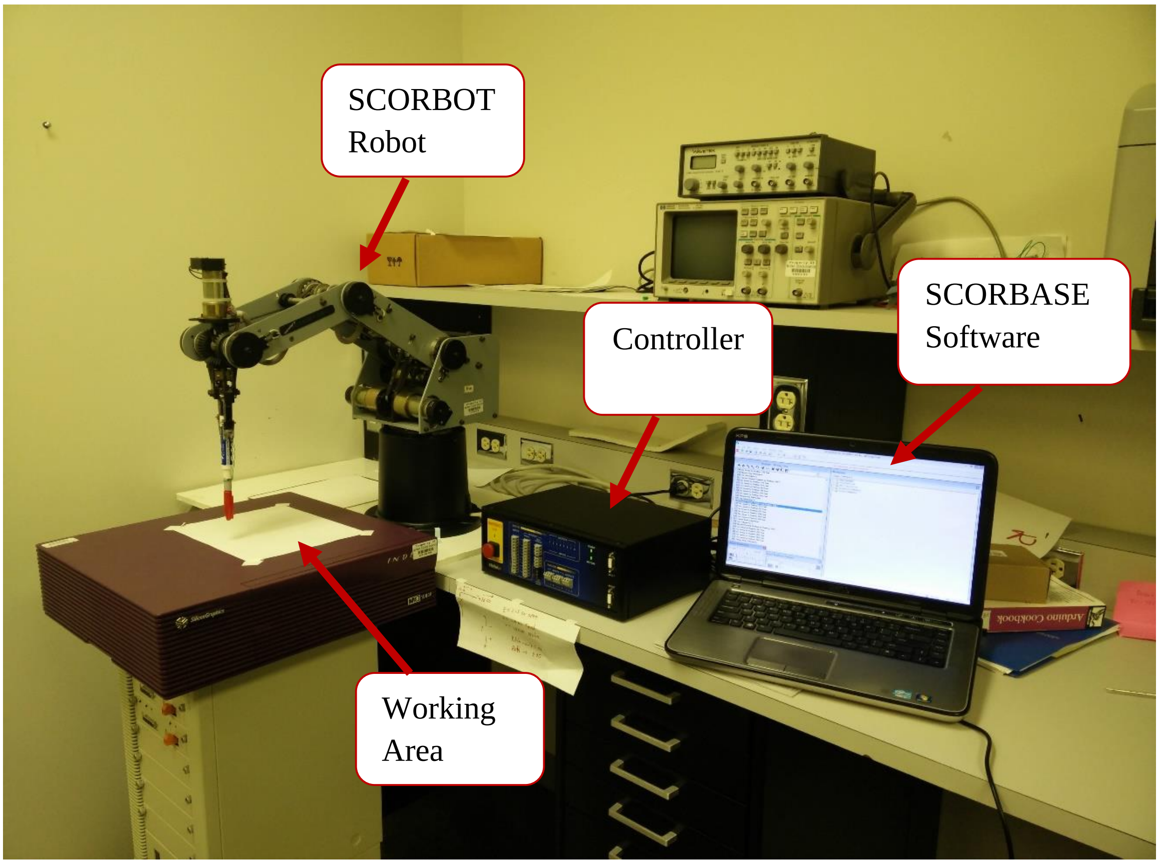


Figure 21: The setup of robot writing system.

The Program Code:

Remark: Enter the letters that needs to written:

Call Subroutine R

Call Subroutine O

Call Subroutine B

Call Subroutine O

Call Subroutine T

Remark: Main Program Starts Here

Set Subroutine A

Record Present Position as Position 10 !

```
Go Linear to Position 11 Fast
Go Linear to Position 12 Fast
Go Linear to Position 13 Fast
Go Linear to Position 14 Fast
Go Linear to Position 15 Fast
Go Linear to Position 16 Fast
Go Linear to Position 17 Fast
Return from Subroutine
Set Subroutine B
Record Present Position as Position 20 !
Go Linear to Position 21 Fast
Go Linear to Position 22 Fast
Go Linear to Position 23 Fast
Go Circular to Position 25 through 24 Fast
Go Linear to Position 26 Fast
Go Circular to Position 28 through 27 Fast
Go Linear to Position 29 Fast
Return from Subroutine
Set Subroutine C
Record Present Position as Position 30 !
Go Linear to Position 31 Fast
Go Linear to Position 32 Fast
Go Circular to Position 34 through 33 Fast
Go Linear to Position 35 Fast
Return from Subroutine
Set Subroutine D
Record Present Position as Position 40 !
```

```
Go Linear to Position 41 Fast
Go Linear to Position 42 Fast
Go Circular to Position 44 through 43 Fast
Go Linear to Position 45 Fast
Return from Subroutine
Set Subroutine E
Record Present Position as Position 50 !
Go Linear to Position 51 Fast
Go Linear to Position 52 Fast
Go Linear to Position 53 Fast
Go Linear to Position 54 Fast
Go Linear to Position 55 Fast
Go Linear to Position 56 Fast
Go Linear to Position 57 Fast
Go Linear to Position 58 Fast
Go Linear to Position 59 Fast
Return from Subroutine
Set Subroutine F
Record Present Position as Position 60 !
Go Linear to Position 61 Fast
Go Linear to Position 62 Fast
Go Linear to Position 63 Fast
Go Linear to Position 64 Fast
Go Linear to Position 65 Fast
Go Linear to Position 66 Fast
Go Linear to Position 67 Fast
Return from Subroutine
```

```
Set Subroutine G
Record Present Position as Position 70 !
Go Linear to Position 71 Fast
Go Circular to Position 73 through 72 Fast
Go Circular to Position 75 through 74 Fast
Go Linear to Position 76 Fast
Go Linear to Position 77 Fast
Go Linear to Position 78 Fast
Return from Subroutine
Set Subroutine H
Record Present Position as Position 80 !
Go Linear to Position 81 Fast
Go Linear to Position 82 Fast
Go Linear to Position 83 Fast
Go Linear to Position 84 Fast
Go Linear to Position 85 Fast
Go Linear to Position 86 Fast
Go Linear to Position 87 Fast
Return from Subroutine
Set Subroutine I
Record Present Position as Position 90 !
Go Linear to Position 91 Fast
Go Linear to Position 92 Fast
Go Linear to Position 93 Fast
Go Linear to Position 94 Fast
Go Linear to Position 95 Fast
Go Linear to Position 96 Fast
Go Linear to Position 97 Fast
Return from Subroutine
Set Subroutine J
Record Present Position as Position 100 !
Go Linear to Position 101 Fast
Go Linear to Position 102 Fast
Go Linear to Position 103 Fast
Go Linear to Position 104 Fast
Go Circular to Position 106 through 105 Fast
Go Linear to Position 107 Fast
Return from Subroutine
Set Subroutine K
Record Present Position as Position 110 !
Go Linear to Position 111 Fast
Go Linear to Position 112 Fast
Go Linear to Position 113 Fast
Go Linear to Position 114 Fast
Go Linear to Position 115 Fast
Go Linear to Position 116 Fast
Go Linear to Position 117 Fast
Return from Subroutine
Set Subroutine L
Record Present Position as Position 120 !
Go Linear to Position 121 Fast
Go Linear to Position 122 Fast
Go Linear to Position 123 Fast
Go Linear to Position 124 Fast
```

```
Return from Subroutine
Set Subroutine M
Record Present Position as Position 130 !
Go Linear to Position 131 Fast
Go Linear to Position 132 Fast
Go Linear to Position 133 Fast
Go Linear to Position 134 Fast
Go Linear to Position 135 Fast
Go Linear to Position 136 Fast
Go Linear to Position 137 Fast
Return from Subroutine
Set Subroutine N
Record Present Position as Position 140 !
Go Linear to Position 141 Fast
Go Linear to Position 142 Fast
Go Linear to Position 143 Fast
Go Linear to Position 144 Fast
Go Linear to Position 145 Fast
Go Linear to Position 146 Fast
Return from Subroutine
Set Subroutine O
Record Present Position as Position 150 !
Go Linear to Position 151 Fast
Go Linear to Position 152 Fast
Go Circular to Position 154 through 153 Fast
Go Circular to Position 156 through 155 Fast
Go Linear to Position 157 Fast
```

```
Return from Subroutine
Set Subroutine P
Record Present Position as Position 160 !
Go Linear to Position 161 Fast
Go Linear to Position 162 Fast
Go Linear to Position 163 Fast
Go Circular to Position 165 through 164 Fast
Go Linear to Position 166 Fast
Return from Subroutine
Set Subroutine Q
Record Present Position as Position 170 !
Go Linear to Position 171 Fast
Go Linear to Position 172 Fast
Go Circular to Position 174 through 173 Fast
Go Circular to Position 176 through 175 Fast
Go Linear to Position 177 Fast
Go Linear to Position 178 Fast
Go Linear to Position 179 Fast
Go Linear to Position 167 Fast
Return from Subroutine
Set Subroutine R
Record Present Position as Position 180 !
Go Linear to Position 181 Fast
Go Linear to Position 182 Fast
Go Linear to Position 183 Fast
Go Linear to Position 184 Fast
Go Linear to Position 185 Fast
Go Circular to Position 187 through 186 Fast
Go Linear to Position 188 Fast
Return from Subroutine
Set Subroutine S
Record Present Position as Position 190 !
Go Linear to Position 191 Fast
Go Linear to Position 192 Fast
Go Linear to Position 193 Fast
Go Circular to Position 195 through 194 Fast
Go Circular to Position 197 through 196 Fast
Go Linear to Position 198 Fast
Go Linear to Position 199 Fast
Return from Subroutine
Set Subroutine T
Record Present Position as Position 200 !
Go Linear to Position 201 Fast
Go Linear to Position 202 Fast
Go Linear to Position 203 Fast
Go Linear to Position 204 Fast
Go Linear to Position 205 Fast
Return from Subroutine
Set Subroutine U
Record Present Position as Position 210 !
Go Linear to Position 211 Fast
Go Linear to Position 212 Fast
Go Circular to Position 214 through 213 Fast
Go Linear to Position 215 Fast
Go Linear to Position 216 Fast
Return from Subroutine
Set Subroutine V
Record Present Position as Position 220 !
Go Linear to Position 221 Fast
Go Linear to Position 222 Fast
Go Linear to Position 223 Fast
Go Linear to Position 224 Fast
Return from Subroutine
Set Subroutine W
Record Present Position as Position 230 !
Go Linear to Position 231 Fast
Go Linear to Position 232 Fast
Go Linear to Position 233 Fast
Go Linear to Position 234 Fast
Go Linear to Position 235 Fast
Go Linear to Position 236 Fast
Return from Subroutine
Set Subroutine X
Record Present Position as Position 240 !
Go Linear to Position 241 Fast
Go Linear to Position 242 Fast
Go Linear to Position 243 Fast
Go Linear to Position 244 Fast
Go Linear to Position 245 Fast
Go Linear to Position 246 Fast
Return from Subroutine
```

```
Set Subroutine Y
Record Present Position as Position 250 !
Go Linear to Position 251 Fast
Go Linear to Position 252 Fast
Go Linear to Position 253 Fast
Go Linear to Position 254 Fast
Go Linear to Position 255 Fast
Go Linear to Position 256 Fast
Return from Subroutine
Set Subroutine Z
Record Present Position as Position 260 !
Go Linear to Position 261 Fast
Go Linear to Position 262 Fast
Go Linear to Position 263 Fast
Go Linear to Position 264 Fast
Go Linear to Position 265 Fast
Return from Subroutine
```

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Testing the Program

After discussing all the steps required for the robot to complete the task, an experiment was set to test the practicality of the idea with the robot. For the robot to start function, we need to set some initial parameters in the SCORBASE software. The first thing is to set an initial position for the robot on the writing surface. By selecting the manual movement of the robot, an initial position can be set. The aim here is to set the robot's arm position at the top left of the page. Based on the positon of the robot and the working area, the initial axis values are given below:

- X- Axis Displacement = -65.00mm.
- Y-Axis Displacement = 425.00mm.
- Z- Axis Displacement = 203.00mm.
- Pitch = -90.00 deg.
- Roll = 0.00 deg.

The word "ROBOT" was given to write and the results are discussed. When the first testing was analyzed, some errors were found. The reason behind these errors and the countermeasures are also discussed here. Figure 22 shows the initial results of the robot writing the letters "ROBOT."

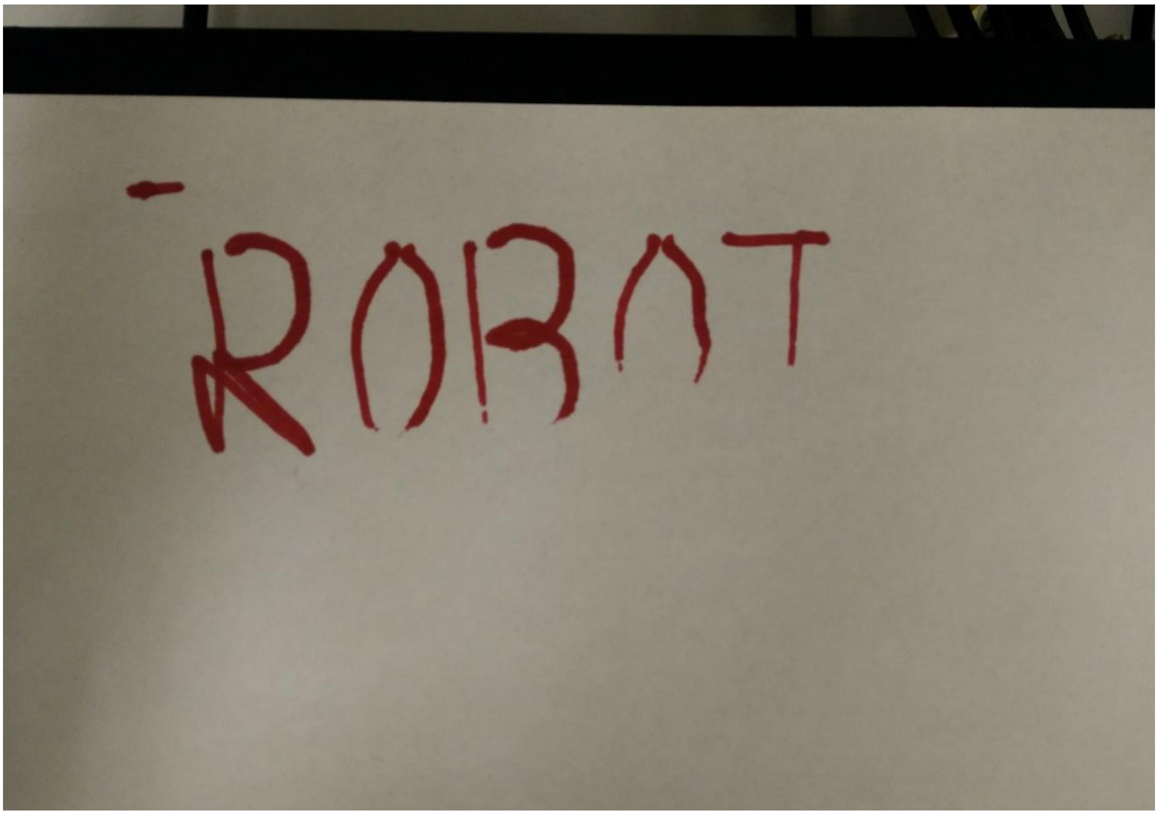


Figure 22: The initial testing of the word “ROBOT.”

5.2 Error Correction

As you can see in Figure 22, the top portion of the letters are written as expected. But the bottom portion had some difficulties in writing. This error is caused by motor movement. As we know, that the robot is moved horizontally and vertically by five motors. When the robot is instructed to move from one position to another, the controller uses one or all the motors to rotate and thus causing the pen to move to that location. But the pen didn’t exactly go to that location as planned. There was a slight error of 1-2mm displacement from desired position. This may have been because of the weight of the links or motor inertia. This motor movement error caused the robot arm to leave the paper without writing some portions of the writing. This error was eliminated by using a spring-loaded pen instead of a regular pen. This spring-loaded pen basically is a pen supported by a spring at the end of it. This setup is covered by a plastic hosing causing the pen move in and out of the hose smoothly when pressure is applied on it. This to and fro motion will eliminate the motor movement error. Figure 23 shows the setup of the spring-loaded pen. The testing of all the characters is shown in Figure 24 and 25. The writing test was again done using the modified pen and the result is shown in Figure 26.

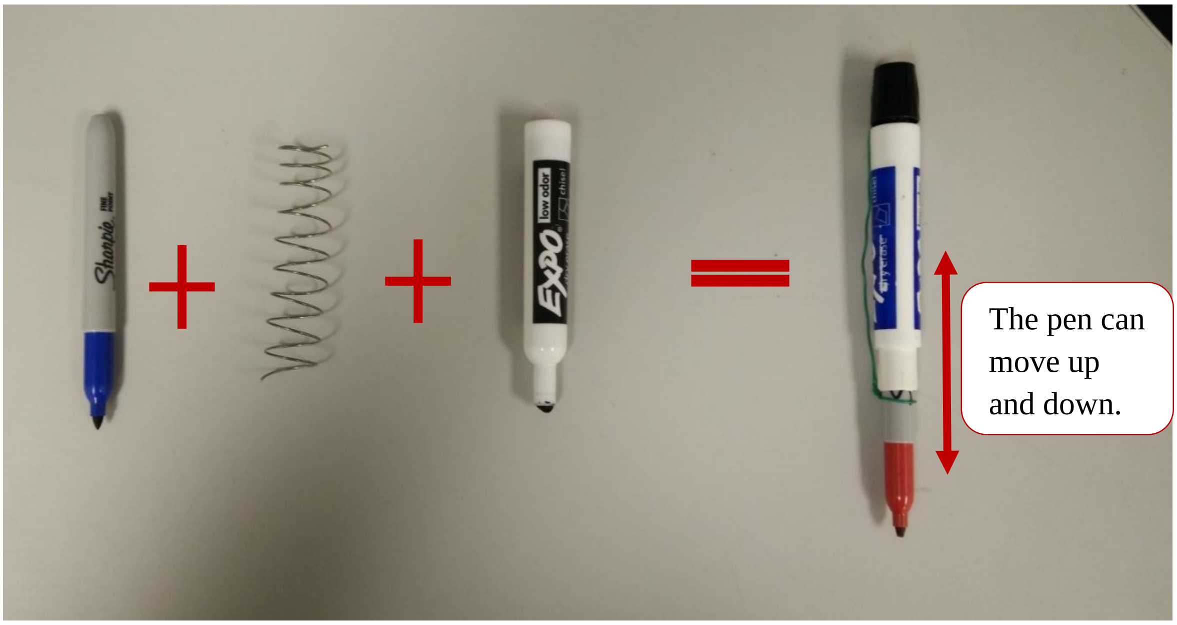


Figure 23: The setup of spring-loaded pen.

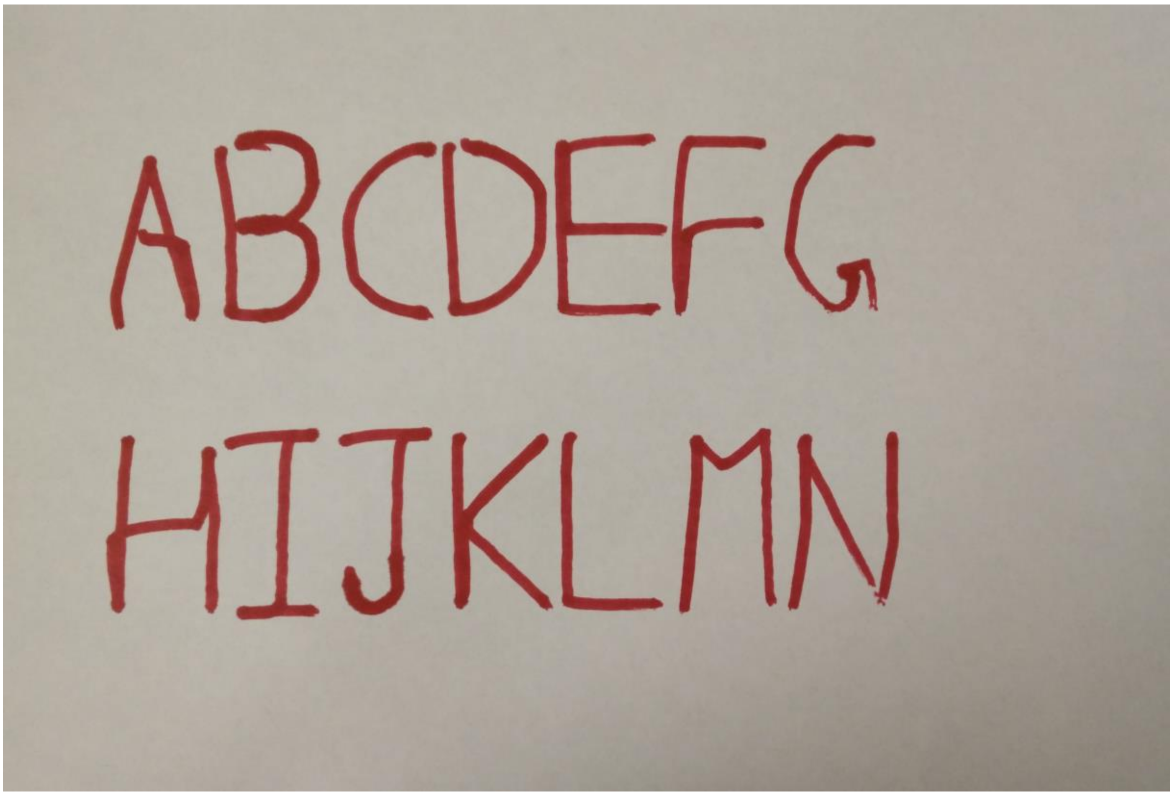


Figure 24: Testing of letters A to N.

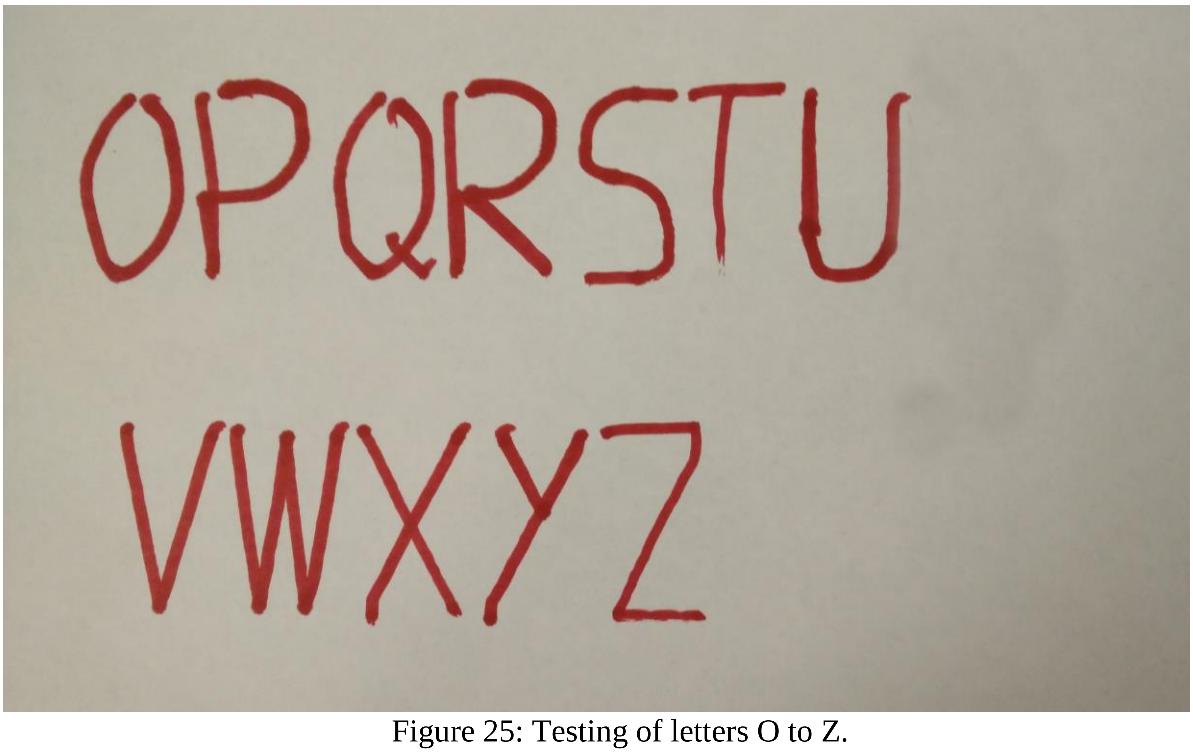


Figure 25: Testing of letters O to Z.

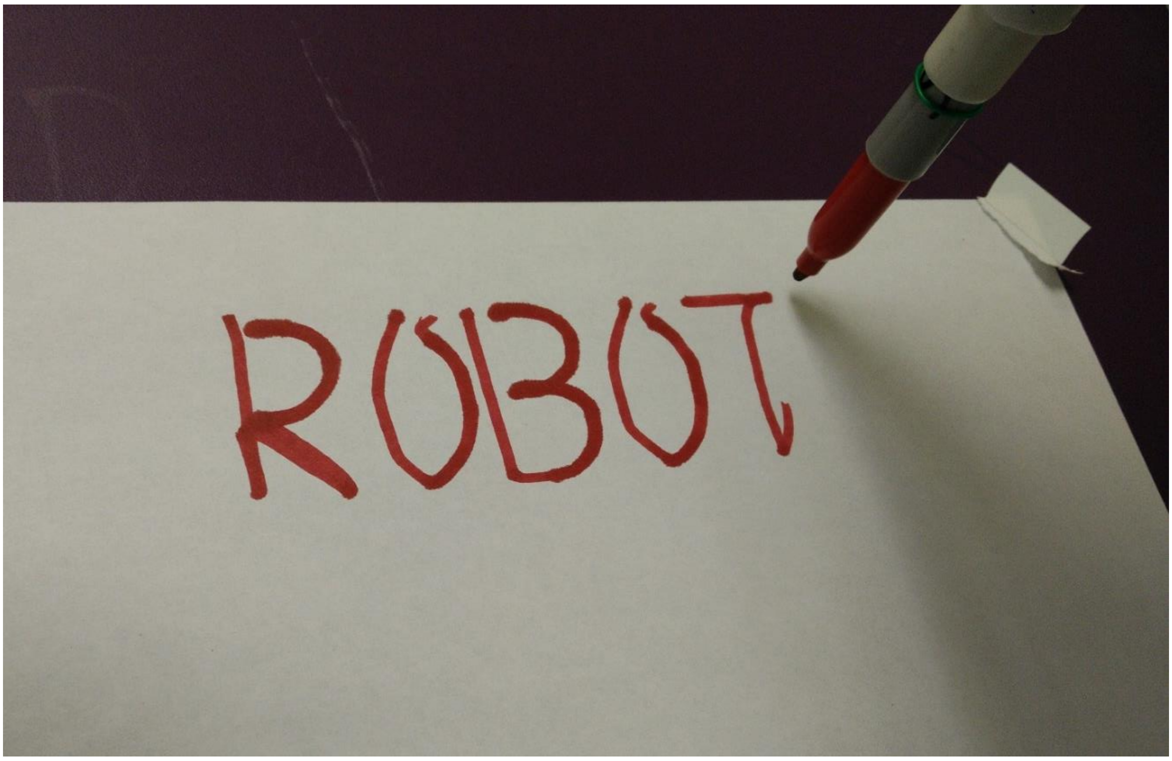


Figure 26: Testing of word “ROBOT” after using the modified pen.

The software has a capability to trace the error caused from the motors. These graphs are plotted and shown here to acquire a better knowledge of the robot movement error. The Figures 27, 28, 29 shows the error graph of each axis (X, Y, Z) respectively when the robot wrote the word “ROBOT.” Similarly Figures 31, 32, 33 shows the error graph of each axis (X, Y, Z) respectively when the robot wrote the word “SCORBOT.”

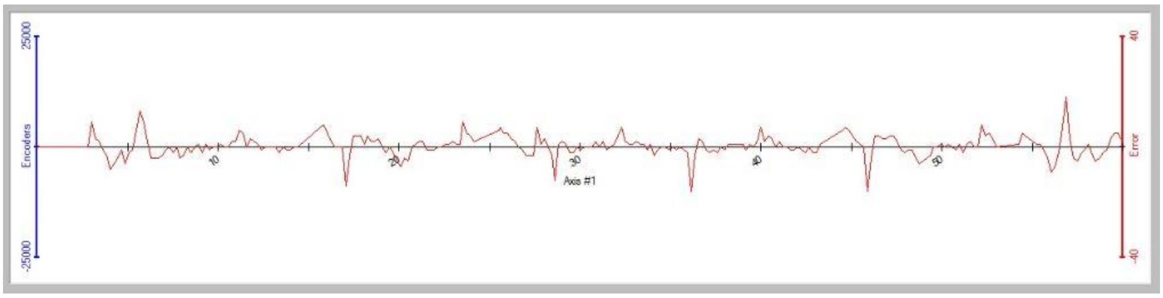


Figure 27: Error graph in Axis 1 for the word “ROBOT.”

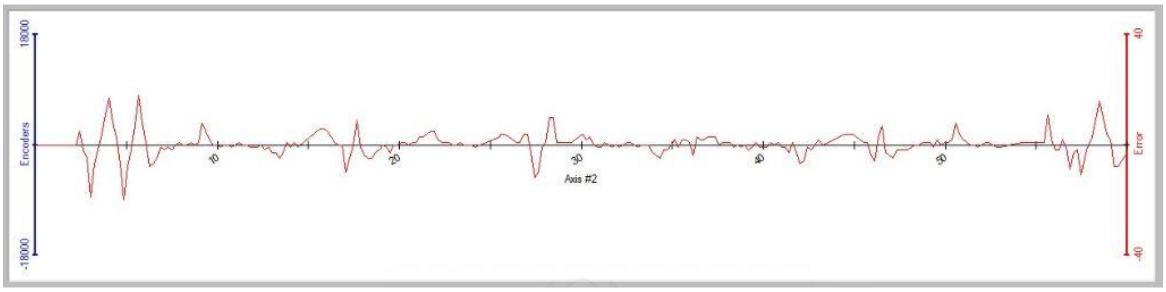


Figure 28: Error graph in Axis 2 for the word “ROBOT.”

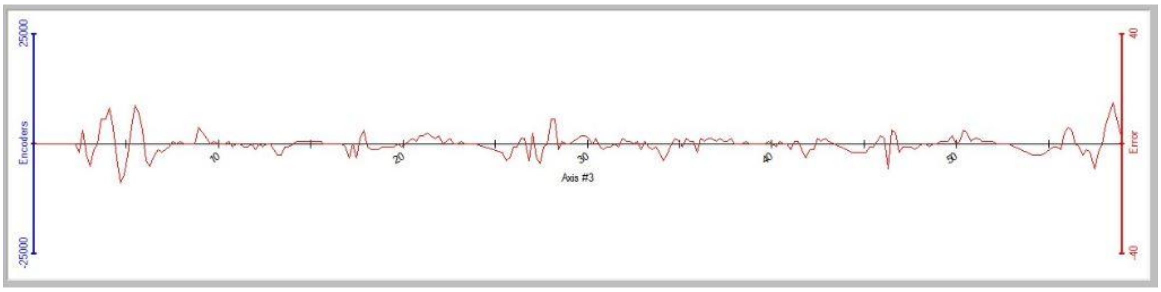


Figure 29: Error graph in Axis 3 for the word “ROBOT.”

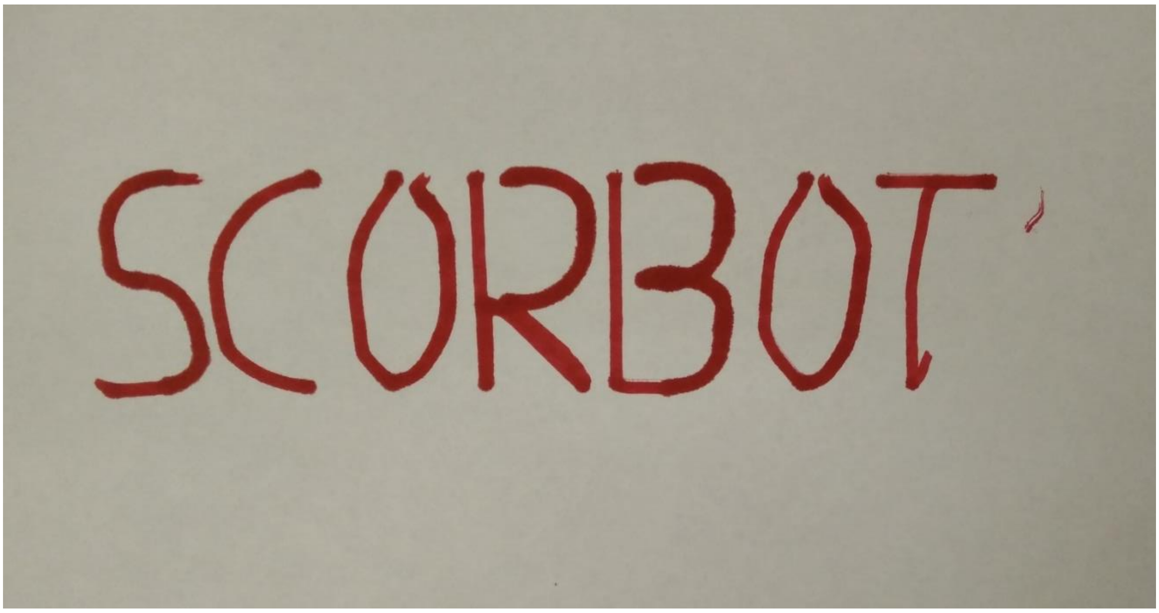


Figure 30: Testing of the sample word “SCORBOT.”

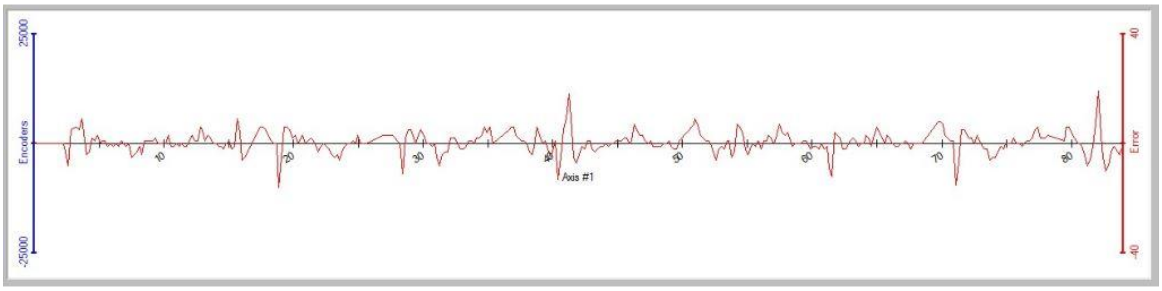


Figure 31: Error graph in Axis 1 for the word “SCORBOT.”

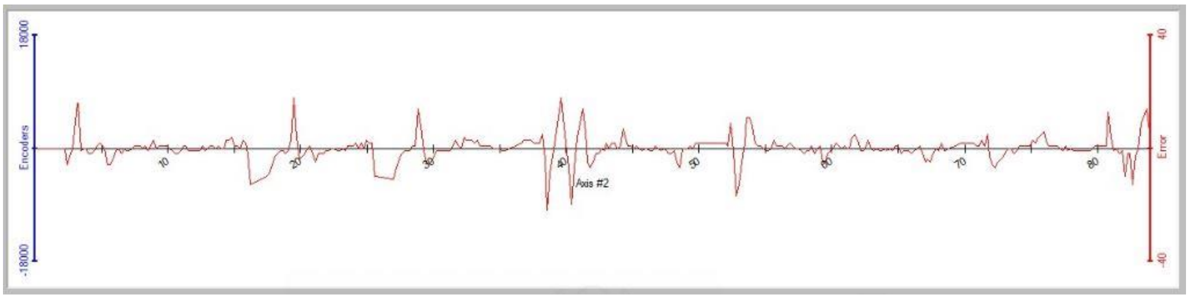


Figure 32: Error graph in Axis 2 for the word “SCORBOT.”

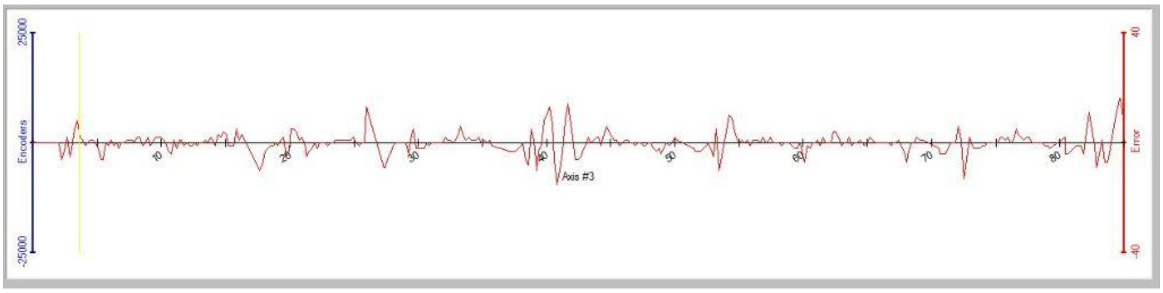


Figure 33: Error graph in Axis 3 for the word “SCORBOT.”

CHAPTER 6

CONCLUSION AND FUTURE WORK

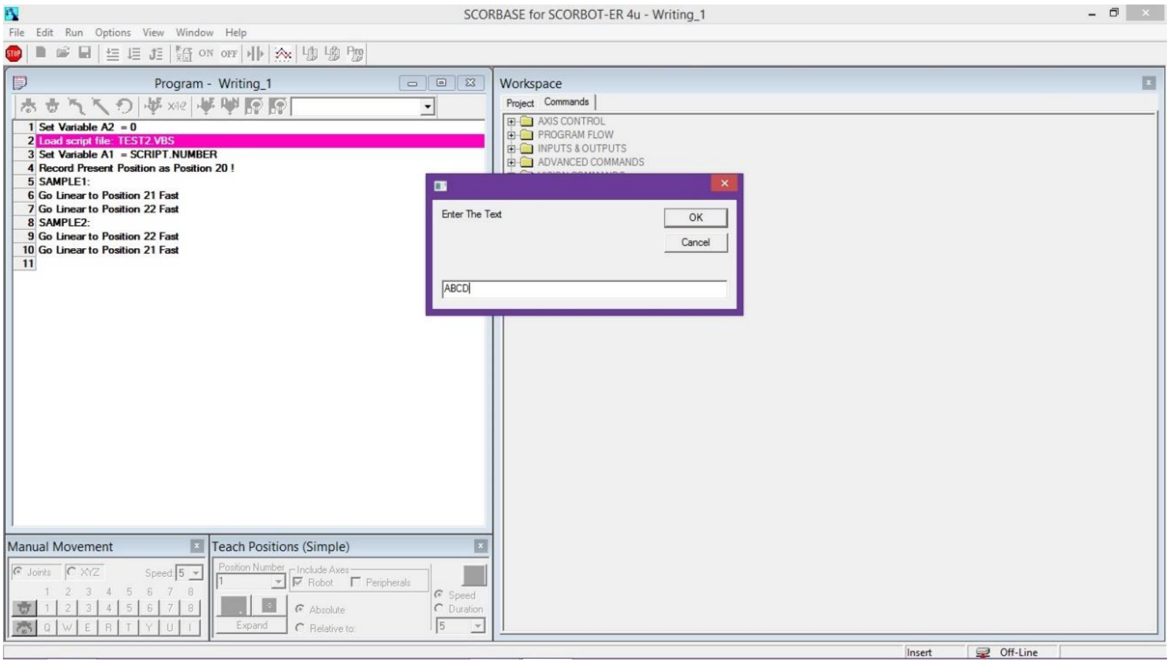
6.1 Conclusion

The idea of writing using SCORBOT-ER 4u has been proposed and demonstrated in this thesis. The proposed method of segmenting alphanumeric characters has been successfully designed and as well as a method to use them in an optimized way. By integrating these concepts together, it is possible to write a program to enable the robot arm to write characters. A SCORBASE program has been developed to test the algorithm. The application performed successfully with minor errors and countermeasures were taken eliminate them and thus the proposed algorithm has been verified. This thesis shows a clear approach that robots can help disabled people to write.

6.2 Limitations

One of the few limitations of this thesis is the limited functionality of the SCORBOT controller. It doesn't have any possibility of acquiring input from external devices. It even doesn't have a console window to give inputs during runtime. So this thesis is limited to inputs directly in the program before executing it. Although the controller has eight digital input lines, they cannot be efficiently used because the controller can access only one digital input line at a given time. One of the possible solutions might be to use Visual Basic (VB) scripts. The SCORBASE software can execute VB script files. Visual

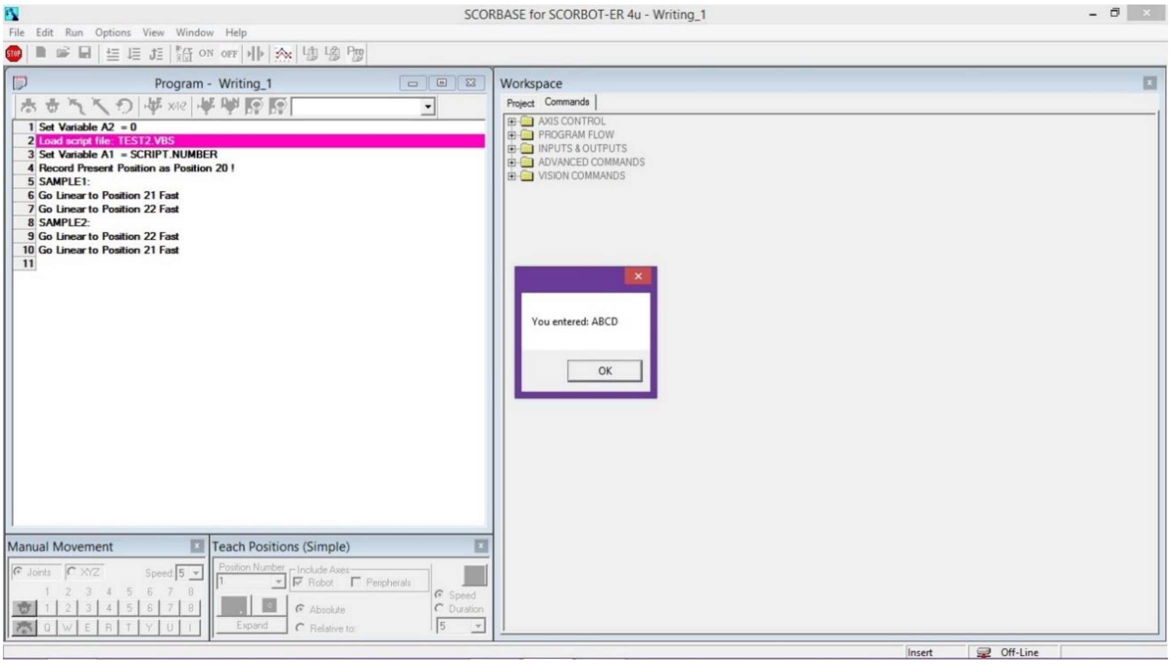
Basic script files can be used to create an input/output console to get inputs from real time. Some initial work has been done to achieve this console window using Visual Basic scripts. A console window has been created that takes the letters that are instructed to it. But the main difficulty lies in incorporating these Visual Basic variables into the SCORBASE program. The main disadvantage of the SCOBASE software is the VB variables cannot be used to call the subroutines in the SCORBASE program. So even after we tried to get inputs from the console using VB Scripts, we couldn't connect them to the program variables. Figure 34 is a screenshot of the program taking letter inputs and Figure 35 shows a window displaying the written text.



```
Set Variable A2  = 0
Load script file: TEST2.VBS
Set Variable A1  = SCRIPT.NUMBER
Record Present Position as Position 20 !
SAMPLE1:
Go Linear to Position 21 Fast
Go Linear to Position 22 Fast
SAMPLE2:
Go Linear to Position 22 Fast
Go Linear to Position 21 Fast
```

Enter The Text: ABCD

Figure 34: A screenshot of program taking inputs using I/O console.

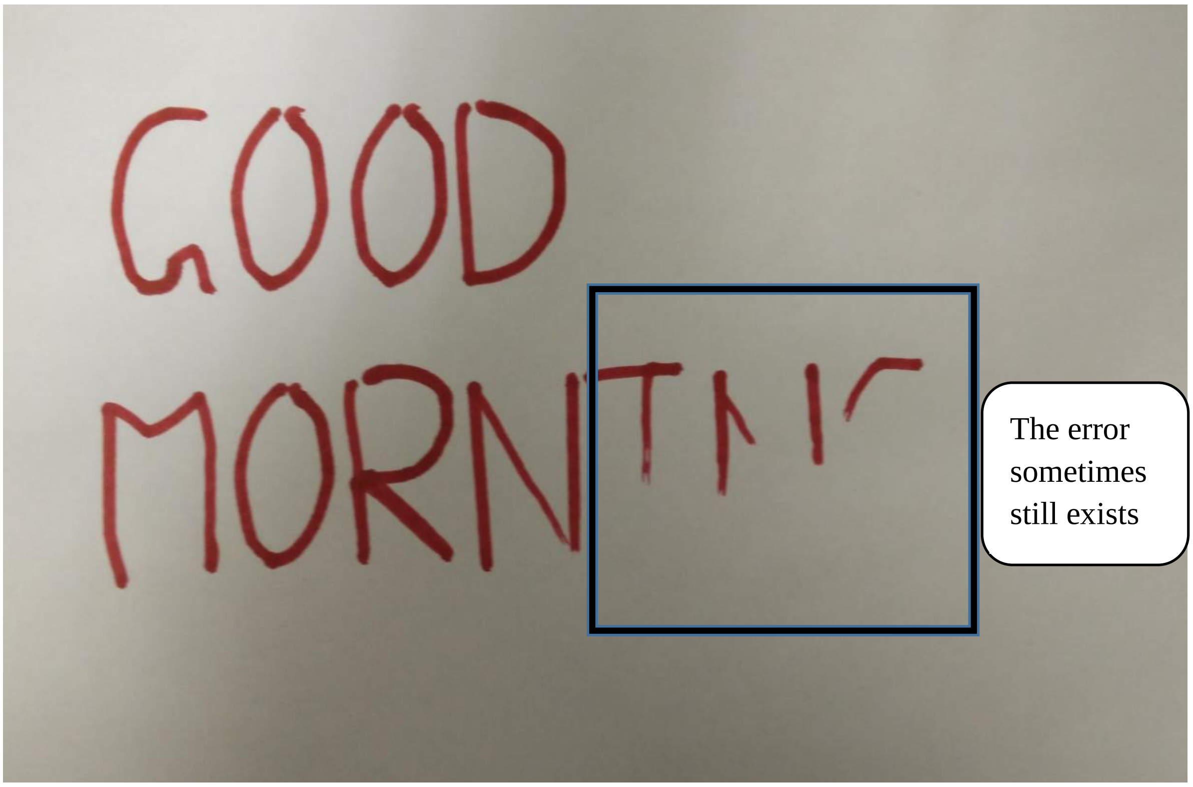


```
Set Variable A2 = 0
Load script file: TEST2.VBS
Set Variable A1 = SCRIPT.NUMBER
Record Present Position as Position 20 !
SAMPLE1:
Go Linear to Position 21 Fast
Go Linear to Position 22 Fast
SAMPLE2:
Go Linear to Position 22 Fast
Go Linear to Position 21 Fast
```

You entered: ABCD

Figure 35: A Screenshot of program showing the entered letters.

Another limitation of the thesis is the spring-loaded pressure pen. Depending on the position of pen on the paper, sometimes the pressure pen compensates the error very efficiently and the letters are written very accurately. But in a few conditions the spring is not sufficient enough to compensate the error and the pen moves up a little, causing the portions of letters that cannot be written. Figure 36 shows an example of how the pen couldn't process the writing completely. The robot was instructed to write the words "GOOD MORNING"; the word GOOD was written properly, but the word MORNING had some troubles. The robot had trouble in writing the last three letters "ING" because the pen moved up from the paper by a small amount sufficient enough to cause the error.



The error sometimes still exists

Figure 36: The robot error in writing GOOD MORNING.

6.3 Future Work

Another extension of this wok could be done by upgrading the present controller to a new controller which may have the ability to acquire inputs from external devices like camera or audio device. By doing that this thesis can be elevated into a whole new level. A few applications in that category can be using a camera to detect a text from a poster or a board by using image-processing techniques and try to replicate the detected text. Another idea could be by using an audio device to record an audio sample from a user and writing the detected text using speech-processing techniques. This will be most helpful for physically disabled people. They can simply tell the robot what needs to be written and the robot writes it.

Another major area can be replacing the present controller with the LabVIEW controller which was previously done by Makana in the lab (Makana, 2012) and it can be improved by interfacing the remaining encoders and microswitches and developing the same ideas stated above.

Another major step based on the concept of this thesis is by extending this idea to portrait drawing. The drawing can be acquired by an external camera, and using image-processing techniques it can be detected and separated into segments and redrawn using the proposed algorithm in this thesis.

REFERENCES

ESHED ROBOTEC. (December 1988). *SCORBOT-ER V Plus, USER'S MANUAL.* Princeton, NJ: ESHED ROBOTEC INC.

Intelitek. (September 2001). *SCORBOT ER-4u User Manual.* Manchester, NH: INTELITEK INC.

Intelitek. (October 2005). *RoboCell, User Manual* Manchester, NH: INTELITEK INC.

Intelitek. (February 2006). *SCORBASE User Manual.* Manchester, NH: INTELITEK INC.

Intelitek. (February 2007). *Controller USB User Manual.* Manchester, NH: Intelitek INC.

Kanji. (2014). Retrieved from wikipedia: http://en.wikipedia.org/wiki/Kanji

Makana, A. (2012). *Vision feedback for a SCORBOT.* NORTHERN ILLINOIS UNIVERISTY.

Masui, S., & Terano, T. (1994). Calligraphic robot with fuzzy logic. *Proceedings of The third IEEE Conference on Fuzzy Systems*, 1598-1603.

McCloy, D., & Harris, M. (1986). *Robotics An Introduction.* Open University Press.

Michelman, P., & Allen, P. (1993). Compliant Manipulation with a Dextrous Robot Hand. *Proceedings of IEEE International Conference on Robotics and Automation*, 711-716.

ScorbotUserGroup (2007). Retrieved from Yahoo Groups: http://tech.groups.yahoo.com/group/ScorbotUserGroup/message/818

The Scorbot Robotic Arms. (2010). Retrieved from The Old Robots Web Site: http://www.theoldrobots.com/scorbot.html

Yossuf, S., Anuar, A., & Fernandez, K. (2005). Algorithm for Robot Writing Using Character Segmentation. *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, 21-24.