IJCSES International Journal of Computer Sciences and Engineering Systems, Vol. 4, No. 3, July 2010
CSES International © 2009 ISSN 0973-4406

Logic Algorithm for Contour Following Task: An Evaluation Using Adept SCARA Robot

Anton Satria Prabuwono[1], Samsi Md. Said[2] and Burhanuddin Mohd. Aboobaider[3]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia43600 UKM Bangi Selangor D. E., Malaysia, *E-mail: antonsatria@ftsm.ukm.my*

[2]Faculty of Electrical and Automation Engineering Technology, TATI University CollegeTeluk Kalong 24000 Kemaman, Terengganu, Malaysia, *E-mail: samsi@tatiuc.edu.my*

[3]Faculty of Information and Communication Technology, Universiti Teknikal Malaysia MelakaAyer Keroh 75450 Melaka, Malaysia*E-mail: burhanuddin@utem.edu.my*

Abstract: *This paper presents the performance evaluation of logic algorithms for contour following task in order to automate the manual programming process. The main objective of this work is to evaluate and compare the performance of single logic and dual sensor logic algorithms for autonomous contour following in industrial robot. Those algorithms have been implemented using Adept SCARA robot. The algorithms were tested on a semicircle object of 40 millimeter radius. The result shows mean of error and standard deviation value of single logic algorithm is a bit lower than dual sensor logic algorithm.*

Keywords: *Single Logic, Dual Sensor Logic, Contour Following, Industrial Robot.*

1. INTRODUCTION

One of basic task in industrial robot manipulation is contour following process. In this process, the robot is holding a tool to follow the contour of an object whose shape and pose are often unknown [1]. These applications include part polishing, inspection, sealing, painting, cleaning, modeling, etc. During the process, the tool is constrained on the surface to maintain contact force while moving along some tangential direction. Innovative developments have been done in automating and enhancing teaching process of the robot applications as mentioned in the references [2], [3], [4], and [5].

In order to use robotics for such application, two sequences of step need to be considered i.e. the programming phase and the playback phase. In the programming phase, teaching a group of points is required while for playback phase, the robot Tool Centre Point (TCP) will follow the taught points recorded previously. This programming phase particularly for contour following application such as in arc welding, sealing and painting application is quite tedious and time consuming [6]. As an example, in order to follow an arc, the robot programmer needs to manually use teaching box or teaching pendant to jog the robot TCP to three points that enclosed an arc [7]. Furthermore, different kind of parts variations for one production process is required in the recent Flexible Manufacturing Process (FMS). A high number of robot programming is required to cater parts variations and uncertainties per production process. This requirement is difference if compared to the old days of batch or mass production concept.

This research compares and evaluates the performance of single logic and dual sensor logic algorithm for contour following task in real industrial robot. The algorithms were developed and tested using Adept SCARA robot.

2. LOGIC ALGORITHM

This section describes single logic and dual sensor logic algorithm that applied in the experiment.

2.1. Single Logic

This method depends on sensor logic condition whether it is ON or OFF. If sensor is ON the condition is to command the robot TCP to climb upward and if the sensor condition is OFF the robot TCP will move downward. The climbing and descending condition are at constant value of dX, $+dZ$ and $-dZ$. A horizontal sampling distance dX at time T must be defined at the beginning of the process. The sampling distance dX, climbing distance dZ

Manuscript received February 2, 2009
Manuscript revised September 20, 2009

and descending distance *-dZ* is the critical parameter that defines the zig zag almost like a staircase trajectory that approximating contour positive or negative gradient. The longer the horizontal distance *dX* and climbing/ descending distance *dZ*, the coarser the curve modeled.

The constant parameter value must be adjusted by user to accommodate the gradient slope that is to be tracked. The ratio of sampling distance *dZ* over *dX* will define the maximum gradient slope. User must estimate the steepest contour slope and adjust the parameter accordingly. Figure1 shows climbing and descending position of single logic method.

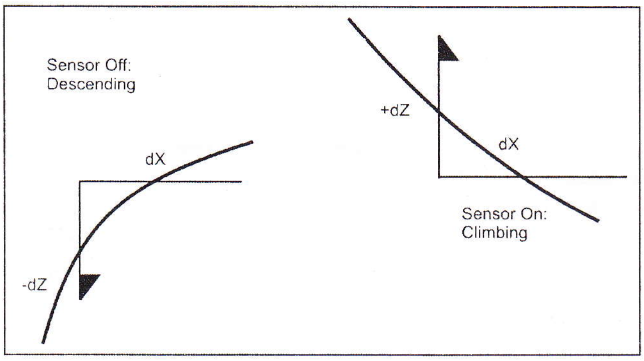


Figure 1: Climbing and Descending Decision

Figure 2 illustrates incremental logic over contour using this method.

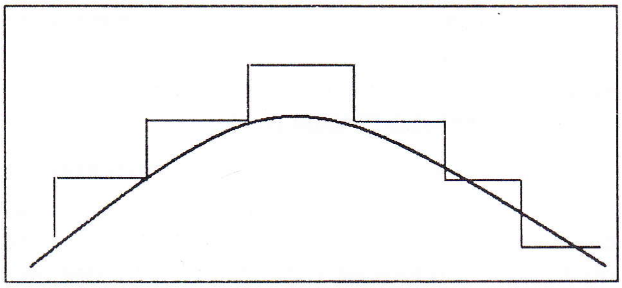
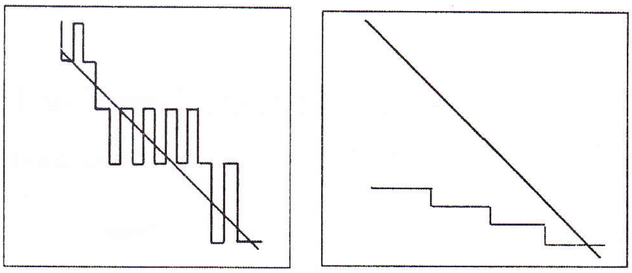


Figure 2: Incremental Logic Over Contour

Figure 3 (a) and (b) show the effect of employing high ratio of dZ/dX on a curve of lower gradient value. High value of dZ/dX ratio can accommodate low slope contour but the low value of dZ/dX cannot track high gradient contour.

The upward logic causes incremental climbing motion along contour while downward logic causes incremental descending motion along contour. The position *p[i]* contour information is stored in the position database to be used repeatedly in the playback mode. In this way the whole contour is being approximated and the positions stored. The positions can be used for $P_{i+1} = P_i D(i + 1)$.



(a) Too high (b) Too low

Figure 3: Effect of dZ/dX Value Along Gradient Contour

The drive function that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & n_A \bullet dX \\ 0 & 1 & 0 & s_A \bullet dX \\ -S(\theta) & 0 & C(\theta) & a_A \bullet dZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Total trajectory point generated at point *N* and local gradient value is:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \quad (2)$$

$$\delta_{XZ} slope = \frac{a_A \bullet dZ}{n_A \bullet dX} \quad (3)$$

The detail flowchart of single logic algorithm is shown in figure 4.

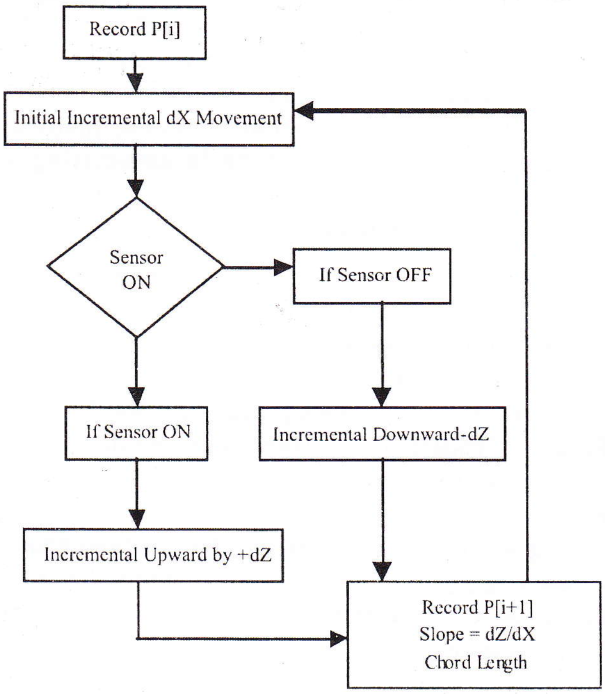


Figure 4: Single Logic Algorithm

2.2. Dual Sensor Logic

In this method, smart tool was designed to hold two digital proximity sensors of different sensing distance [8] as shown in figure 5 and figure 6.

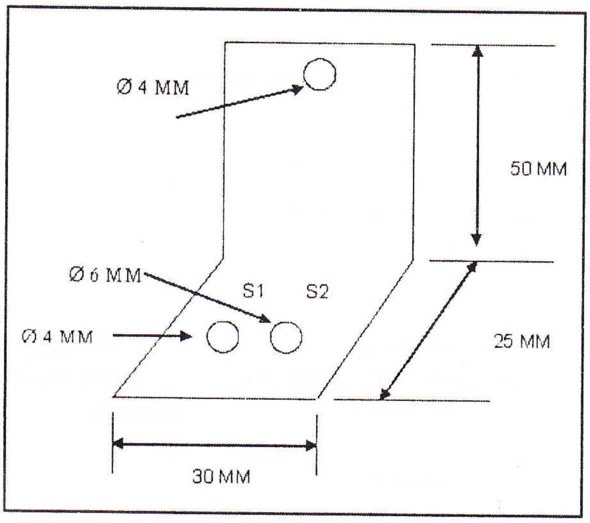


Figure 5: Dimension of Smart Tool Holder

This sensing distance difference (SDD) is measured by finding the downward vertical variable difference and is adjusted to be as low as 1 mm to as high as 5 mm. Sensing range (SR) is adjusted of about 15 mm for safety reason to avoid collision to contour. In order to accommodate mechanical design of sensor, the sensor spacing (DX) is fixed to 10 mm. The contour slope sensitivity index is a ratio of SDD over DX. The lesser the index value the more sensitive the contour slope to be sensed.

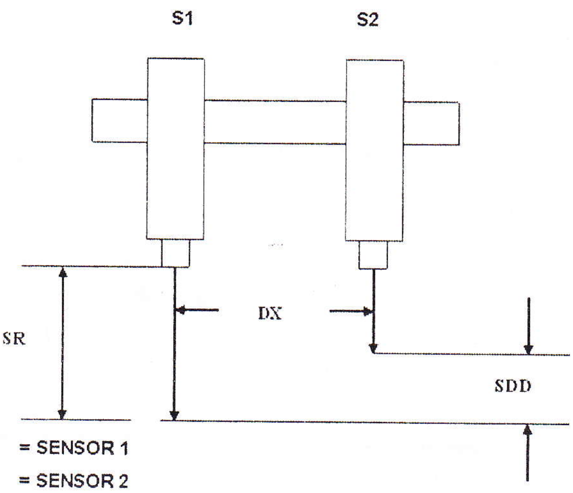


Figure 6: Tool Holder Attached to Adept SCARA Robot TCP

The sensor number 1 and 2 will detect the current contour gradient whether it is positive climbing, negative descending or just horizontal flat. The contour gradient based on the sensors 1 and 2 logic conditions are described as follow:

(a) If both sensors were switched on, then TCP move up by SDD mm and move horizontally by SDD mm.

(b) If both sensors were switched off, then TCP move down by SDD mm and move horizontally by SDD mm.

(c) If sensor one was on and sensor two was off, move horizontally by SDD mm or until condition a, or condition b was reached (whichever comes earlier).

Movement logic of sensors is shown in table 1.

Table 1
Movement Logic of Sensors

S1	*S2*	*Operation*
0	0	Descend
1	0	Horizontal flat
1	1	Ascend

Sensors movement is shown in figure 7.

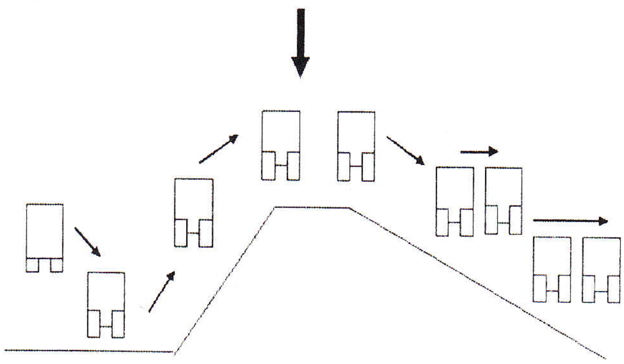


Figure 7: Movement of Sensors

The programming takes advantage of the current information of sensor 1 and sensor 2. If both sensors' readings are on that means at current location the contour slope is positive climbing. If only sensor 1 reading is on that means TCP on a flat horizontal shape. If both sensors' readings are off then the TCP is on descending mode. The next motion of $P(i+1)$ is being commanded using current data and be tested using same rules. The program starts with user definition of contour length as configured previously.

The algorithm will find the slope in X-Y plane and total distance diagonally and x and y vector distance. If the starting point is greater than sensor 1 sensing range (SR), both sensors input to the robot controller is in zero volt (OFF) and condition descending is applied. Robot will incrementally move downward until the contour will within sensor 1 sensing range (SR). At this point the sensor 1 input value is 24 volt (ON) while sensor 2 value is zero volt (OFF). This indicates flat condition and the robot will move horizontally until it senses the climbing positive slope contour where both sensors 1 and 2 reading's are on. At this stage the condition climbing is applied and robot will climb. The transition of variable slope changing from positive slope to flat and negative slope will cause the program to implement noise motion. In climbing mode, any variable slope transition vertex will cause flat noise motion while in descending mode multiple flat noise motion staircase deviation happen due to both sensors are off in descending logic. Noise study and optimization were carried out in this research especially in descending mode. These phenomena will be improved in future research. Figure 8 shows flowchart of dual sensor logic algorithm.

3. DRIVE TRANSFORM MODELING

Smart sensor feedback and programming algorithm will guide the TCP to approximate the curve with a straight line segments that knot from points to points in three dimensional Cartesian X-Y-Z plane. The measured knot points and segment slope at any points will be stored in the database and will be used repeatedly in robot part program playback mode. The objectives of automating tedious and time consuming contour tracking programming process will be achieved. The algorithm will further explain the incremental position of δ_X, δ_Y and δ_Z of general incremental drive transform described in Eq. 6. As an initial, in order to simplify the mathematical formulation, four degree of freedom Adept SCARA industrial robot is used. In future research, a six degree of freedom robot can be used to test the robustness and applicability of the algorithms. Utilizing drive transform equation for four degree of freedom SCARA robot will simplify a lot of things [9], [10]. For example, only one yaw orientation angle exist. Then, the chord segment relative path transformation drive transform is being decomposed only into one rotation matrix to orientate tool about Z axis and one straight line translation matrix also along tool axis. In order to achieve the motion between two consecutive Cartesian knot points, the derivation of segment drive transform is very useful since motion from i to $i+1$ is related to drive transform as:

$$T_4(i+1) = C_{Workobject}\, P_i D(i+1)\, ({}^{tool}T_{i+1})^{-1} \quad (4)$$

$T_4(i+1)$ is the transformation stored to the database and contain both tool position and orientation at any points which also becomes input to the inverse kinematics routine in order to get local coordinate of individual robot joint angles (another joint level cubic polynomial trajectory planning or differential Jacobian method which is not discussed here). After some mathematical operation, the position of consecutive knot points at beginning from i to end of segment $i+1$ is a function of drive transform as:

$$P_{i+1} = P_i D(i+1) \quad (5)$$

The general transformation matrices drive transform that summarizes all these can be represented as:

$$D(i+1) = \begin{bmatrix} C(\theta) & 0 & S(\theta) & \delta_X \\ 0 & 1 & 0 & \delta_Y \\ -S(\theta) & 0 & C(\theta) & \delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

The yaw orientation angle θ is actually an input from gross motion task planning that was discussed previously. The detail of incremental position in Cartesian X-Y-Z three dimensional plane δ_X, δ_Y and δ_Z will be explained in several individual alternate algorithm developments in the next section just to automate this differential

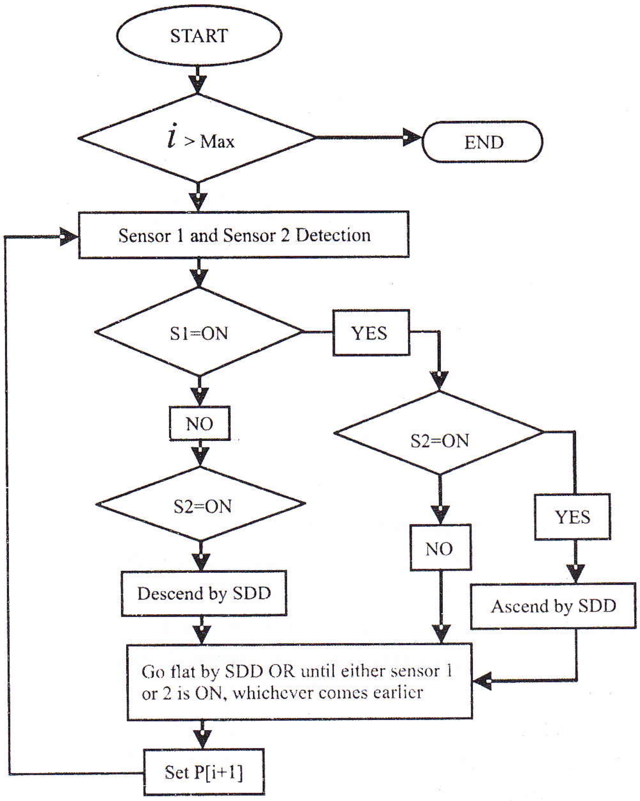


Figure 8: Dual Sensor Logic Algorithm

relative motion. Incremental drive transform that will describe the final position at any point N is generated as:

$$P_N = \prod_{i=1}^{N-1} P_i D(i+1) \quad (7)$$

The related transformation at any point N which became input to inverse kinematics routine for joints space trajectory planning as follow:

$$T_4(N) = \prod_{i=1}^{N} C_{Workobject} P_i D(i+1)({}^{tool}T_{i+1})^{-1} \quad (8)$$

4. RESULT

The algorithms were written in V+ programming. The contour traced by robot TCP on the object which has radius value of 40 millimeter (mm) along the X axis are plotted according to the algorithm used. Semicircle shape was chosen for an object of experiment. It provides an ideal test bed because it contains all ranges of slope gradient that are available in real world. It exhibits infinity value at the very beginning point and progressing down with a finite very high positive slope. The slope decreasing into zero value in the middle of the contour and finally reaches very high negative slope at the other end along the X axis. At the very end of the semicircle contour the infinity slope reappear again. These phenomena cause high reading of Cartesian vector Z for any minute vector X displacement value. These infinity region problems will be avoided by introducing a safety margin ranging from 0.1 mm to 2.5 mm at the both ends of the semicircle geometry. It is anticipated that the tracking error value will be quite high in certain slope region of contour gradient.

The actual contour traced and the tracking error along contour, matching the semicircle geometry of radius 40 mm is plotted in figures 9 and 10. The enlargement of mean of the tracking error with the value of 9.9520 mm and the standard deviation of the tracking error with the value of 3.8030 mm respectively. The safety margin of 0.1 mm to 1 mm is allowed at the beginning and near to the end of semicircle object in order to avoid measuring the very high slope at those regions. The single logic measuring advance parameter of 1 mm is chosen for this contour tracking experiment. The total sample of 79 points was collected over 80 mm horizontal measuring distance.

In dual sensor logic, the actual contour traced and the tracking error along contour, matching the semicircle geometry of radius 40 mm is plotted in figure 11 and figure 12.

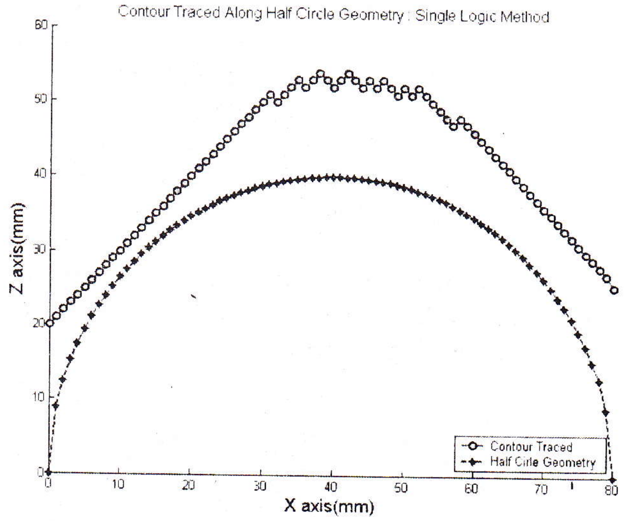


Figure 9: Contour Traced using Single Logic Method

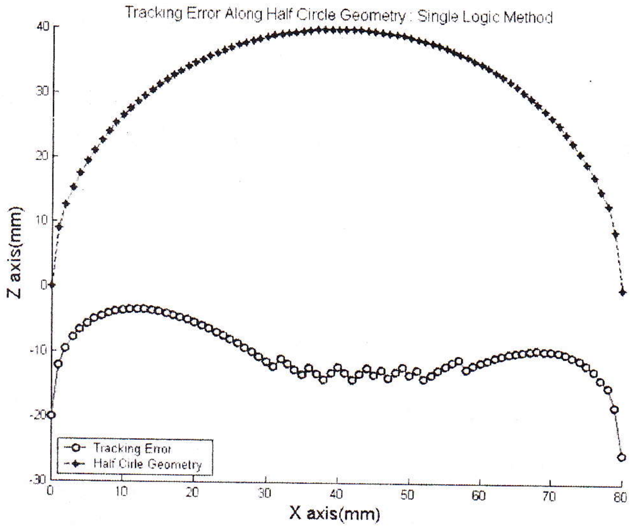


Figure 10: Contour Traced using Single Logic Method

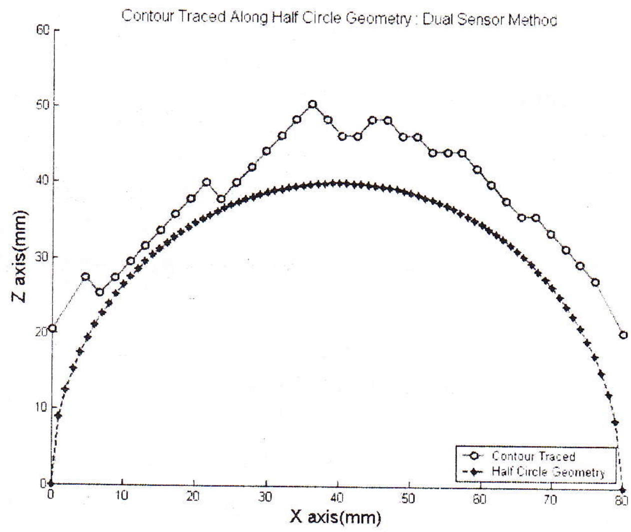


Figure 11: Contour Traced using Dual Sensor Logic Method

The enlargement of mean of tracking error with the value of 13.7015 mm and the standard deviation of tracking error with the value of 4.1247 mm respectively. The safety margin of 0.1 mm to 1 mm is allowed at the beginning and near to the end of semicircle object in order to avoid measuring the very high slope at those regions. The sensor difference between sensor 1 and sensor 2 (SDD) value is about 2.21 mm giving a bit coarse measurement points spaced apart at every 2.21 mm. The total sample of 35 points was collected over 80 mm horizontal measuring distance.

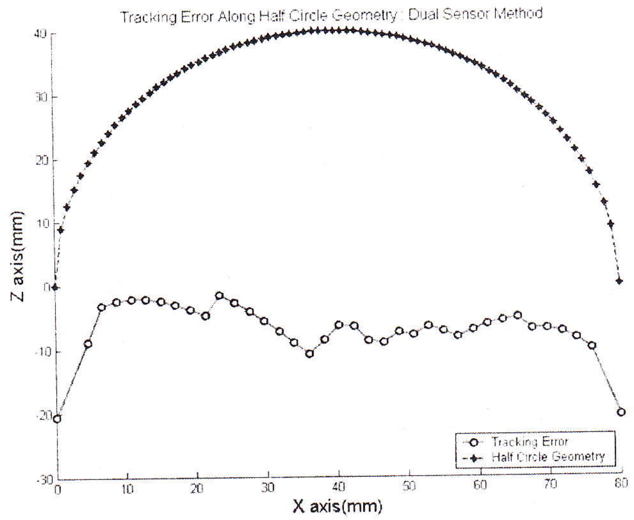


Figure 12: Tracking Error using Dual Sensor Logic Method

5. EVALUATION

Efficiency of the method is measured with regard to the mean of error, standard deviation value and path traveling distance. The better method is defined to be lesser mean of error and standard deviation value as well as shorter path traveling distance. Based on the graphs shown in experimental result, it appears that single logic and dual sensor logic method have similarity in term of path traveling distance. Although single logic method has better pattern of contour traced, but it does not influence the overall performance. In fact, the mean of error and standard deviation value of single logic are a bit lesser than dual sensor logic method. It means single logic provides better performance compared to dual sensor logic algorithm. The results are presented in table 2.

Table 2
Performance Result of Single Logic and Dual Sensor Logic

Criteria	*Single Logic*	*Dual Sensor Logic*
Mean of Error	9.9520	13.7015
Standard Deviation	3.8030	4.1247

6. CONCLUSION

This research provides the performance comparison and evaluation of two types of logic algorithms for autonomous contour following. In order to test the methods, the programming to implement those algorithms was also developed. Experiments were done by employing Adept SCARA robot in real environment. The algorithms were tested on a semicircle object of 40 mm radius. From the experiment results, it can be concluded that single logic algorithm has better performance compared to dual sensor logic algorithm. The results show that its possibility could be used those algorithms in the real manufacturing application.

ACKNOWLEDGMENT

The authors would like to thank Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia for providing facilities and financial support.

REFERENCES

[1] L. Mi, and Y. B. Jia, "High Precision Contour Tracking with Joystick Sensor", in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2004, 804-809.

[2] Z. Rasol, D. A. Sanders, and G. E. Tewkesbury, "New Prototype Knowledge Based System to Automate a Robotics Spot Welding Process", *Elektrika*, **4**, 2001, 28-32.

[3] B. Adolfo C. A. A. Sadek, and A. D. Leszek, "Predictive Sensor Guided Robotic Manipulators in Automated Welding Cells", *Journal of Materials Processing Technology*, **9**, 2001, 13-19.

[4] Y. Yuehong, H. Hui, and X. Yanchun, "Active Tracking of Unknown Surface Using Force Sensing and Control Technique for Robot", *Sensors and Actuators: A Physical*, **112**, 2004, 313-319.

[5] B. Gopalakrishnan, S. Tirunellayi, and R. Todkar, "Design and Development of an Autonomous Mobile Smart Vehicle: a Mechatronics Application", *Mechatronics*, **14**, 2004, 491-514.

[6] A. S. Prabuwono, and M. S. Samsi, "Development of Adapting Gradient Method for Contour Tracking in Industrial Robot Application", in *10th IASTED Int. Conference on Intelligent Systems and Control*, 2007, 592-068.

[7] A. S. Prabuwono, M. A. Burhanuddin, and M. S. Samsi, "Autonomous Contour Tracking Using Staircase Method for Industrial Robot", in *10th IEEE Int. Conference on Control, Automation, Robotics and Vision*, 2008, 2272-2276.

[8] A. S. Prabuwono, M. S. Samsi, R. Sulaiman, and E. Sundararajan, "Contour Following Task with Dual Sensor Logic Algorithm for Adept Selective Compliant Assembly Robot Arm Robot", *Journal of Computer Science*, **5**(8), 2009, 557-563.

[9] R. P. Paul, "Manipulator Cartesian Path Control", *IEEE Transactions on System, Man, Cybernetics*, **9**, 1979, 702-711.

[10] R. P. Paul, "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm", *IEEE. Memo AIM-177*, Stanford Artificial Intelligence Laboratory, Palo Alto California, 1972.