A supervised learning technique for programming a welding arm robot using vision system

Corresponding Author:

Mohamed Hosni Mohamed Ali

Mechanical Engineering Department, Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt

Authors:

*Assoc. Prof. Mostafa Rostom Atia

Mechanical Engineering Department, Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt

*Prof. Farid Abdel Aziz Tolbah

Mechatronics Engineering Department, Ain Shams University, Cairo, Egypt

Abstract

The programming of the welding robot is a challenging problem, especially with complex paths. Extracting path points and suitable welding speed at every path zone is a complicated, time wasting, and costly process. Moreover, the accuracy of extracting these data at the design stage is affected by the inaccuracies in pre-welding processes. This paper introduces a new supervised learning technique for programming a 4 degree of freedom (DOF) welding arm robot with automatic feeding electrode. In this technique, a three-dimensional (3D) machine vision system is developed to grasp the welding position and speed of a complex path by monitoring of an expert welding instructor. Then, these data are used to generate the robot move program. The proposed technique includes fewer steps and hence less consumed time than the conventional one. Moreover, it does not need an expert programmer. From the accuracy point of view, there is no significant difference between the two techniques. These enhancements will improve the share of robots in welding and similar industries.

Keywords: Supervised learning; Robotic arm; Machine Vision.

Introduction

Nowadays, the world of robotic arms collaborative with human workers is expanding in the industrial space [1]. Hence, safety is important in human-robot collaboration [2]. Robots can perform many tasks as well as, or even better than humans [3]. In addition, many applications show the need of integrating between robotic system and computer vision system. The combination between robot system, image processing and tracking algorithm refer to a branch of robot science called visual servoing. This uses feed-back information extracted from a vision sensor to enhance the robot program loop performance. Since it mimics the human sense of vision and allows for noncontact measurement of the environment [4]. Visual servoing can be categorized into image-based visual servoing (IBVS), hybrid approach and position-based visual servoing (PBVS). There are two basic camera configurations. One is to install a camera at the end-effector of the robotic arm (eye-in-hand); the other is to set the robotic arm and camera separately (eye-to-hand) [5].

Moreover, the robot requires an amount of knowledge in programming. Hence, a robot controller programming methods are classified into: teaching pendant, teaching by demonstration, and off-line [6]. Teaching pendant is one of the most common ways for a robot programming as human machine interface (HMI). To program a robot, the operator moves it using the buttons from point-to-point on the pendant and save each position. The robot can redo the process once the whole program has been learned. Almost teaching pendant is connected with a robot controller using cable or wireless [7]. This method might be difficult with who are unfamiliar with programming, which makes it not intuitive. The off-line programming enables the possibility of using simulation technology tools and robot kinematics. That means, the robot is able to be programmed from a computer and tested before moving it to the real world [8]. Industrial applications using robotic arms, requires extra time in programming, enhance its performance of design [9] and obstacle avoidance in the welding path instead of solving production challenges. In order to determine all of the possible movements to follow a planned path or to place the end effector of a robot manipulator at a particular point in space, the movements associated with every joint variable should be computed.

In other words, Kinematics plays an important role in robotics and it could be analysed for robotic manipulator by forward and inverse kinematics [10,11,12]. Forward kinematics will determine where the end effector will be if all joints are known whereas inverse kinematics will calculate what joint variable should be if the desired position and orientation of the tool is determined [13]. On the other hand, lead through programming method is a physical movement of the manipulator by the operator [14,15], while its motion is recorded and duplicated. This method makes the programming simple, faster and intuitive [16].

This paper concerns with the programming difficulties of multi DOF welding robot to weld a complex path. The programmer needs the position of path points and welding speed at every point. Extracting these data in real application is a difficult and time wasting process.

Extracting the path points from computer-aided design (CAD) packages at design stage still needs the welding expert opinion for speed at every point and an expert programmer to generate the robot programming code. The aim of this paper is introducing new programming procedure based on supervised learning technique. The interested welding data will be extracted from the supervising of a welding instructor by a 3D vision system. Based on these data, the robot programming code will be generated automatically.

Experimental setup

As shown in Fig. 1, Experiment framework setup is formed of a Personal Computer (PC), machine vision, pen (120 mm) length attached on the 4 DOF robotic arm end-effector, (590 x 440 mm) white board including the planned path, Arbotix-m robocontroller and Universal Serial Bus (USB) 2 dynamixels. The PC has a processor Intel® core i7-7500U Central Processing Unit (CPU) @ 2.70 Gigahertz (GHz) and 32.0 Gigabyte (GB) Random Access Memory (RAM). Machine vision is connected to pc through USB 3.0 port in eye-to-hand concept (fixed). Programmed with a Software Development Kit (SDK) 2.0 and graphical programming G language, which perceives the depth based on its Time-of-Flight (ToF) technology [17,18], as well as a Red, Green and Blue (RGB) camera to capture videos and images [19]. Labview robotics software is used for applying image processing techniques to recognize the picture, the tip of pen as a welding torch point is still an issue to be located. In order to locate the welding tip, a pen is fixed as a welding torch at the robot arm end-effector. Where, a marker known by the user attached on the pen back. Therefore, the welding point is the pen back. Moreover, the phantom-x pincher robotic arm has 4 revolute joints, it reach (350 mm) vertical and (310 mm) horizontal. Besides that, wires of the actuators on the robot arm are daisy chain connected through USB2Dynamixels. This device is connected directly to the PC via USB port 2.0. However, the manipulator using dynamixel AX-12A servo actuator with built-in microcontroller, which can record, save and play back its motion. Hence, the positions and speed are used to generate robot program. Also, the speed and position can be fine-tuned using the design user interface. The use of this work enables the expert welder directly grasps the robotic arm welding torch to pass from one discrete point to another within the robot’s working envelope. While the movement, the points are recorded upon the operator selection. Moreover, the vision system provides the actual location of every point in the planned path.

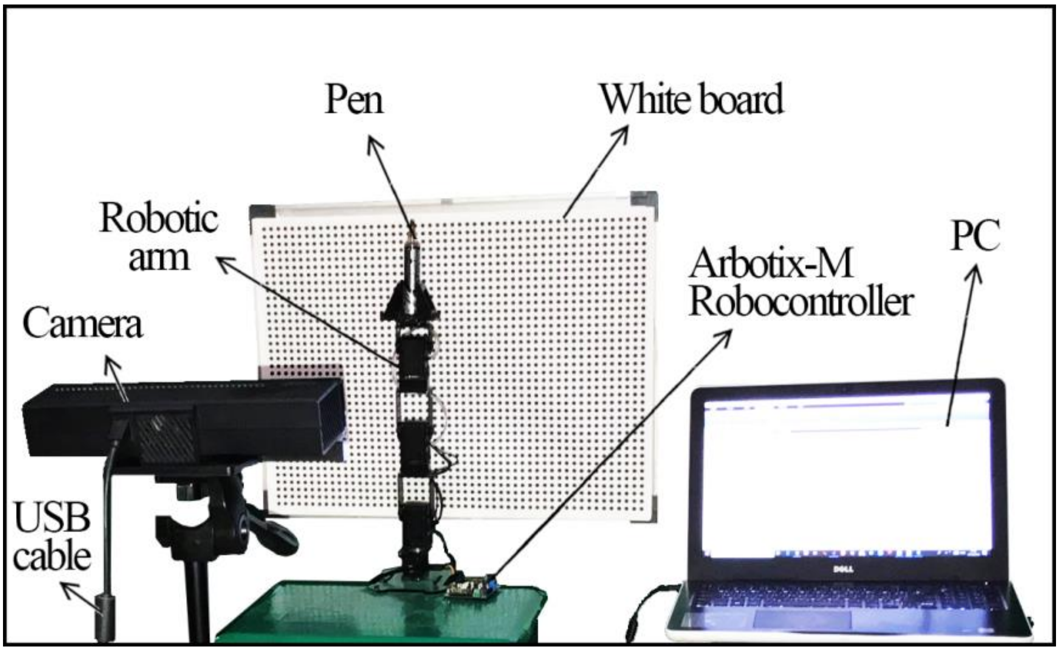


Fig. 1: Experimental setup

Machine Vision

The vision system depends on features of the manipulator specification to insure that the robot arm will deal with the path or not and to make sure that the tip of the manipulator reached the desired point. Hence, the robot programming code is obtained through a 3D machine vision system to capture the points in (x, y) directions which represents the complex path. The robotic arm microcontroller receives the location of points to follow the planned square sine wave path as shown in Fig. 2.

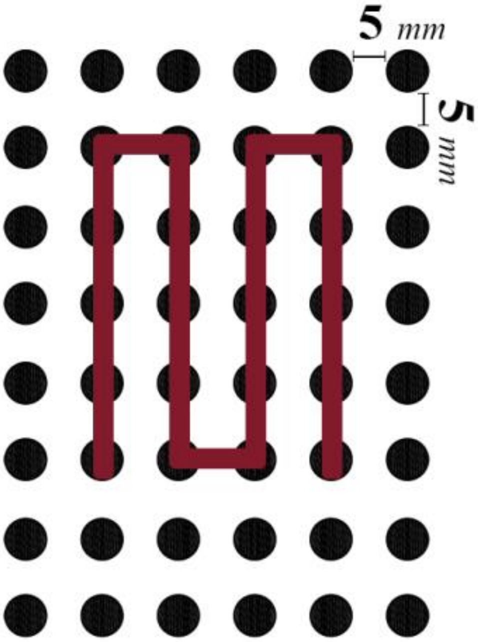
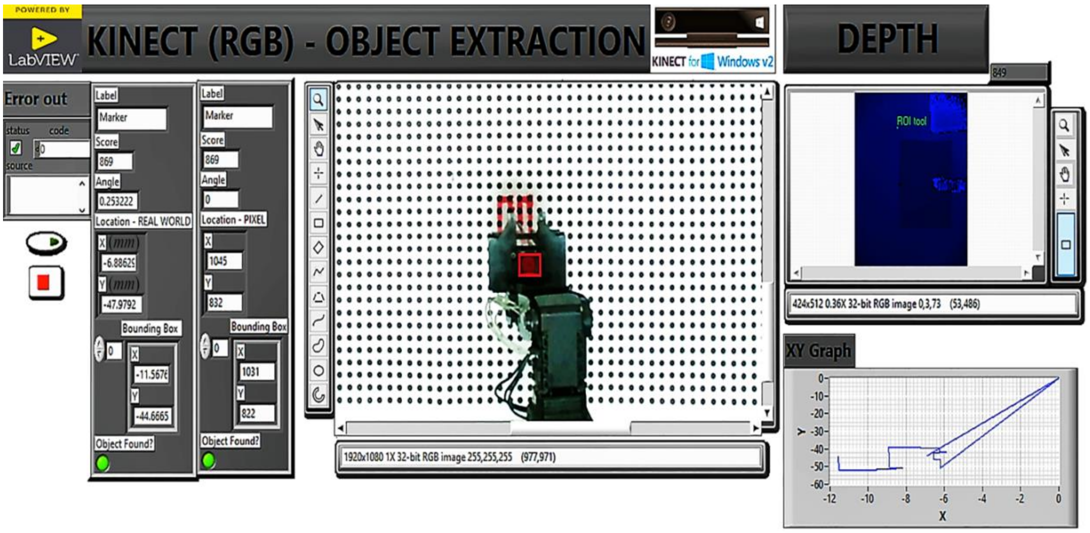


Fig. 2: Planned path

Marker attached at the pen back and grid of points map known by the user are the environment in the machine vision system. Image processing tools recognize the grid of point's image and to locate the end effector location in real world. Hence, object tracking tool is to search for the predetermined marker attached at the end effector and its origin is defined as in an inspection image. The input grid of points and marker are matched then image x, y position displayed in pixels, the pixel value is converted into mm length unit. In this method each point is identified in real world by mm unit as shown in Fig. 3.



Error out
status ☑ code 0
source

Label: Marker
Score: 869
Angle: 0.253222
Location - REAL WORLD
X (mm): -6.88629
Y (mm): -47.9792
Bounding Box: 0
X: -11.5676
Y: -44.6665
Object Found? ●

Label: Marker
Score: 869
Angle: 0
Location - PIXEL
X: 1045
Y: 832
Bounding Box: 0
X: 1031
Y: 822
Object Found? ●

1920x1080 1X 32-bit RGB image 255,255,255 (977,971)

849

424x512 0.36X 32-bit RGB image 0,3,73 (53,486)

Fig. 3: User interface

Kinematics

Fig. 4 shows the four degree of freedom arm robot. Each joint has AX-12A dynamixel servo actuator. All actuators have a running degree (deg) from 0 deg to 300 deg or continuous turn.

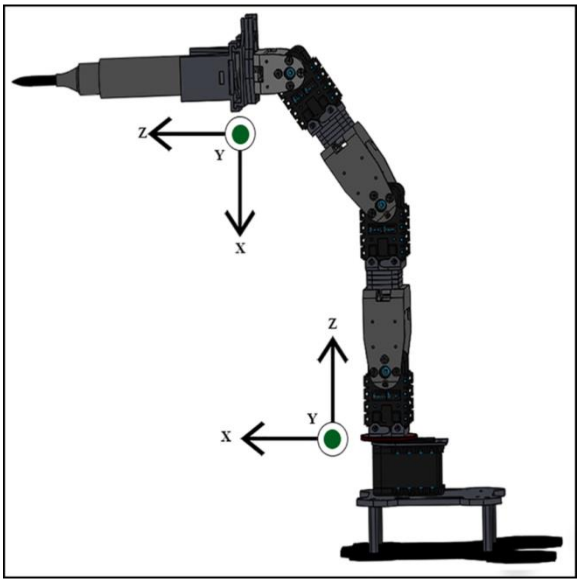


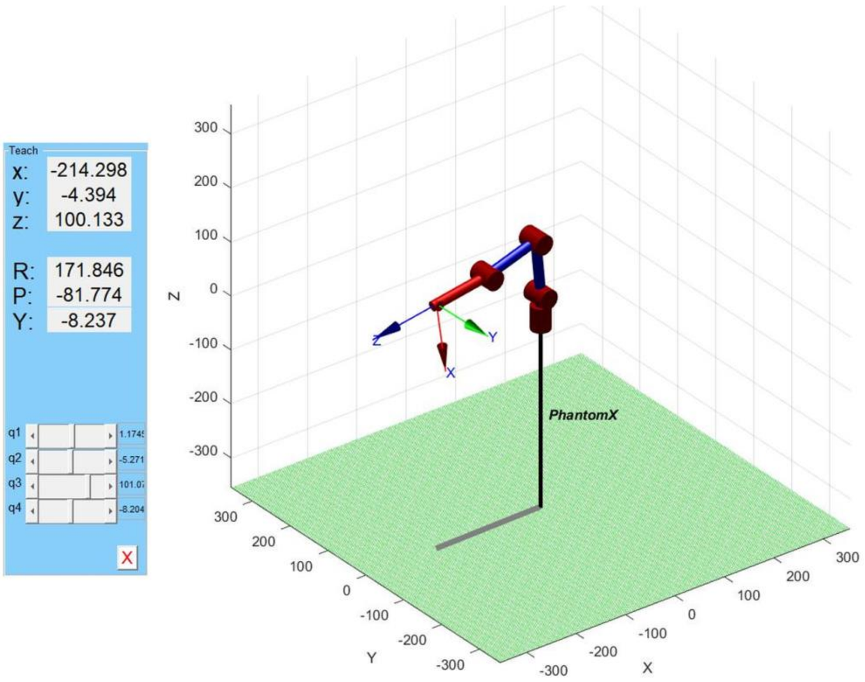
Fig. 4: PhantomX-Pincher - CAD model

Kinematics model of the robotic arm is represented by Denavit-Hartenberg (DH) design. This design is commonly used in the kinematic chain analysis [20]. It is based on attaching a reference system for each of the joint and specifying four parameters known as DH parameters for each link, and using these parameters to construct a DH table as shown in table 1.

Table 1: DH parameters

Joint	(Joint angle) ϴ	(Link offset) *d*	(Link length) A	(Revolute angle) α
1	ϴ1 Base	45 mm	0	0
2	ϴ2 Shoulder	0	150 mm	0
3	ϴ3 Elbow	0	150 mm	90 deg
4	ϴ4 Wrist	0	90 mm	0

The goal of gaining the robot arm forward kinematic is to determine the current pose of the end-effector in order to then find the new position to which the manipulator must move. After estimating this position, the robot’s inverse kinematics is found by using the numerical method [21]. Fig. 5 shows the results implemented in Matlab.



Teach

x: -214.298
y: -4.394
z: 100.133

R: 171.846
P: -81.774
Y: -8.237

q1 1.1745
q2 -5.271
q3 101.07
q4 -8.204

Fig. 5: Robot arm kinematics solution

Experimental work

This paper illustrates a new lead through learning by demonstration technique for programming a serial robot in welding operation. In this technique as shown in Fig. 6, the expert welder holds the robot end effector as a welding torch. He moves through the planed trajectory as he actually welds. Note that, the accuracy of instructor movement is important as well as it effects in proposed technique accuracy and performance investigation.

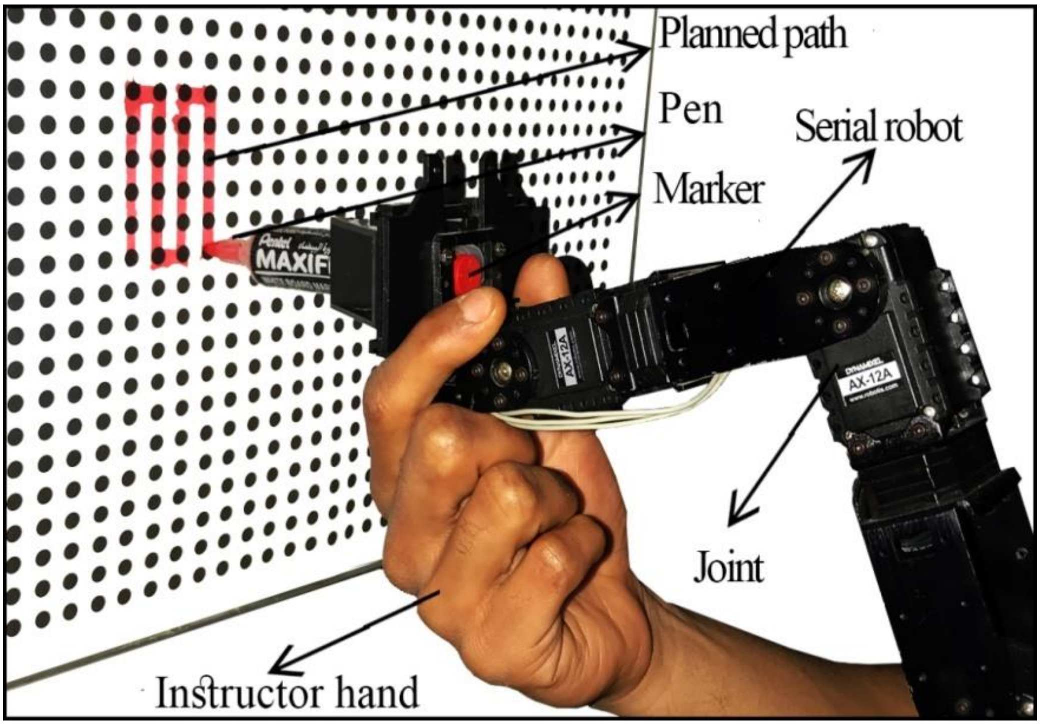


Fig. 6: Proposed technique

The vision system captures the position of the end effector to track the trajectory. While the sensor, which is encapsulated in every robot arm actuator, captures the angle and speed of every joint. The angles and speeds of every arm are stored with the related end effector position. Later, these data are used to generate the robot programme. Hence, the programming data are grasped from a real welder the robot can easily mimics the expert in welding movements.

This technique has advantages over the conventional ones. This is because there is no need to measure the path points from a real workpiece or calculate them from a CAD file. Welding speed theoretical calculations are sophisticated and not reliable as the expert welder. While in the proposed technique, the robot arm angles and angular speed are measured directly even without using inverse kinematics. In addition, the welder is able to avoid obstacles during learning step. These advantages decrease the programming time and effort and increase the accuracy.

Results

This section presents the practical applications of the proposed programming technique. The aim of this experiment is to check the efficiency of the proposed technique from the programming time and accuracy point of views. The applications include programming a 4 DOF robot arm to follow several paths as shown in Figs. 7-8.

The first three paths represent rather simple paths in positive X, Y directions and other three represent the complex paths in positive and negative X, Y directions. For every path, absolute error and executed time were measured. Furthermore for every path, the absolute difference error graph between the real path and the desired path can be clearly distinguishable in mm unit. This absolute error is produced by the machine vision calibration using LABVIEW software to detect the square marker attached at the end of the pen, which representing the welding electrode. In addition, the absolute error validation figures do not follow the same trend in each path because each point has a different absolute error in lens distortion.

The first set of paths is distinguished by simple move in one direction has no effect of the backlash error in robot hardware. They are vertical line shape #1, Horizontal line shape #2 and stairs like line shape #3. In general, the first point of every path has a high absolute error value. This is because the robot is in the arbitrary point before reaching the desired point.

Then in Fig. 8, all of the experiments applied on the second set are complex shapes. Such as square shapes .The robot backlash is observed and the absolute error increases in turn points then decreases as shown in path #4, path #5 and path #6.

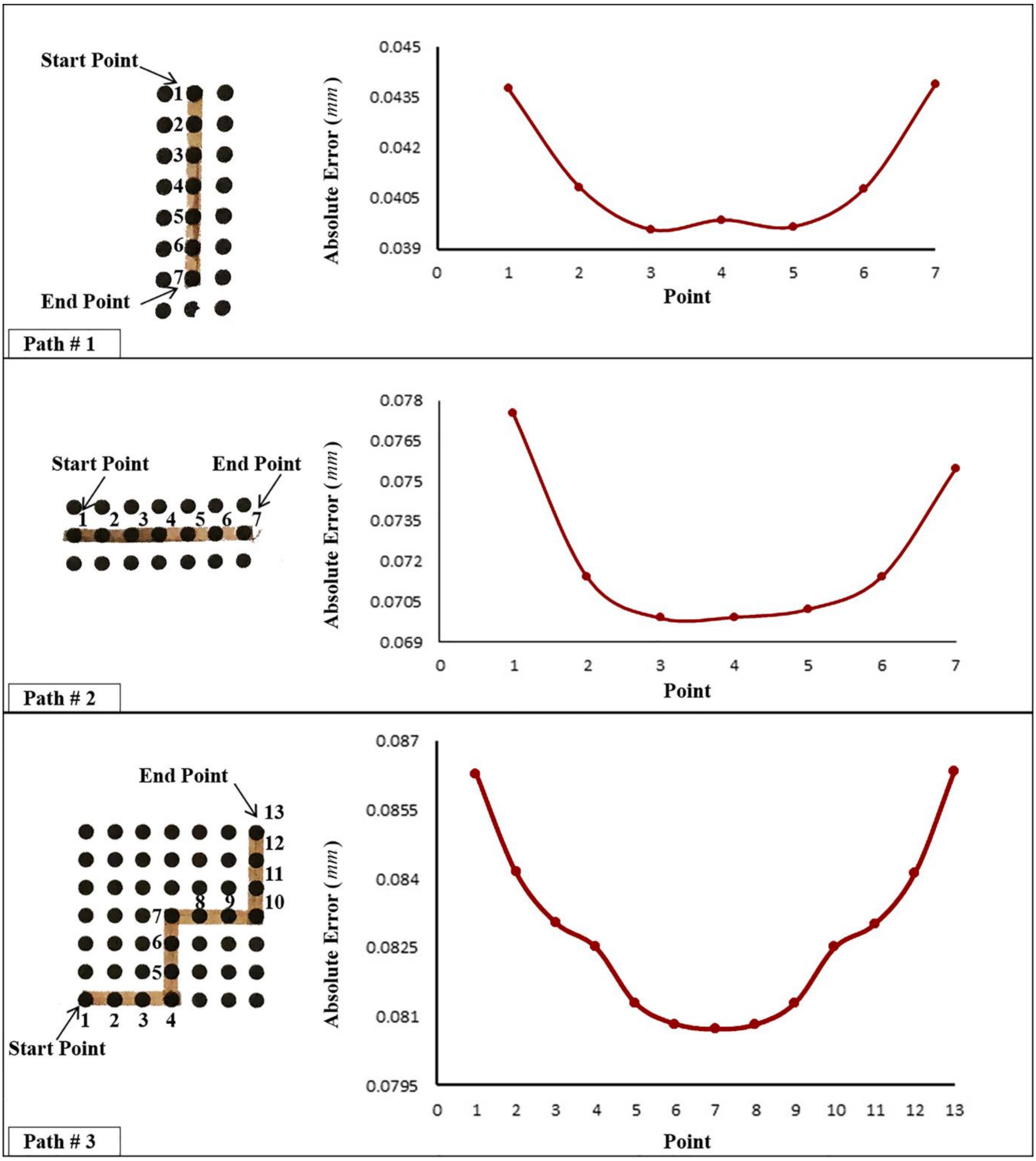


Fig. 7: Simple path

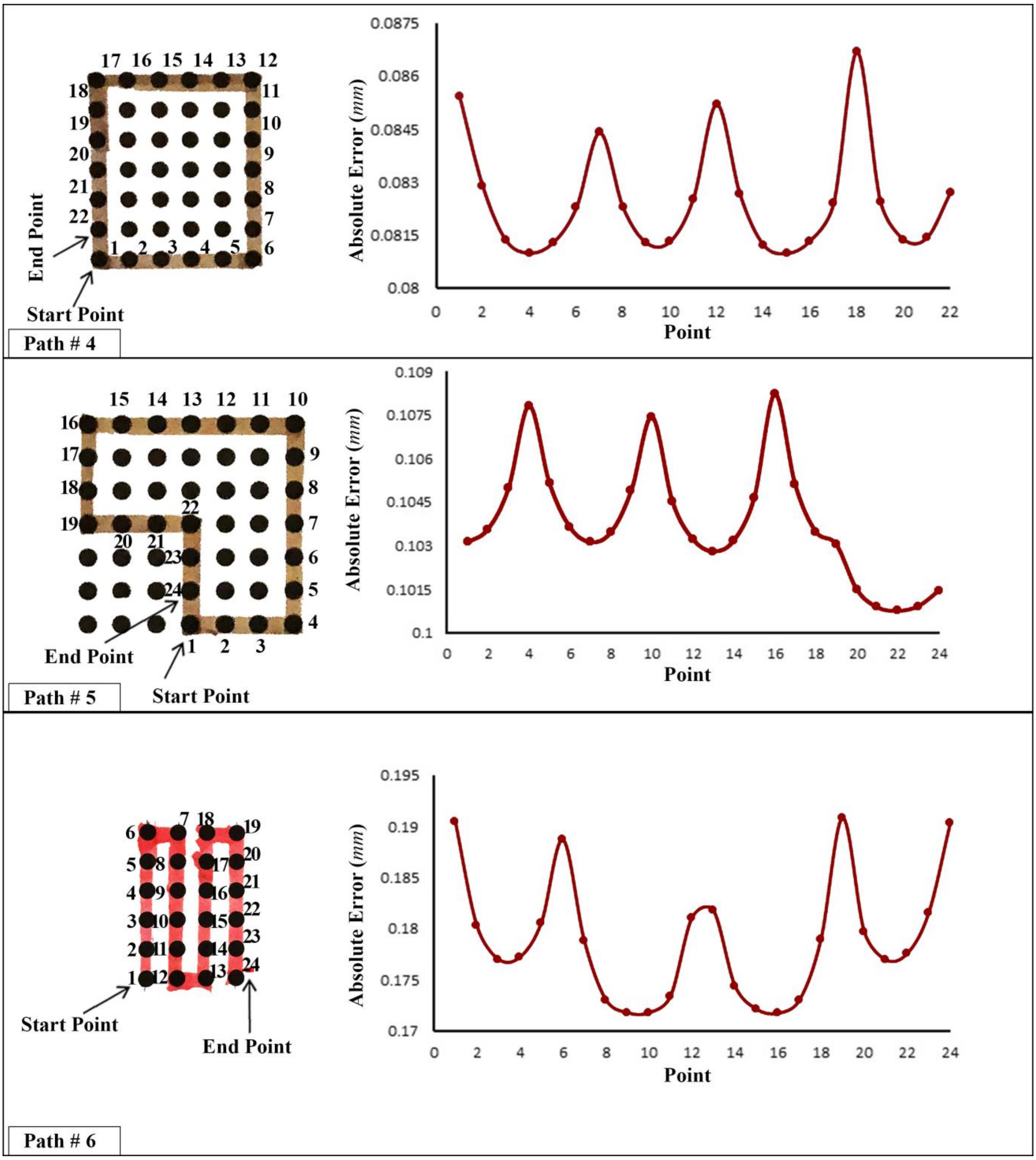


Fig. 8: Complex paths

Later, shape #6 only is chosen to be tested on the conventional technique for comparison with the proposed technique programming time. The path is virtually selected in Virtual Experimentation platform (V-rep) within 10 min. Next, 3D CAD model of the robot arm is created in solidWorks within 2 h and loaded to the virtual environment. In this stage, the robot arm follows the planned path virtually as shown in Fig. 9.

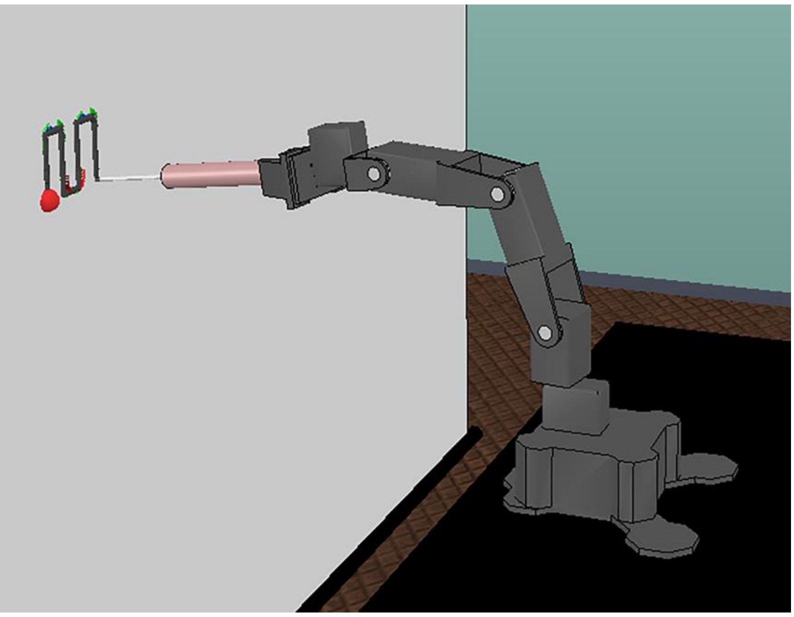


Fig. 9: Virtual environment

Moreover, the robot arm kinematics was obtained by numerical solution and the control point's data file is programmed by the user in MATLAB software within 2 h. Finally, the program is executed in real world within 39 sec after analysing the virtual complex path and results are satisfactory.

Due to the programming time for this shape in the conventional technique, it took 4 h and 10 min. Note that, this period of time depends on the programmer experience. While in proposed technique it took 23 sec. Despite that, the execution time is the same for the two techniques.

Discussion

The difficulty in conventional technique is specifying where the robot needs to move, and the paths, speeds, and motion types used to get there [22]. Furthermore, robot work cell accuracy becomes an important issues and solutions such as calibration, error identification, and compensation [23]. Hence, a CAD file is used for kinematic behaviour of the robot and collision detection in the work environment. Moreover, the accuracy of virtual robotic arm probably not always 100% in representing the real world.

Due to the complexity of programming, it took long time to follow the sequences required if desired task changed. Besides that, programming error could results a bad trajectory. Consequently, the program code or the quality of the simulator still needs to be modified. Flow chart describing sequences between the two techniques is illustrated in Fig. 10.

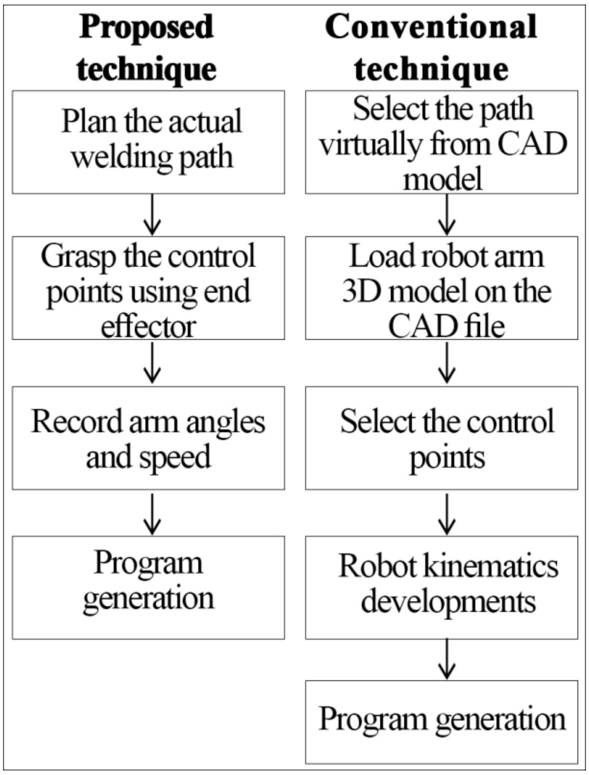

Proposed technique
Plan the actual welding path
Grasp the control points using end effector
Record arm angles and speed
Program generation
Conventional technique
Select the path virtually from CAD model
Load robot arm 3D model on the CAD file
Select the control points
Robot kinematics developments
Program generation

Fig. 10: Comparison between the two techniques

Hence, table 1 shows the programming ease and requirements differences in the two techniques.

Table 2: Programming techniques differences

	Proposed technique	Conventional technique
Time of programming	Short time	Long time
Expert programmer	Not required	Required
CAD simulation	Not required	Required
Kinematics solutions	Not required	Required

Table 3 shows the number of points, total length in mm, average of absolute error, standard deviation of the absolute error, time of programming and execution in sec for every path in the proposed technique.

Table 3: Proposed technique results

	Path #1	Path #2	Path #3	Path #4	Path #5	Path #6
Number of points	7	7	13	22	24	24
Total path length (mm)	35	35	60	105	115	115
Average of absolute error (mm)	0.041	0.063	0.083	0.082	0.103	0.178
Standard deviation of absolute error (mm)	0.001	0.009	0.002	0.002	0.002	0.006
Time of programming (sec)	6	6	11	17	20	23
Time of execution (sec)	5	5	12	18	21	24

As it can be observed in the table, standard deviation of absolute error increases as horizontal movements increase.

Conclusions

This paper introduces a supervised learning technique for programming a 4 DOF welding arm robot with automatic feeding electrode. A complete setup is established to assess the proposed technique experimentally. A vision system supported with in home-developed image processing program is used to measure the robot path accuracy. The results show that the proposed technique includes fewer steps and consumes less time in programming than the conventional ones. It was tested with a range of shapes. One of the major advantages of this technique is its ability to follow the real path including pre-processes errors. The easiness of the introduced programming technique will attract more users to the field of robotics.

References

1. Villani V, Secchi C, Leali F, Pini F. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Me. Sci. I. Mach. 2018; 55: p. 248-266.
2. Ragaglia M, Zanchettina A, Roccoa P. Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. Me. Sci. I. Mach. 2018; 55: p. 267-281.
3. West D. What happens if robots take the jobs ? The impact of emerging technologies on employment and public policy. CTI. 2015 October.
4. Hutchinson S, Hager G, Corke P. A Tutorial on Visual Servo Control. IEEE Trans. Robot. Autom. 1996 October; 12(5).
5. Mahony , Hamil T. Image-Based Visual Servo Control of Aerial Robotic Systems Using Linear Image Features. IEEE T-RO. 2005 April; 21(2).
6. Kuttan KA. Robotics. 1st ed. New Delhi: I.K International Publishing House Pvt. Ltd.; 2007.
7. Kim TW, Lee KY, Kim J. Wireless Teaching Pendant for Mobile Welding Robot in Shipyard. IFAC Proc. Vol. 2008 July; 41(2): p. 4304-4309.
8. Yong Y, Bonney M. Handbook of Industrial robotics "Off-line Programming". 2nd ed. Shimon , Y , Nof , editors. New York: John Wiley & Sons; 1999.
9. Raza K, Aized Khan T, Abbas N. Kinematic analysis and geometrical improvement of an industrial robotic arm. JKSUES. 2018 July; 30(3): p. 218-223.
10. Mohamed A, Sunar M. Kinematics Modeling of a 4-DOF Robotic Arm. IEEE ICCAR. 2015;: p. 87-91.
11. Somasundar A, Yedukondalu G. Robotic path planning and simulation by jacobian inverse for indstrial applications. Procedia Comput. Sci. 2018; 133(338–347).
12. Manjaree S, Nakra B, Agarwal V. Comparative analysis for kinematics of 5-DOF industrial robotics manipulator. Acta Mech. Autom. 2015 December; 9(4).
13. Deshpande V, George PM. Kinematic modelling and analysis of 5 DOF robotic arm. IJRRD. 2014 April; 4(2).
14. Bascetta L, Ferretti G, Magnani G. Walk-through programming for robotic manipulators based on admittance control. Robotica. 2013 October; 31(7).
15. Jr M, Wei. An Industrial Application of Control of Dynamic Behavior of Robots - A Walk Through Programmed Welding Robot. IEEE ICRA. 2000 April.
16. Qi L, Zhang D, Zhang J, Li J. A Lead-Through Robot Programming Approach Using A 6-DOF Wire-based Motion Tracking Device. IEEE ROBIO. 2009 December;: p. 1773-1777.
17. Lachat E, Macher H, Mittet M, Landes T, Grussenmeyer P. First experiences with kinect v2 sensor for close range 3D modeling. ISPRS. 2015 February; XL-5/W4.
18. Paul S, Basu S, Nasipuri. Microsoft Kinect in Gesture Recognition: A Short Review. IJCTA. 2015; 8(5): p. 2071-2076.
19. Rahman M. Beginning Microsoft Kinect for Windows SDK 2.0: Motion and Depth Sensing for Natural User Interfaces. 1st ed. Montreal: Apress; 2017.
20. Jazar R. Theory of Applied Robotics. 2nd ed. United States of America: MASpringerUS; 2010.

21. Corke P. Robotics, Vision & Control: Fundamental Algorithms in MATLAB. 2nd ed.: Springer; 2011.
22. Rossano G, Martinez C, Hedelind M, Murphy S, Fuhlbrigge T. Easy Robot Path Programming Concepts: An Industrial Perspective on Path Creation. IEEE ISR. 2013 October;: p. 1-6.
23. Nee A. Handbook of Manufacturing Engineering and Technology. 1st ed. London: Springer-Verlag London; 2015.