International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Algorithm for Line Follower Robots to Follow Critical Paths with Minimum Number of Sensors

Nakib Hayat Chowdhury[a]*, Deloara Khushi[b], Md. Mamunur Rashid[c]

[a,b,c]*Department of Computer Science and Engineering, BAUST, Saidpur Cantonment, Nilphamari, Bangladesh*

[a]*Email: nakib2025@gmail.com*

[b]*Email: khushi.hstu@gmail.com*

[c]*Email: mamunst2003@yahoo.com*

Abstract

This paper introduces an approach which help a line follower robot to achieve the ability to autonomously follow a path that has straight lines, curve, 90 degree bends, T-junctions and + junctions using minimum sensors. Generally a line follower robot uses matrix of sensors to achieve this ability. This paper discuss about the algorithm through which a line follower can do the same thing using an array of sensor with minimum number of sensors. Because, if we able to decrease sensor number, we can decrease the complexity of the robot. Additionally, it also reduce the cost. But, here main challenge is ensuring efficiency. As we know, for a line follower robot, it is difficult to detect and follow T-junctions, 90 degree bends and grid junctions. To overcome this challenge, people add extra sensors to detect 90 degree bends, T-junctions and + junctions and design algorithm to follow them. If we reduce sensor, those robots will not able to detect these critical junctions and so, lose its efficiency. This paper discusses about line sensors configuration, their positions and also develop an algorithm so that a robot can follow a path with curve, 90 degree bends, T-junctions and + junctions using only four sensors.

Keywords: Autonomous line follower; IR Sensor; Minimum Sensors; 90 degree bends; T-junctions; + junctions.

1. Introduction

Line follower robots are autonomous robot having the ability to detect and follow a line using on board hardwired control circuit [1,2]. Now a day, to reduce human effort and ensure efficient automatic transport system line followers are becoming popular. Especially in industrial areas, these are using in large number [3].

----

* Corresponding author.

To be an ideal industrial element, these robot need to provide high efficiency at minimum cost. That is, robot should have the ability to follow complex lines and on other hand, it have to be simple, easy to operate and inexpensive. Beside all other things, if we reduce the number of sensor this will reduce the complexity and cost. But, if we reduce the sensor, it become very difficult for a line follower to detect critical angels such as 90 degree bends, T-junctions and + junctions. So, to achieve high efficiency with less sensors, we have to find an appropriate algorithm, as well as we have to set sensors at appropriate positions [4]. Here, we discuss about some experimental result on sensor positioning and controlling strategy of our robot (figure 1) to develop an algorithm through which a line follower robot can detect T-junction, + junction and 90 degree turn accurately with minimum sensor. We have tested our robot on a sample path which was built based upon requirement of a line following robot competition, RoboTour, 2015, organized by IEEE RUET Student Branch [5] as shown in figure 2.

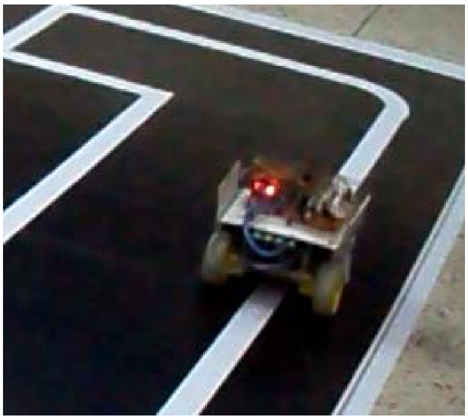

Figure 1: Line Follower Robot

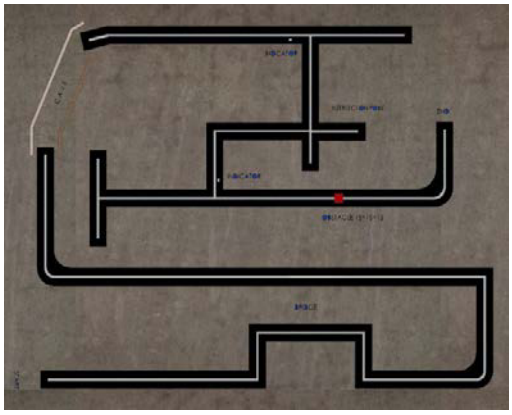

Figure 2: RoboTour, 2015 competition track

2. Materials and Methods

2.1 Robot Hardware

We have used Arduino Uno as main circuit board of our robot. To detect line, it used an array of four sensor units. Each unit contain an IR transmitter and IR receiver. As shown in figure 3, IR sensor reflects light when it is on white surface but it does not reflected on black surface [6]. Using this property of IR sensor robot can easily detect a while line of black surface or vice versa. Our robot also used comparator to tune sensors' value.

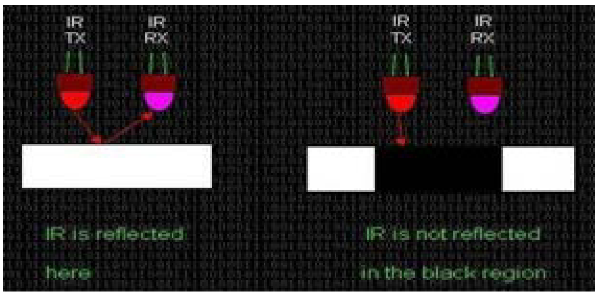

Figure 3: IR sensor working principle

This robot used two 12 V DC motor for motion, a motor controller to control these two motors' speed. In this paper, we are not discussing about the robot hardware details such as underlying circuit, power supply unit, sensor array circuit, motor controller etc. Rather, this paper discuss how a robot can follow a complex line with minimum sensors.

2.2 Robot's path analysis

As mentioned earlier robot's path was built based on a track designed for line following robot contest [5]. We also followed all its rules and regulations. This ensure that the sample path was complex enough to test efficiency of a line following robot. Referring to Figure 2, robot have to follow a white line on black surface. As shown in Figure 4, the white line is 4 cm wide and it is at the middle of 30 cm black surface.

Figure 4: Sample Road Track Dimension

As shown in figure 2, this path has three 90 degree left bends, two 90 degree right bends, three T-junctions and one + junction. This sample pitch also have 45 cm cave and an obstacle to avoid. But this paper only focus on line following algorithm with minimum sensors, so, we leave those.

2.3 Sensors' arrangement

For successful navigation on critical paths, number of sensor and their position play vital role. In general, a matrix of sensors with eight sensor or at least six sensor make a robot able to detect critical junctions. In our robot we set different experiments with sensors to reduce its number. To detect any line minimum two sensors are required [7]. Using these two sensor a robot can easily follow a straight line or simple wavy line. In our robot we used an array of only four sensor, where middle two sensors were set for follow simple line. We set this two sensor such that in ideal position the line remain just between these sensors. But using only two sensor it is not possible to detect a T-junction, + junction or any 90^0 angle. So, we add additional two sensor on both sides, which helped our robot to detect T-junction and 90^0 angle. Though, using only four sensor our robot still not able to directly detect a + junction. But, we developed a feedback based algorithm such that, whenever any + junction appears, our robot able to choose the correct path from three alternative options: *go straight, take a* 90^0 *left turn or take a* 90^0 *right turn*. Figure 5, 6 shows full sensor arrangement.

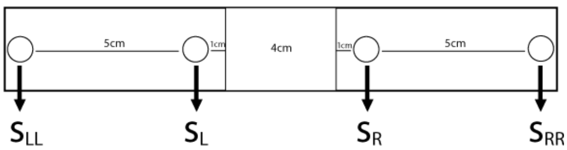

Figure 5: Sensor array

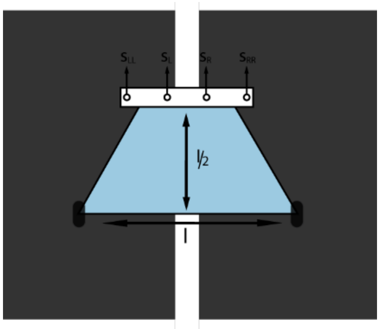

Figure 6: Robot's block diagram

Here middle two sensor S_R and S_L are used to detect simple line. Outer left sensor S_{LL} were set to detect 90^0 left turn and outer right sensor S_{RR} to detect 90^0 right turn. Distance between S_R and S_L sensors is 6 cm. So, in ideal position these two remained 1 cm far from main base line. Distance between sensors S_{RR} to S_R and S_{LL} to S_L were set 5 cm for both cases.

2.4 Line following algorithm

Following flow-chart (figure 7) describes the basic working principle of our robot.

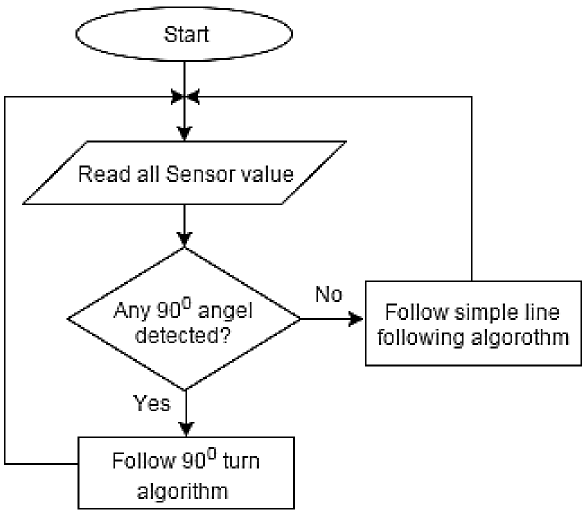

Figure 7: Basic working principle.

As shown on this flow chart, the navigational strategy is divided into two parts: *simple line detection* and *simple line following algorithm*, *90 degree angle detection and 90 degree turn algorithm*.

2.5 Simple line detection and simple line following algorithm

This robot detect a line as simple line and follow simple line following algorithm if and only if both of its outer sensors (S_{LL}, S_{RR}) are on black surface. Here, three situations can be happened, as shown in figure 8, 9 and 10.

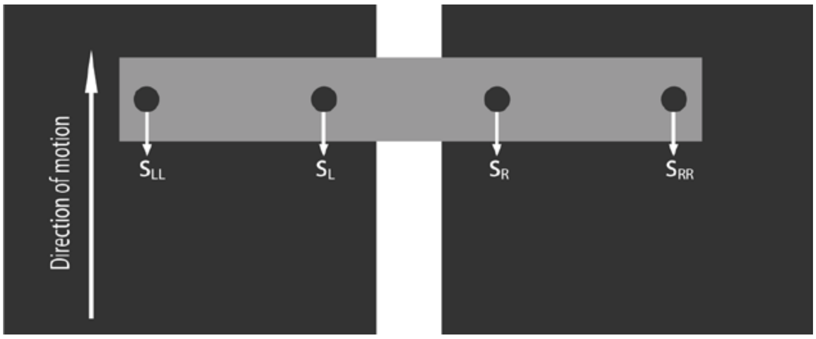

Figure 8: Ideal situation, move Straight

Consider figure 8, when all sensors are on black surface. This the ideal position for the robot. In this situation robot will move forward with full speed by rotating it's both motors at maximum rpm.

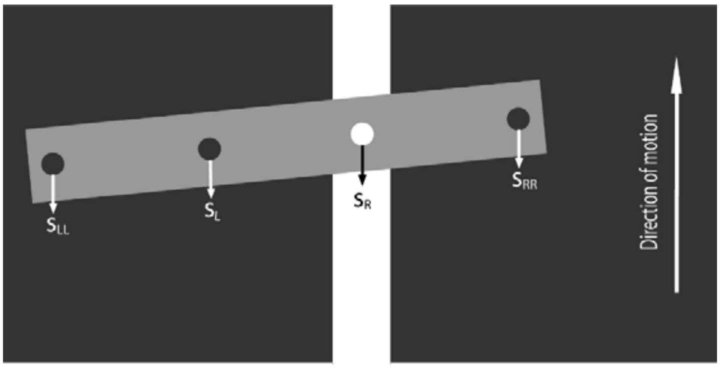

Figure 9: Normal right turn

In second situation (Figure 9) both outer two sensors (S_{LL}, S_{RR}) and middle left sensor (S_L) are on black surface, only the middle right sensor (S_R) is on white line. So, the robot need to move right until middle right sensor (S_R) backs to black surface (to move back to ideal position). To do so, this robot slow down the right motor speed to half and keep left motor to highest speed. Rather than stopping the right motor, we reduce its speed to gain better navigation.

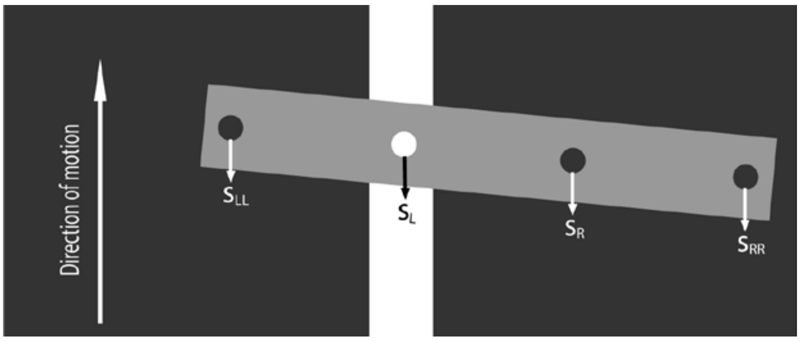

Figure 10: Normal left turn

In third situation (figure 10), middle left sensor (S_L) is on white line and other three sensors are on black surface.

So, the robot need to move left until middle left sensor (S_L) backs to black surface. So, we reduce left motor's speed to half and keep other's high. The following algorithm describe the full procedure.

Simple line following algorithm

1. Start
2. Read S_{LL}, S_L, S_R and S_{RR} sensor's value.
3. If S_{LL} and S_{RR} on black surface
4. If S_L and S_R on black surface
5. Move forward (rotate both motor on full speed)
6. Go to step 2
7. If S_R on white line
8. Move Right (reduce Right motor speed to half)
9. Go to step 2
10. If S_L on white line
11. Move Left (reduce Left motor speed to half)
12. Go to step 2
13. If S_{LL} and/or S_{RR} is/are on white surface
14. Not a simple line
15. Follow 90 degree turn algorithm

Figure 11: Simple line following algorithm

2.6 90 degree angle detection and 90 degree turn algorithm

If the robot detects any 90 degree angle in its path, then it will follow 90 degree turn algorithm. This can be happen in two ways- *90 degree left turn or 90 degree right turn*.

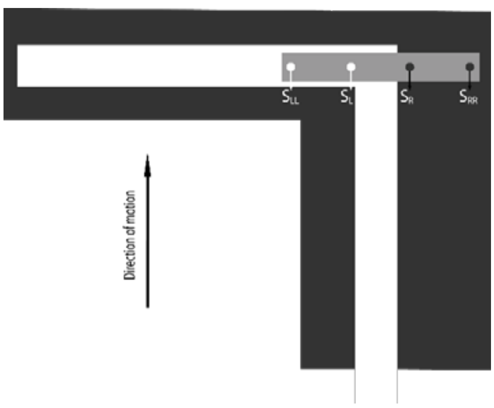

Figure 12: 90 degree turn - left

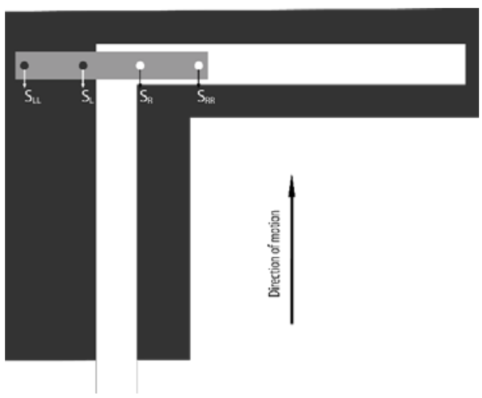

Figure 13: 90 degree turn - right

As shown in figure 12 and figure 13, our robot will detect a path segment as 90 degree angle if either both S_L and S_{LL} sensors are on white line or both S_R and S_{RR} sensors are on white line. If both S_L and S_{LL} are on white line, this robot will take a 90 degree left turn. To do so, we rotate left motor in reverse way with full speed and rotate right motor in forward way with full speed, until both S_{RR} and S_{LL} sensor are back on black surface. To take a 90 degree right turn, we do the same thing in reverse way. Following algorithm describe full procedure.

90 degree turn algorithm

1. Start
2. Read S_{LL}, S_L, S_R and S_{RR} sensor's value.
3. If S_{LL} and S_{RR} on black surface
4. Follow simple line following algorithm
5. If S_{LL} and S_L on white line
6. // 90^0 left turn
7. While S_{LL} and S_{RR} are not back on black surface
8. Rotate left motor in reverse direction and right motor in forward direction
9. Read S_{LL}, S_L, S_R and S_{RR} sensor's value.
10. If S_{RR} and S_R on white line
11. // 90^0 right turn
12. While S_{LL} and S_{RR} are not back on black surface
13. Rotate right motor in reverse direction and left motor in forward direction
14. Read S_{LL}, S_L, S_R and S_{RR} sensor's value.
15. Go to step 2

Figure 14: Simple line following algorithm

3. Results and Discussion

Our robot able to follow all parts of sample test path smoothly. Using above two algorithms, this robot easily able to detect and follow straight line, simple curve and 90^0 left or right bends. But whenever any T-junction or + junction appears on the path, our robot need additional instruction to choose correct paths from alternatives.

A T-junctions opens two alternatives paths to robot: *a 90^0 left turn* or *a 90^0 right turn*. As mention earlier at a + junction the robot has to choose correct path from three alternatives: *go straight, take a 90^0 left turn or take a 90^0 right turn*.

If we closely analysis these alternative options from both T-junction and + junction, we can see that our robot can follow any one of these using above two algorithms. It just need to choose correct option. To do so, we have used a counter variable with initial value 0.

As, shown in figure 15, when all sensors were on white line, then robot detected it as a T-junction. Again, for + junction, this robot will detect it as T-junction because it has only an array of four sensors.

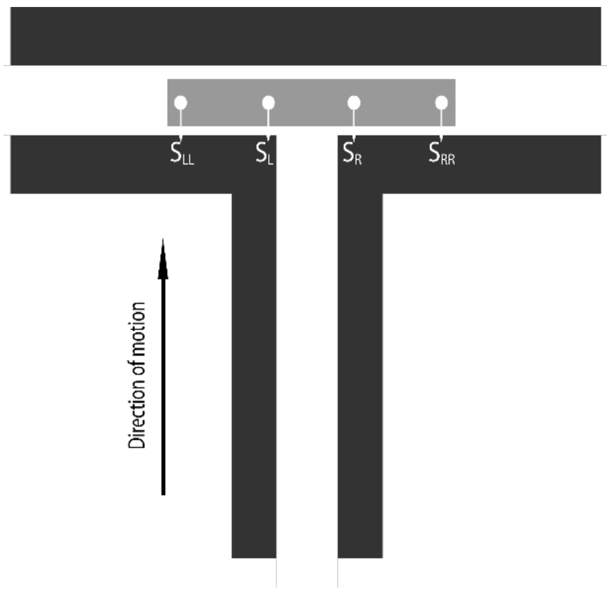

Figure 15: T-junction detection

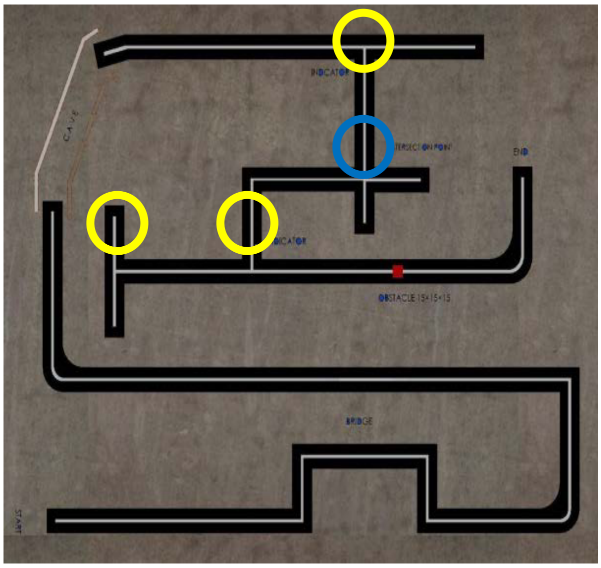

Figure 16: T-junctions and + junction on track

Consider our sample path, there were three T-junctions and one + junction (figure 16). At first T-junction, our robot detected it as 90 degree right bend because here sensor S_R and sensor S_{RR} was in white line. So, it took 90 degree right turn using 90 degree turn algorithm. Which was correct option, so we did not need to take any additional activities for the junction.

After that robot reached at + junction. As mention earlier, our robot detect this + junction as T-junction. This was first T-junction detection by the robot and here its counter value was 0.

Here, robot had three options and among them, 90 degree right turn was correct option. So, we set a condition on our algorithm such that, if robot detect any T-junction and counter variable's value is 0, then take 90 degree right turn and set counter variable's value to 1.

At second T-junction robot had to take left 90 degree turn and here counter variable's value was 1. So, we set the condition, if robot detect any T-junction and counter variable's value is 1, then take 90 degree right turn and set counter variable's value to 2. Third T-junction did not appear on robot path. By, following this strategy, we can that, our robot have the ability to follow any path with any number of T-junctions or + junctions.

Again, sensor position on robot is another important issue to consider. Though, we have used four sensors placed in a straight line, distance between sensors is very important. For example, if we increase the distance between S_R and S_L robot will shake too much and will create difficult angels. This also reduce the speed of the robot.

On the other hand, for two outer sensors (S_{LL} and S_{RR}), if we placed these sensors too far from the inner sensors (S_L and S_R), these may lead the robot in a complex position when entering in any 90 degree bend. In reverse, if they are two close they would not able to detect any 90 degree bend. Figure 16 illustrate this two problem.

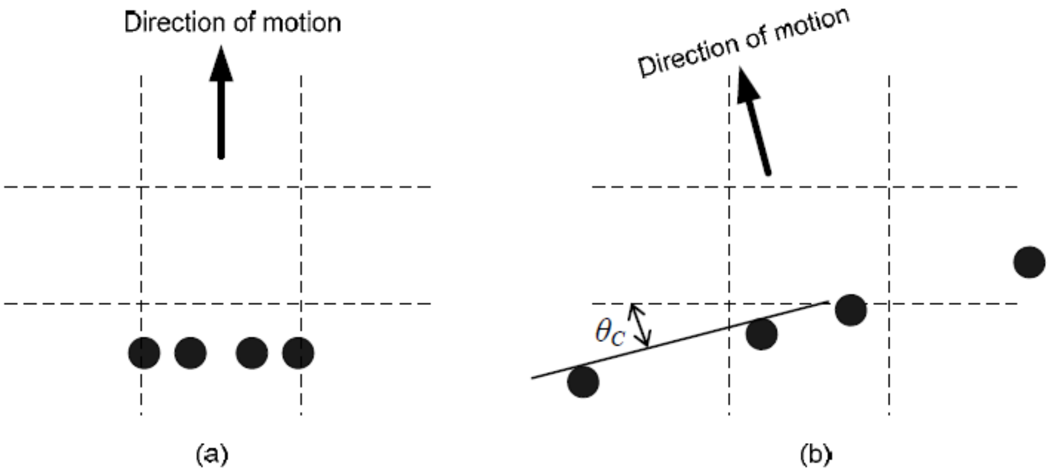

Figure 16: Sensors' arrangement: (a) sensors placed too near, (b) sensors placed too far

To determine the distance between outer two sensors, we can use equation 1 [6].

$$\text{width of the line} < |S_{RR} - S_{LL}| < \frac{\text{width of the line}}{\tan \theta_c} \quad (1)$$

Here, θ_c is the allowed critical angle to entry in a 90 degree bend.

If we allowed maximum critical angle 20 degree (what is quite reasonable) to entry in a 90 degree turn, the distance between outer two sensors would be

$$|S_{RR} - S_{LL}| < \frac{\text{width of the line}}{\tan \theta_c}.$$

Here width of the line has been determined by the RoboTour was 4 cm (RoboTour 2015). So the distance

$$|S_{RR} - S_{LL}| < \frac{4\ cm}{\tan 20^0} < 11 \text{ cm (approximately)}$$

So, we set the distance between sensors S_{RR} to S_R and S_{LL} to S_L 5 cm for both cases as mentioned earlier.

4. Conclusion and Future Recommendations

Though, the use of line follower robot is very common today. But, select optimal sensors to achieve high efficiency is still now a great challenge. This paper gives a solution on this matter. It discuss about the algorithm through which a line follower robot able to follow any 90 degree bends, T or + junctions using minimum number of sensors. It also discuss about the number of sensors a robot need to follow a simple line and calculate the minimum number of sensors a robot should have if it have to follow a path with critical sections like 90 degree turn, T-junction and + junction using this algorithm. Our robot successfully able to follow the sample path. Which proves the effectiveness of our proposed method. However, this robot had some limitations too. When we try to increase robot's speed, some problem occurs on 90 degree bend and junctions. It was because of the momentum of the robot it got through higher speed. We, solved this problem by reducing robot's speed at bends and junctions. In future we can extend our work on this point to ensure efficiency at higher speed.

References

[1]. Pakdaman,M.Sanaatiyan,M.M.,"Design and Implementation of Line Follower Robot,", Second International Conference on Computer and Electrical Engineering TCCEE '09, vol.2, pp.585-590,Dec.2009.

[2]. PriyankPatil,"AVR Line Following Robot," Department of Information Technology K. 1. Somaiya College of Engineering Mumbai, India. Mar 5, 2010.

[3]. Aziz, A.R.A., Designing and Constructing an Automatic Steering Vehicle (AGV), in Dept of Electrical Engineering. 2004, University Tenaga Nasional: Malaysia. p. 50.

[4]. Ashari, S., Line, Gap and Colour Sensors, in Dept of Electical Engineering. 2004, University Tenaga Nasional: Malaysia. p. 70.

[5]. "RoboTour and Programming Contest, 2015", Organized by IEEE RUET Student Branch, Bangladesh

[6]. M. Zafri Baharuddin, Izham Z. Abidin, S. Sulaiman Kaja Mohideen, "Analysis of Line Sensor Configuration for the Advanced Line Follower Robot", Department of Electrical Engineering, Universiti Tenaga Nasional, Malaysia

[7]. Bajestani, S.E.M., Vosoughinia, A., "Technical Report of Building a Line Follower Robot" International Conference on Electronics and Information Engineering (ICEIE 2010), vol 1, pp. v1-1 v1-5, 2010.