A New Approach for Line Following Robot Using Radius of Path Curvature and Differential Drive Kinematics

Jitendra Singh
Department of Electrical Engineering,
Sardar Vallabhbhai National Institute of Technology
Surat, India
jitt776@gmail.com

Prashant Singh Chouhan
Computer Science Department,
M.B.M. Engineering College
Jodhpur, India
prashantsinghforyou@gmail.com

***Abstract*— This paper presents a new idea about the development of line following algorithm for a differential drive mobile robot. The paper describes in detail this idea using Geometry to determine the radius of path curvature and based on it the desired difference between angular velocity of two wheels is calculated using Differential drive Kinematics. This robot uses an array of IR reflective sensors to find out the degree to which a circular arc deviates from a straight line. In this scheme a single PID controller is used to control the desired angular velocity for both wheels. To maintain the circular path of desired radius with constant linear velocity, the output of same PID controller is added in left wheel actuator and subtracted from right wheel actuator. Therefore the desired Radius of circular arc is followed by robot. `**

Keywords*— *Circular Path Planning; Differential Drive Mobile Robot; Line Following Robot.

I. Introduction

Line following is one of the most useful mobile robot which has many applications in industrial, Domestic, Guidance and Health Care Management System [1]. Generally, two wheel differential drive mobile robot consists of chassis with two wheels which is actuated by electrical motors and one caster wheel for balancing purpose.

Several researchers have published papers on the design of line following robot. Some of them consider only Bang-Bang controller [2], while others consider PID controller [3], [4]. Kinematic modeling and control of Differential Drive mobile robot were introduced [5], [6]. In [7], the entire trajectory control of the mobile robot has been discussed. Mathematical and Differential Drive Kinematics modeling to control the trajectory is introduced in [6]. According to the Angular deviation from the straight line, error is calculated and this error is input to the PID controller. The controller then gives the appropriate control signal to the actuator and controls or reduces the error to zero [4]. While our approach is different, we develop the PID controller to control the desired difference between angular velocities of two actuators to follow desired radius of curvature. The main idea of this paper is to develop a line following algorithm which will enable the robot to follow the line with constant linear velocity. However the output of PID is sent in such a way that the linear Velocity will be held constant, i.e. speed of left and right motor is modified by same amount but with negative correlation according to the PID controller's output.

The paper is organized as follows. In section II we present the main contribution of paper in Problem formulation. Mathematical modeling of overall system is presented in section III. Section IV describes the Control design of overall system. Simulation of overall system in MATLAB/Simulink and Hardware Implementation are provided in sections V and VI respectively. Finally, further work and conclusions are given in sections VII and VIII respectively.

II. Problem Formulation

We consider a path or line MN as given in Fig. 1, which have two section. First section MP is the straight line and another section PN is the circular arc of radius R, which is to be followed by robot. Referring to Fig. 1, SM is the center line of robot. When center line of wheels EF intersects at point P, then the path deviation ΔX from center line(PS) of robot is measured by IR sensor array. In this case IR sensor array read the deviation ΔX at point A. ΔY is the fixed distance between point A and Q. To follow the circular arc of radius R, we have to calculate R in terms of ΔX and ΔY.
From Fig. 1 OAQ is the right angle triangle, in triangle OAQ by using Pythagoras theorem, we can say:

$$(OA)^2 = (OQ)^2 + (AQ)^2 \tag{1}$$

From Fig. 1 OA is R, OQ is $(R - \Delta X)$ and AQ is ΔY.then

$$(R)^2 = (R - \Delta X)^2 + (\Delta Y)^2 \tag{2}$$

Solving Equation (2), we get:

$$\mathrm{R} = \frac{\Delta X^2 + \Delta Y^2}{2\Delta X} \tag{3}$$

Equation (3) indicates that when ΔX is zero, then R is infinite, means curvature is zero, i.e. it is a straight line This calculated radius R is used to determine the difference of angular velocity to follow the circular arc of radius R. However, this will be explained further in section III.

978-1-5090-4874-8/17/$31.00 ©2017 IEEE

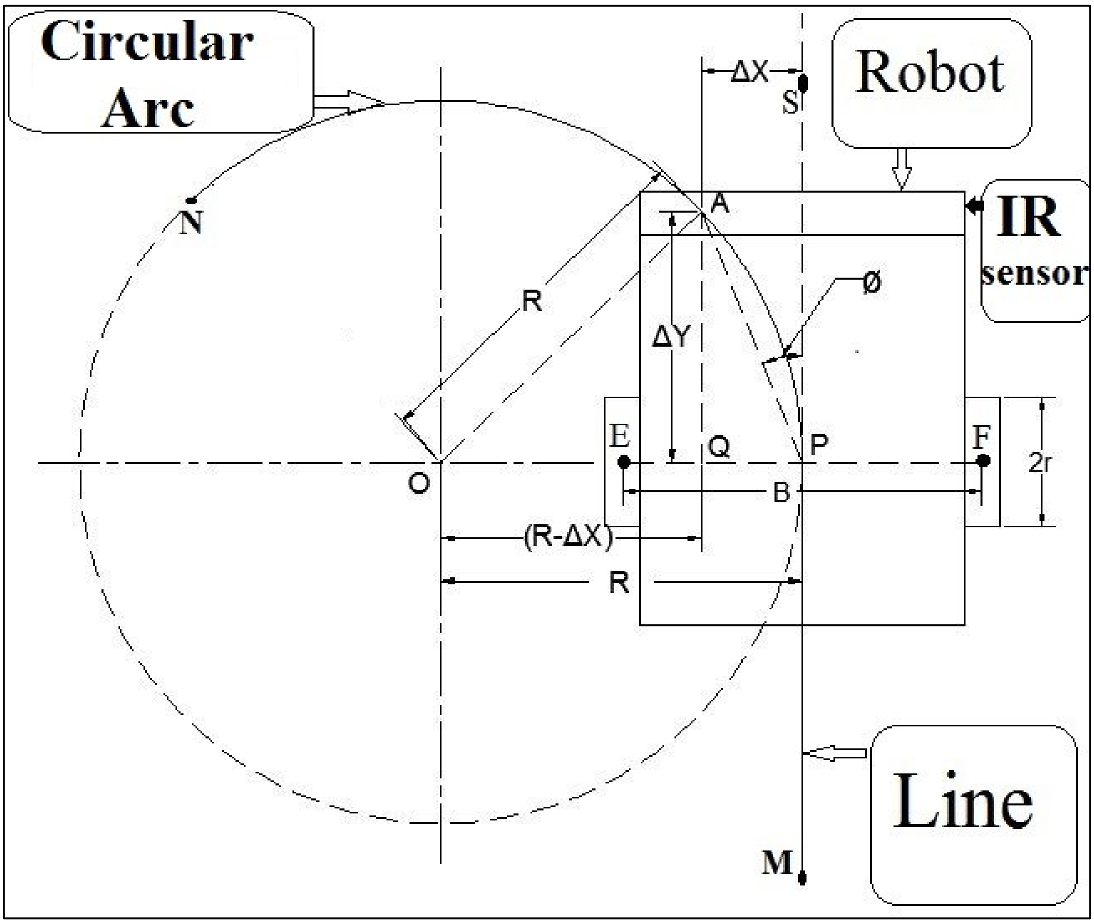

Fig. 1. Calculation of radius of circular arc using geometry

III. Mathematical Modelling Of Overall System

In this section, we first estimate the linear velocity of both left and right wheel V_L and V_R respectively. Robot is at point P(x,y) in XY plan, shown in Fig. 2. Referring to Fig. 2, R and $(R+\frac{B}{2})$ and $(R-\frac{B}{2})$ are the radius of ARC2 and ARC3 and ARC1 respectively. ICC is the instantaneous centre of curvature. ω is the angular velocity of robot around ICC. B is the distance between two wheels.

A. Kinematic Modelling

Referring Fig. 2, using relation between linear and angular velocity:

$$V = \omega R \tag{4}$$

$$V_L = \omega(R+\frac{B}{2}) \tag{5}$$

$$V_R = \omega\left(R-\frac{B}{2}\right) \tag{6}$$

Equation (5) and (6) indicates the linear velocity of left and right wheel respectively in terms of radius R and width B. Subtracting Equation (6) from (5), we get

$$V_L - V_R = \omega B \tag{7}$$

Both wheels are identical therefore radius of left and right wheel is r, then ω_L and ω_R are angular velocities of left and right wheel respectively and it is given by wheel encoder, then:

$$V_L - V_R = r(\omega_L - \omega_R) \tag{8}$$

By using Equation (7) & (8), we get:

$$\omega B = r(\omega_L - \omega_R) \tag{9}$$

Substituting ω in above Equation from (4), we get above Equation in terms of R:

$$\frac{VB}{R} = r(\omega_L - \omega_R) \tag{10}$$

Equation (10) indicate that to make the curvature of radius R, the difference of linear velocity is equal to $\left(\frac{VB}{R}\right)$.

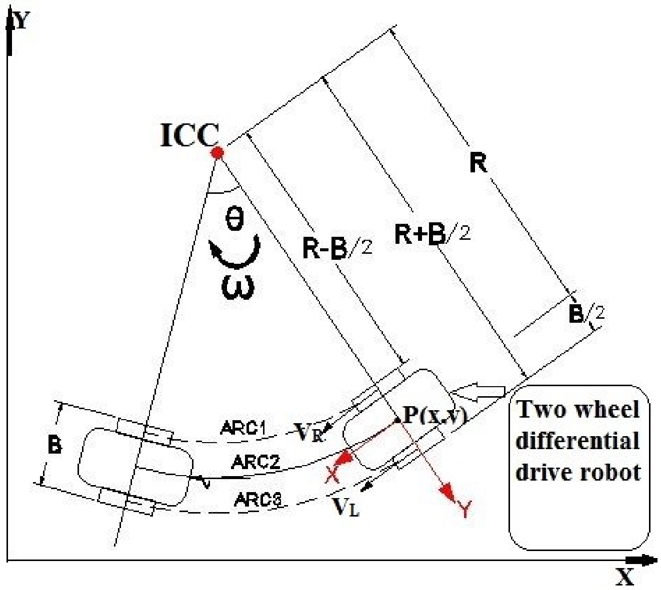

Fig. 2. Differential drive kinematics model.

B. Modelling of PMDC Motor

The relationship between input voltage and angular velocity produced by the PMDC motor should be known for designing the controller. So an appropriate model for dc motor has to be derived which relates input voltage and angular velocity of a PMDC motor [6].

The transfer function between the angular velocity output and voltage input can be calculated as:

$$\frac{\omega(s)}{\mathrm{V_{in}}(s)} = \frac{K_t}{[(L_a J_m)S^2 + (R_a J_m + b_m L_a)S + (R_a b_m + K_t K_b)]} \quad (11)$$

Where J_m is the moment of inertia of the rotor, b_m is the viscous friction coefficient, k_t is torque constant, k_b is the back-emf constant, R_a is the armature resistance and L_a is the armature inductance.

IV. CONTROL DESIGN

The entire control design is based on the fact that robot follow the path curvature of desire radius R with linear velocity V is kept constant. i.e.

Adding Equation (5) and (6), we get:

$$V_L + V_R = 2\omega\mathrm{R} \quad (12)$$

Form Equation (12) and (4), we get:

$$V = \frac{(V_L + V_R)}{2} \quad (13)$$

Equation (13) indicates that the linear velocity is the average of linear velocity of left wheel and right wheel

Control design of this system is based on Equation (9), it indicates that the difference of linear velocities $(V_L - V_R)$ of left wheel and right wheel must be equal to ωB. This is controlled by PID controller.

First we calculate R from equation (3), then keep V as constant, then calculate ω from equation (4) according to the desired value of radius R , i.e.

$$\omega = \frac{V}{R}$$

then the desired value of controller will be ωB. Then error signal e is given from the condition of equation (9)

$$\mathrm{e} = \omega\mathrm{B} - \mathrm{r}(\omega_\mathrm{L} - \omega_\mathrm{R}) \quad (14)$$

The main role of controller is to reduce error signal to zero, however, equate equation (14) to zero, i.e.

$$\mathrm{e} = \omega\mathrm{B} - \mathrm{r}(\omega_\mathrm{L} - \omega_\mathrm{R}) = 0$$

Therefore controller is tuned to obtain above condition

According to the above condition, PID controller gives appropriate signal to actuators in such a way that, we add it in input voltage of left motor and at same time subtract it from input voltage of right motor to maintain constant linear velocity, we can say that linear velocity will be constant from equation (13). Only the angular velocity of the robot is controlled.

The control scheme based on Equation (9) and (11) and (14) is shown in Fig. 3.

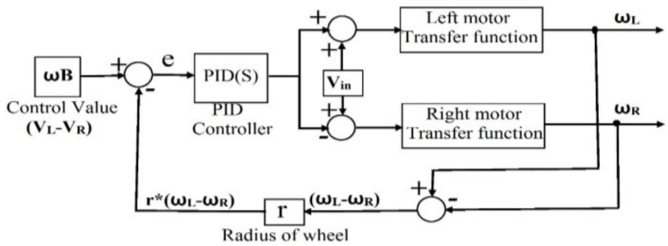

Fig. 3. Block diagram of proposed controller

V. SIMULATION OF OVERALL SYSTEM IN MATLAB/SIMULINK

The simulation of overall system, including the basic transfer function of PMDC motor, PID controller, feedback speed sensor, differential drive kinematics is shown in Fig. 4(a). The subsystem block of control strategies based on Fig. 3. is shown in Fig.4(b).

SimMechanics block diagram is the block representation of 3D model of robot, This 3D model is made in Solidworks CAD software and the interfacing between Solidworks and MATLAB is based on [8]. Mechanics Explorers shows the trajectory of robot in 3D plane and the position of robot is shown in Fig. 5. All system parameters are given in the table 1. For PID controller, the tuning criteria are (i) less settling time is 1.53 sec. (ii) overshoot is 7.3%. (iii) rise time is 0.4 sec. Tuning was done manually using trial and error method.

TABLE I. SYSTEM PARAMETE

S.No.	Parameter	Value	Unit
1	J_m	0.01	Kg.m^2
2	b_m	0.1	N.m.s
3	R_a	1	Ω
4	L_a	0.5	H
5	K_t	0.01	N.m/A
6	K_b	0.01	V/rad/s
7	V	0.1047	m/s
8	B	0.2	m
9	R	0.3	m
10	r	0.02	m

Fig. 6(a) shows the step response of the proposed system. The rise time of the system is about 0.4 seconds which is due to mass inertia of the system and actuation limitations. Fig. 6(b) shows the circular path traced by the robot, which has a radius of 30cm. Fig. 6(c) shows the angular velocity of Left and Right wheels, which is constant with respect to time. The values of angular velocities of Left and Right wheels are 6.97rad/sec and 3.48rad/sec respectively. Fig 6(d) shows the linear velocity of differential drive mobile robot, which is constant with respect to time and the value is 10.46 cm/sec.

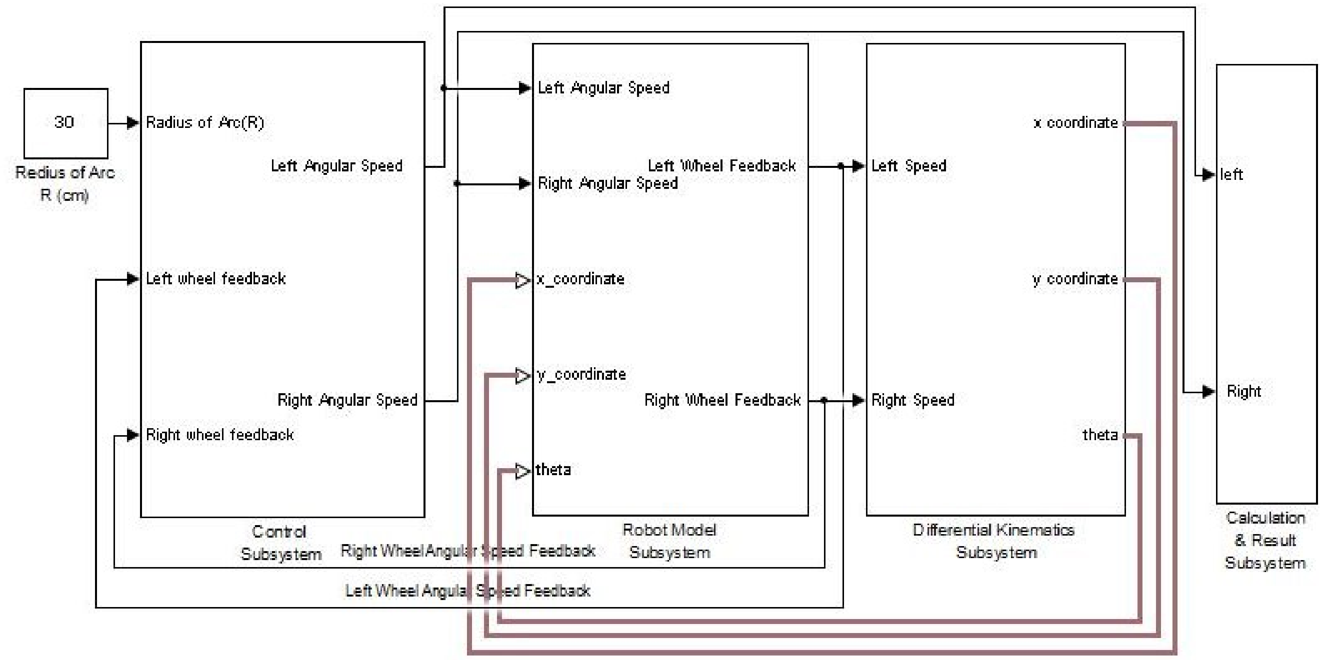

Fig. 4(a). Simulation model of overall system.

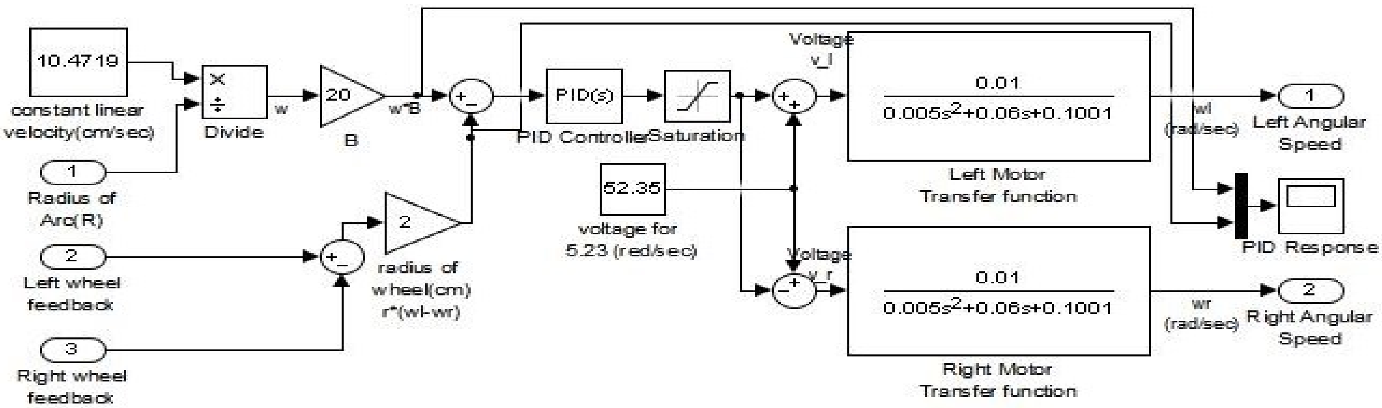

Fig. 4(b). Subsystem of control strategies

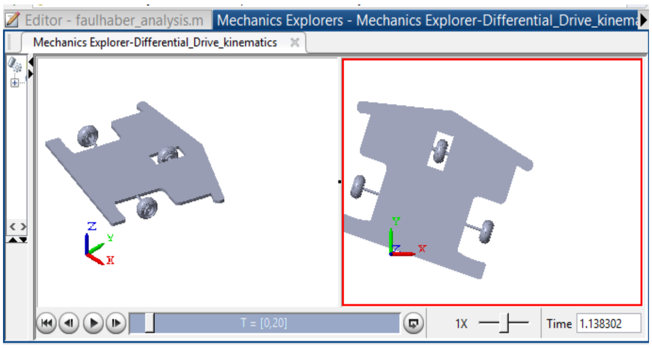

Fig. 5. 3D view in mechanics explorers(MATLAB).

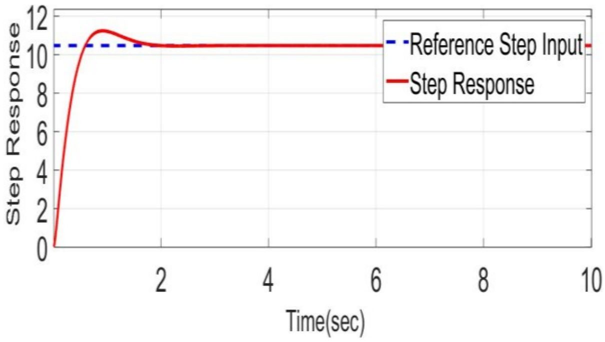

Fig. 6(a). Step response of proposed system.

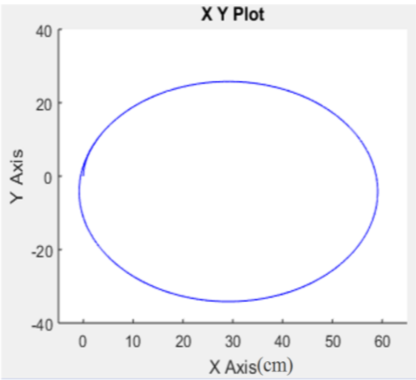

Fig. 6(b). Circular path tracking.

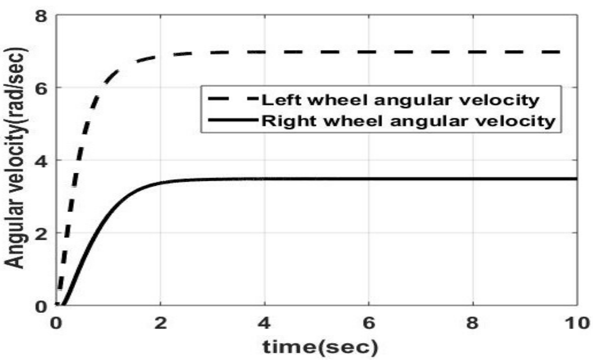

Fig. 6(c). Left and Right wheel angular velocity.

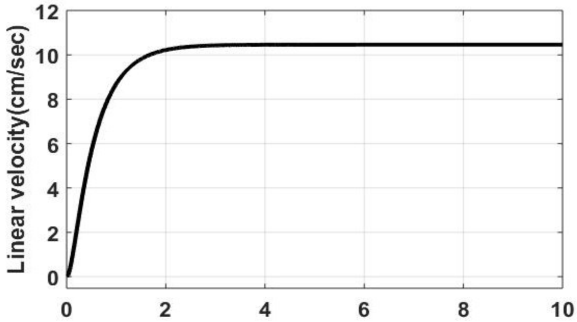

Fig. 6(d). Linear velocity of differential drive mobile robot.

VI. Hardware Implementation

In this paper, we implemented hardware for traversing circular path of desired radius and radius of circular path is manually set, not calculated by IR –sensor as we describe in section II.

The robot used for implementation is shown in Fig. 7(a). It has been developed in laboratory. The system parameters for hardware implementation are taken from Table I. The microcontroller which is used for implementation of control strategies is Arduino mega2560 board having 8 bit Atmega2560 microcontroller. Two 130 rpm dc gear motors provides the actuation. The motors are used for actuation is Faulhaber 1524E006S123 motor with 15/5S141:1K832 gearhead and HES164A encoder. The gear ratio of motor is 141:1. PWM technique is used to control the angular velocity of motors. The motors are driven by a circuit board having L298 H bridge. The maximum input voltage to the motor shall be ±9V.

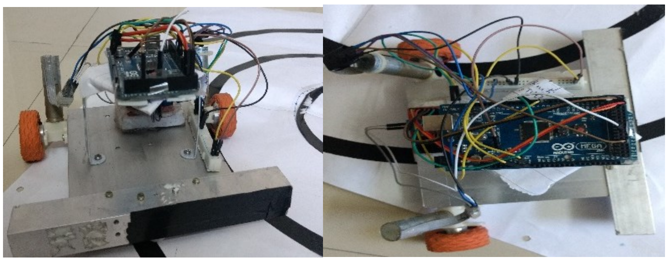

Fig. 7(a). Front and Top view of differential drive mobile robot.

The angular velocities of wheels are measured using Hall effect based wheel encoder. For measuring the angular velocities quickly, external event counter programming is used by connecting the encoder pins to external event counter/timer pins of the microcontroller. The motor shaft rotates 141 times for every 1 time the gearhead output shaft rotates. The interrupt service routine is activated with a completion of loop. The time(t_p) taken for completion of loop is measured in seconds. The numbers of pulses counted by external event counter is N in the time t_p. Angular velocity in rad/sec is given by :

$$\omega = \frac{N*2\pi}{141*t_p} \qquad (15)$$

The measured value of difference between angular velocities acts as feedback to the PID controller. The control algorithm is executed every 100ms. The circular path tracing of robot was tested with 30 cm radius, which is shown in Fig. 7(b). The maximum deviation from the path is 5 cm. linear velocity response with time is shown in Fig. 7(c), which is not constant at many points and value of linear velocity is 10.47 cm/sec. These errors are due to noise in the encoders.

Simulation and experimental results are presented in section V and VI respectively for validation.

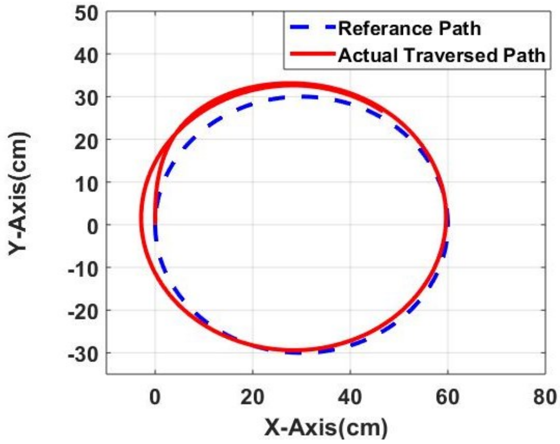

Fig. 7(b). Circular path tracking by real robot.

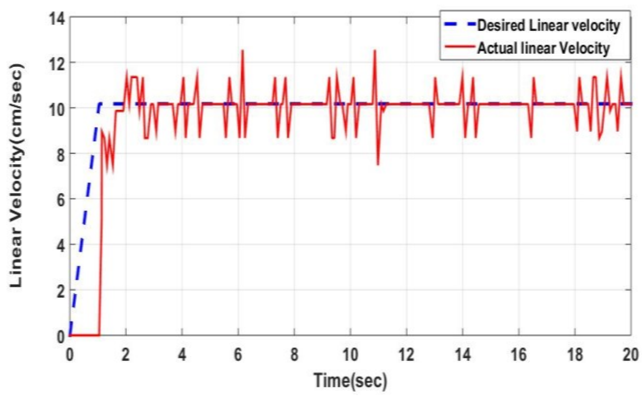

Fig. 7(c). Linear velocity of real robot.

VII. Further Work

Future work will include the design of a Line following robot based on the concept described in section II. The radius of circular arc will be the input to the control system, which has described in section IV. The deviation(ΔX) of circular arc from central line of robot will be calculated by using weight average method. The weight of ΔX will be measured by using IR-sensor array and radius will be calculated by using Equation (3). Hence robot will follow the circular arc of desired radius to follow the line.

VIII. Conclusions

A new control strategy for circular path tracking of a differential drive mobile robot is presented. The advantage of this method is that it is simpler and the robot follows the circular path with constant linear velocity and it uses only a single PID controller to control the circular path trajectory. Simulation results show that the robot can follow the radius of any value. Simulation and experimental results are presented for validation and to illustrate the good performance of the proposed approach. This method can further be used to implement it on line following robot which has been proposed in this paper.

References

[1] Punetha, Deepak, Neeraj Kumar, and Vartika Mehta. "Development and Applications of Line Following Robot Based Health Care Management System." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 2, no. 8 (2013): 2446-2450.

[2] Pakdaman, Mehran, M. Mehdi Sanaatiyan, and Mahdi Rezaei Ghahroudi. "A line follower robot from design to implementation: Technical issues and problems." In Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, vol. 1, pp. 5-9. IEEE, 2010.

[3] Engin, Mustafa, and Dilşad Engin. "Path planning of line follower robot." In Education and Research Conference (EDERC), 2012 5th European DSP, pp. 1-5. IEEE, 2012.

[4] Balaji, Vikram, M. Balaji, M. Chandrasekaran, and Irraivan Elamvazuthi. "Optimization of PID Control for High Speed Line Tracking Robots." Procedia Computer Science 76 (2015): 147-154.

[5] Maulana, Eka, M. Aziz Muslim, and Akhmad Zainuri. "Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot." In Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), 2014, pp. 93-98. IEEE, 2014.

[6] Salem, Farhan A. "Dynamic and kinematic models and control for differential drive mobile robots." International Journal of Current Engineering and Technology 3, no. 2 (2013): 253-263.

[7] Nitulescu, Mircea. "Solutions for Modeling and Control in Mobile Robotics." Journal of Control Engineering and Applied Informatics 9, no. 3, 4 (2007): 43-50.

[8] Yilmaz, Ercan Nurcan, and Onur Battal. "Using active learning in motor control and matlab simulation." Journal of Automation and Control Engineering Vol 3, no. 4 (2015).