Programming ABB Industrial Robot for an Accurate Handwriting

ABIGO IZABO[1], TARIG FAISAL[1*], MAHMUD IWAN[1],
H M A A AL-ASSADI[2], HANIF RAMLI[2]
[1]Faculty of Engineering, Technology & Built Environment, UCSI University,
Cheras, Kuala Lumpur, 56000, MALAYSIA
[2]Faculty of Mechanical Engineering, Universiti Teknologi MARA (UiTM),
Shah Alam, Selangor, 40450, MALAYSIA
*tarigfaisal@ucsi.edu.my

Abstract:-. Handwriting is considered an important human task, and accordingly it has attracted attention of researchers in the robotics field. This paper devoted to programming of ABB robot to write different characters from A to Z. Additionally, a special end effector instrument had been attached to the ABB robot. The writing instrument had been design and fabricated, also equipped with a cleaning tool. The cleaning tool was added to the end effector to combine two applications of writing and cleaning at different Tool Center Point (TCP) orientation. The RAPID program is uploaded as a backup system into the flex pendant of the robot and then simulated through the IRC5 controller. As a result, the project was able to simulate all the characters in the alphabets at maximum operating speed of the robot. All the paths for each letter were written by the robot accurately. Thus, is one step in enabling a commercial industrial robot on performing some of human abilities.

Key-Words:- ABB Serial Robot, Handwriting, Writing and Cleaning Instrument, RAPID,.

1 Introduction

The Encyclopedia defines Mechatronics as a blend of mechanics and electronics. Mechatronics has however come to mean the synergistic use of precision engineering, control theory, computer science and sensor and actuator technology to design improved products and processes.

A robot is defined as an automatically operated machine that replaces human effort although it may not resemble a human. A robot is designed and meant to be controlled by a computer or similar device. Mechatronics deals with the study of robotics and robotics is the art, knowledge base, and the know-how of designing, applying and using robots in human endeavors. A robotic system consists not just of robots, but also other devices and systems that can be used together with the robots to perform the necessary tasks [1]. The motions of the robot are controlled through a controller that is under the supervision of the computer which is running some type of program. Thus if the program is changed, the actions of the robot will be changed accordingly. Thus the robot is designed to be able to perform any task that can be programmed (within its limit and work envelope) simply by changing the program [1].

The aim of this project is to program a robot to perform the task of a human being, which in this case, was to program a writing robot. This programming was done with an industrial robot (ABB IRB 1410) using Robotstudio software and RAPID GUI programming language. The IRB 1410 ABB robot also has a control module and a flex pendant where all the programming instructions are uploaded and edited.

2 System Description

To illustrate this project, the block diagram of the system is shown in Fig. 1.

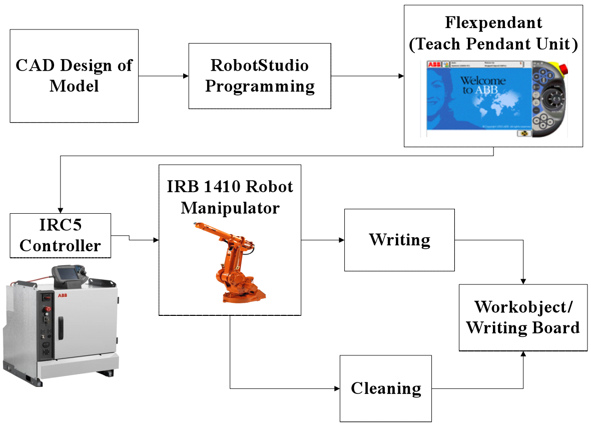

Fig. 1: System block diagram

2.1 The CAD Design Software

Computer Aided Design (CAD) software are used in this project for the design of the end effector and the design of the of the alphabet letters. The design software used was SolidWorks Design software [2]. SolidWorks design software is a feature based parametric solid modeling design tool which uses a graphical user interface to create fully associative 3-D solid models. Solid modeling is the most complete type of geometric model used in CAD systems. During the course of this project a lot of SolidWorks designs were used.

2.2 RobotStudio Programming

Robotstudio is a Graphical User Interface (GUI) software tool used for configuration and offline programming of ABB robots with RAPID programming as the basic programming language. It uses a robot virtual technology that enables the user to be able to watch a single robot execute a RAPID program in the graphical environment. Robotstudio also has modeling features in which the imported CAD files are modeled and path created from the model.

In RobotStudio, a reachability test is done on the robot to actually see if the paths and targets in programming are within the workspace of the robot manipulator [3]. After programming, a virtual controller is also in RobotStudio in which all the paths are synchronized into before off-line simulation. Robotstudio has an off-line simulation tool which can be used to see the motions of the robot during simulation. Any errors in the simulation will also lead to an error in the interfacing [4]. Through this off-line simulation, the program can be debugged before uploading into the flexpendant (Teach Pendant unit).

2.3 Flexpendant

The flexpendant which is also referred to as the teach pendant unit, is a hand held operator unit used to perform many of the tasks involved during the operation of a robot system such as modifying the workobject, jogging the robot manipulator, modifying RAPID programs and can also be used to create instructions using teach method and jogging [5].

2.4 IRC5 Controller

The IRC5 controller contains all the functions required to move and control the robot manipulator. The controller consists of two modules, the control module and the drive module. The control module contains all the control electronics such as main computer I/O boards and flash memory. The control module runs all software necessary for operating the robot system.

The drive module contains all the power electronics supplying the robot motors. An IRC5 drive module may contain nine drive units and can handle six internal axes plus two or additional axes depending on the robot model [6].

2.5 IRB 1410 Manipulator

The IRB 1410 is a 6-axis floor mounted industrial robot, designed specifically for manufacturing industries that use flexible robot-based automation. The robot is equipped with an operating system called BaseWare OS (Operating System). This BaseWare OS controls the motions, development and execution of programs.

The IRB 1410 robot manipulator has a weight of 225 kg and a repeatability position accuracy of 0.05mm (average result from ISO test). The average speed of the robot based on TCP velocity is 2.1 m/s. The robot uses a voltage supply of 200 – 600 V with 50 or 60 Hz frequency with a power rating of 4 Kva/7.8 kVA with external axis and the robot has a handling capacity of 5kg payload and a reach of 1.44m. During the paths and target programming of motions, the robot can attain a speed of up to 7000mm/s during full automatic, the robot speed is unlimited [7]. The workspace diagram is shown in Fig. 2. Table 2 shows the D-H table for the manipulator.

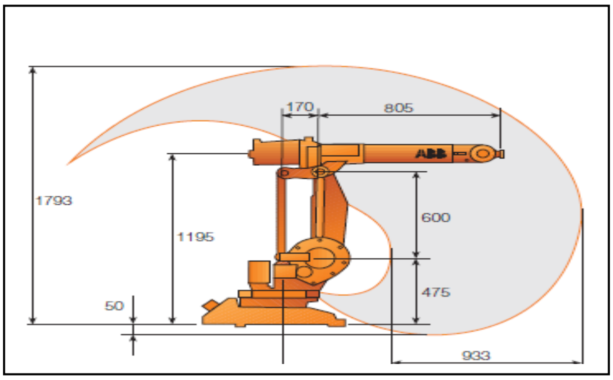

Fig. 2: IRB 1410 showing the workspace diagram

Links	$\theta_i(°)$	$\alpha_{i-1}(°)$	$\alpha_{i-1}(mm)$	$d_i(mm)$
1	θ	90	0	475
2	θ	0	150	0
3	θ	90	600	0
4	θ	-90	120	720
5	θ	90	0	0
6	θ	0	0	85+d

Table 1: D-H Table of the IRB 1410

2.6 End Effector for Writing and Cleaning

The end effector is connected to the last joint of the manipulator (IRB 1410). The end effector used in this project was a writing end effector which was designed with CAD software for use in the programming and also hardware design of the end effector for attaching to the manipulator for Robot interfacing of the program.

As shown in Fig. 3, the end effector was designed with dimensions of 100 mm x 70 mm x 70 mm with a extruded cut hole of 10 ± 5mm diameter for the pen. The length of the pen is about 130 mm with an outer diameter of 10 ± 5mm. the right view of the end effector has an open section cut through with a dimension of 80 x 50 x 50 mm. For attaching the end effector to the robot, an extruded circle of diameter 60 mm with an extrusion of about 8mm and an extrude of length 50 mm with a diameter of 30 mm. The end effector was later modified and a cleaning tool was added at the side. The cleaning tool dimensions are 140 x 55 mm with an extrusion of 30mm. and a block with a dimension of 70 x 100 x 20 mm.

3 Project Development

A CAD design of the writing tool and the cleaning tool were designed separately with SolidWorks Design software and then assembled in using SolidWorks assembly before being imported into an existing system in Robotstudio where the writing and cleaning tool are created for use in the RAPID programming. After the tool (end effector) is created, a writing workstation is created in RobotStudio where the position of the workobject (writing board) in RobotStudio is set within the robot's workspace (x =800, y = -315, z = 420 mm).

The CAD design of the letters are also imported into the RobotStudio and positioned on the writing workobject at the same position initially set. The letters are modeled using and a border is created around the bodies. Path and targets are then created from the modeled letters and the instructions for the paths are converted to RAPID programming language.

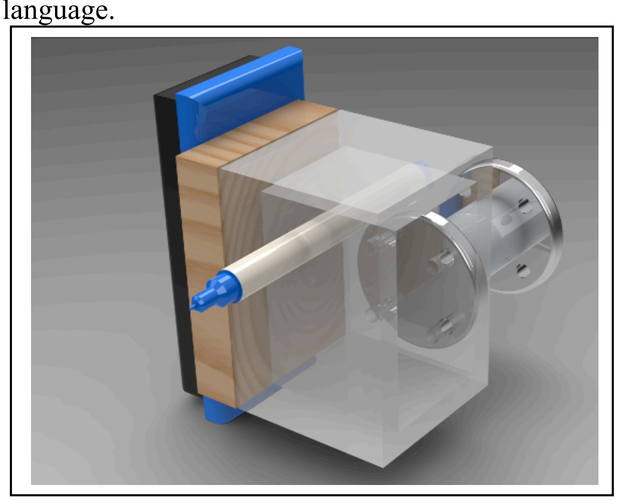

Fig 3: The designed end effector

After programming the program is uploaded into the flexpendant through a backup folder and then the instructions are uploaded in the controller which controls the motion of the speed of the actuators of the robot manipulator and coordinates the motions along the paths and targets. All the routines and paths of the writing instructions will then be executed by the robot on the workobject after writing; the cleaning instructions will also be executed. The programming flowchart is shown in Fig. 4.

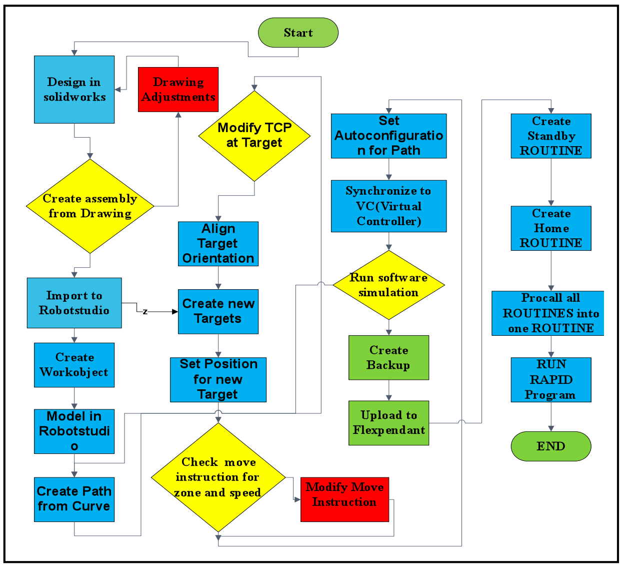

Fig 4: Flow chart for software programming.

3.1 Importing the End Effector into Robotstudio

After the design has been created and assembled in SolidWorks, it is then saved in *.sat or *ACIS format of which both formats can be used in Robotstudio software.

The design is imported in Robotstudio and then created into a tool by aligning it with the origin in Robotstudio so the base to be attached to the robot is on the base axis in RobotStudio, creating a frame at the centre point to be used, modeling it as a tool by filling in the information such as center of gravity and mass saving as a library file in Robotstudio.

3.2 Creating a Workstation in RobotStudio

In this case, a writing board and a platform to support the board had to be positioned to suit the reachability of the robot in its workspace (see Figure 5). A horizontal platform was used since the manipulator workspace enables it. The following measurements were used to set the workstation: Distance of the platform from origin = 900mm, Height of the platform = 38.5mm and Control module distance from origin = 1200mm. Since the workstation had already been created, for every programming to be done in this workstation, the workobject will automatically be the whiteboard [8].

The letters have to be first modeled using any CAD software; in this case SolidWorks was used. In SolidWorks, after the drawing the model, the model is extruded to about 10mm with a height of about 50mm and then using SolidWorks assembly, the letters were assembled using coincident relationship with the board.

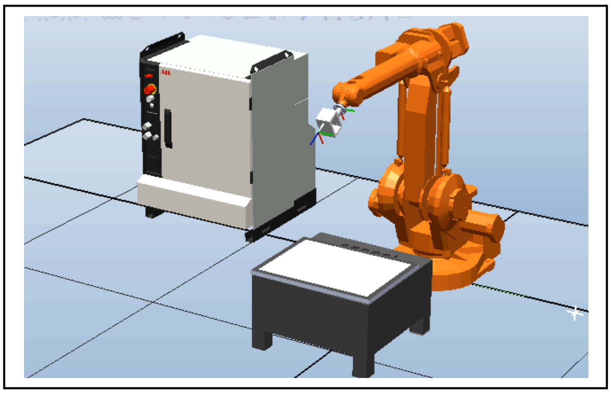

Fig. 5: Writing workstation in RobotStudio.

3.3 Modeling in RobotStudio

Assembly CAD files are of much more importance in Robotstudio modeling since each part is separated from the other parts. For this project, the letters were first drawn separately and then using SolidWorks assembly to assemble the letters [9]. This assembly distinguishes each letter from the next. In RobotStudio modeling, border around bodies was used to create a border around each surface of the letters. The border around surface makes the shape of the part to be on the board thereby enabling a path to be created from that curve.

3.4 Programming the Letters

In programming these letters, path programming was used, the letters are grouped into three (3) categories based on their shape. These categories include:

Fully linear targets: which are A, E, F, H, I, K, L, M, N, T, V, W, X, Y, and Z. Linear plus circular Targets: which are categories B, D, G, J, P, Q, R and U and fully circular targets which include C, O and S.

3.5 Programming the Cleaning Routine

For the cleaning routine, targets were created one point above and another point below the path to be cleaned. The targets were to be created diagonally so that the cleaning tool will move diagonally on the workobject enabling it to be able to move along a path at least twice before moving onto the next path. After the path programming, all the parts are then synchronized in to the virtual controller in RobotStudio [10]. The results are shown in Fig. 6.

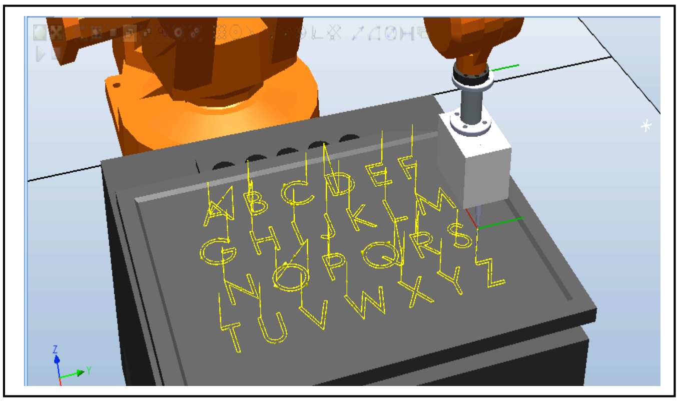

Fig. 6: All the paths set to be synchronized to controller

4 Results and Discussions

4.1 Result

Figs. 7-8 show the picture of the ABB robot with the written letters on the whiteboard and also during the cleaning routione.

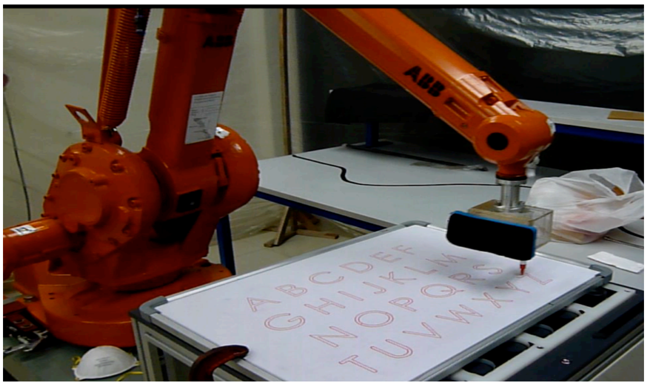

Fig 7: Results from robot interfacing for the letters

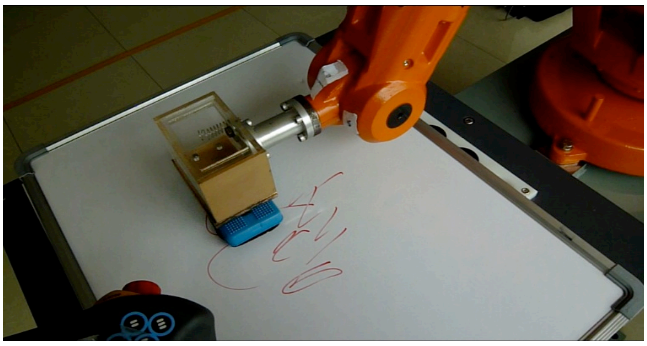

Fig 8: Results from robot interfacing for cleaning

4.2 Discussion

Before the program is uploaded in to the flexpendant, it is first simulated in RobotStudio. The RobotStudio GUI simulation is the same with the interfacing; it shows the actual simulation of the robot moving through the paths and targets.

The program was uploaded into the flexpendant as a backup file and then restored into the system. The workobject was adjusted to enable the writing tool to reach the workobject. A spring during the first interfacing, the robot writing was faint. This was because the workobject in RobotStudio and the workobject used in the interfacing had different orientations. The robot could only write some paths and other paths were partly written. This problem was solved by attaching a spring at the rear end of the writing tool and the end effector to enable the end effector to reach all points.

In the flexpendant, a new routine was created using teach method to jog the robot to two positions, the standby position and the home position. This ensured that after one line of the program before going to the next line (from 'F' to 'G') the robot will move linearly (MOVEL) to point about 600 mm above the workobject and then move down to continue on the next available path.. The zone data was modified for the paths since linear targets with a sharp curve such as "A, F, H, etc require a zone of fine and circular targets such as "O, P, S, etc require a zone from the range of 10mm (z10) to 200mm (z200) depending on the radius of the offset at the edges, see Table 2 and Fig. 9.

Zone data	Value
fine	The robot will go exactly to the specified position
z10	The robot path can cut corners when it is less than 10 mm from the Topoint (destination point)
z50	The robot path can cut corners when it is less than 50 mm from Topoint (destination point)

Table 2: Zone data for two points in RAPID program

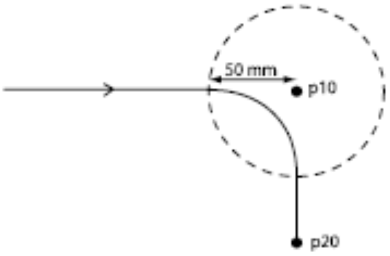

Fig: 9 zone data for two points in a RAPID program

The following RAPID instruction will result in the robot path shown below:

MoveL Target p10, v7000, **z50**, tFYPPEN;
MoveL Target p20, v7000, **fine**, tFYPPEN;

For the robot to move to p10, z50 is used thereby making a curve at 50 mm before p10 and then using a fine (zone fine) at p20 not making any curve [11].

All the 26 letters of the alphabets were successfully programmed and both the offline and Robot Interfacing were successful.

Using the same methodology, I was able to design other models of different sizes by using CAD software and programming the letters by creating paths from the curves. This method proved to be efficient when the sentence "UCSI UNIVERSITY THE UNIVERSITY THE EXPERIENCE" was programmed and an interfacing was done on the robot. The robot was able to move through all the programmed paths at maximum speed. Despite the sizes of the letter, the robot was able to have the same accuracy when writing each letter.

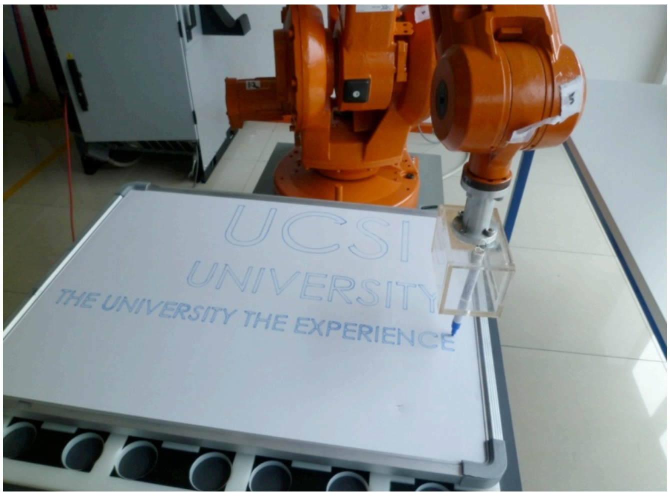

Fig 9: Results from robot interfacing of "UCSI"

5 Conclusion

The aims and objectives that were set at the beginning of this project were successfully achieved including further work on the project. The idea of programming an industrial robot to write letters came to life from the initial idea that was conceptualized. From this project, I was able to design an end effector from CAD drawing and create a tool with two TCP (tool center point position) positions in RobotStudio one for writing all the characters of the alphabet and the other for cleaning. The Robot was able to write with the maximum speed and accuracy for all paths and routines created from the letters and the robot was also able to erase or clean the writing on the same work object.

References:

[1] Saeed B. Niku, Ph.D., P.E *Introduction to Robotics Analysis, Systems and Applications* San Luis Obispo, California, Prentice Hall 2001, pp. 4-26, pp. 29-87, pp. 95-116.

[2] Solidworks*, SolidWorks Essentials Training. 2008.* Pp 1 -250

[3] ABB Robotics *RobotStudio courseware 5.11, 2008. Pp. 1-81*

[4] ABB Robotics*, Operating Manual RobotStudio 5.13,* 2010. Pp 91.

[5] ABB Robotics, *Trouble shooting manual for IRC5 Controller,* 2010. Pp 79 – 283

[6] ABB Robotics *IRC5 Training Manual,* 2011. Pp. 1-190

[7] ABB Robotics *IRB 1410 robot Manual,* 2010. Pp 1-44

[8] Lélio R. Soares Jr. and Victor H. Casanova Alcalde*,* An Educational Robotic Workstation based on the Rhino XR4 robot*, IEEE journal,* 2006, University of Brasilia. pp 1-5.

[9] Mehmet Ismet Can Dede, and Sabri Tosunoglu, “Virtual RAPID Robot Prototyping” *ASME Early Career Technical Journal,* October 2006**,** Florida International University Miami, Florida. Pp. 1-5

[10] Pedro Neto and J. Norberto Pires High-Level Programming for Industrial Robotics: Using Gestures, Speech and Force*, IEEE journal,* 2008, Coimbra Portugal, pp. 1-5.

[11] ABB Robotics*, RAPID instructions, functions and Data types,* 2010. Pp 120- 145