2019 6th International Conference on Control, Decision and
Information Technologies (CoDIT'19) | Paris, France / April 23-26, 2019

Development of Waypoint Tracking Controller for Differential Drive Mobile Robot

Robins Mathew, and Somashekhar S Hiremath

Abstract— **Waypoints are control points through which a robot has to move to reach the goal. Waypoint defined trajectories may have discontinuities in the curvature. The waypoint tracking controller should be designed to take care of such discontinuities. In the current work a control strategy is developed for waypoint trajectory tracking control of differential drive mobile robot. The waypoint trajectory requiring sharp turns is considered in the design of controller. Further the control gains are optimized for reducing the tracking errors. The performance of the controller is analyzed using simulation study. The obtained results suggest that the new controller is a more suitable candidate for waypoint tracking.**

I. INTRODUCTION

Wheeled mobile robots are getting more attention in today's fast-growing market. Such robots are employed for inventory management, factory automation, military surveillance, etc. These robots are classified into different categories such as car-like robot, omnidirectional robot, and differential drive robots; of which differential drive robots is a prominent class. Differential drive robots are suitable in many sectors because of its simple structure and are easy to be produced.

Trajectory tracking controllers are used for controlling the robot through the given trajectory. Several control strategies for trajectory tracking of differential drive mobile robots were developed. One of the simplest methods is to use a switching controller which utilizes straight motion and in place turning control commands [1], [2]. In this method, the robot is first oriented towards the direction of the trajectory then moved forward. In case of any error in the direction of travel or change in the direction of travel, the robot stops and reorients itself again. Although this method is simple, the problem of frequent switching of control command makes this controller, less popular. This has lead to the development of other control methods.

Non-linear controllers are most popular in case of differential drive mobile robot control and includes Fuzzy logic approach [3] [4], neural network based control [5], potential field based control [6], backstepping method [7], inversion method [8], image-based trajectory tracking control [9] etc. Lyapunov theory can be utilized for designing stable non-linear controllers which could control the robot [10]. However, most of the above-mentioned controllers are mainly used in situations where the reference trajectory is smooth and continuous. Waypoint Navigation Systems (WNS) can be used as an alternative to a continuous trajectory [11] .Waypoints are abstract points through which the robot has to be controlled to reach the goal. Path planner which is the upper layer in layered control architecture usually provides waypoint trajectory as the reference trajectory. Here, the reference trajectory is considered to be the line connecting the waypoints. Such trajectories will have sharp turns. The robot might require to tightly follow the line connecting the waypoint to avoid any obstacles nearby. Usually, an approximated smooth and continuous reference trajectory will be generated from the waypoints using a path-smoothening algorithm and given to the smooth trajectory tracking controller. This path smoothening require additional computation. The optimality of the track, as well as the error due to approximation, are some of the issues in such scenarios. Another problem is that the smoothening of the path does not guarantee that the robot will track the trajectory. The smoothened trajectory which works for one robot may not work for another robot. Hence, it is better to directly use the line connecting the waypoints as the reference trajectory.

Controller for tracking smooth trajectory, if appropriately modified could become a suitable candidate for waypoint tracking. Many researchers have tried different strategies in this regard. Sliding mode control has been utilized for waypoint tracking control of spherical robot actuated with a differential drive system [12]. Curvature path-tracking controller can be used for waypoint navigation of car-like robots [13]. Artificial potential field method based control is also a possible solution for waypoint tracking control where simultaneous localization and mapping is possible. Reaching the goal approach is also a solution for waypoint tracking control [14], [15]. In [16], the reference velocities, as well as control gains of a smooth tracking controller is optimized to attain waypoint tracking. However, all these controllers consider the problem as point to point tracking problem rather than trajectory tracking problem.

Few works in the literature propose the direct use of waypoint trajectory [17], [18]. Waypoint navigation of a differential drive robot is considered in [18]. This controller was able to control the robot through a given raw sequence of waypoints as well as could able to take into account different constraints such as velocity limits, acceleration limits and other precision constraints. However, in this work the waypoints were selected from a smoothened trajectory in such a way that sharp turns will not occur. It was observed that these controllers are not suitable when the reference waypoint trajectory has sharp turns. Here a turn having a turning angle greater than $2\pi/3$ is considered as sharp turns.

Robins Mathew is with Indian Institute of Technology Madras, Chennai, Tamil Nadu, India, 600036 (e-mail: connectrobins@ gmail.com).

Somashekhar S Hiremath is with Indian Institute of Technology Madras, Chennai, Tamil Nadu, India, 600036 (+91 44 2257 4681; fax: +914422574652; e-mail: somashekhar@iitm.ac.in).

Usually, path planner generated trajectories will have few such sharp turns.

Most of the waypoint tracking controllers make the robot move from one waypoint to other (point to point tracking), and could not assure that the robot will move through the reference waypoint trajectory which is the line connecting the waypoints. The reason is that the control equation does not account the cross-track error. The distance between the robot's position and the trajectory is called cross-track error. In case of waypoint trajectory, if the robot moves through the line connecting the waypoints, the cross-track error could be avoided.

In the current work, an attempt has been made to develop a waypoint tracking controller for differential drive mobile robot. The controller is designed for tracking waypoint trajectories requiring sharp turns in the trajectory. Such a trajectory needs tight maneuvers to follow the trajectory accurately. Tight manures are situations where precise control over angular and linear velocity is required for proper tracking of the trajectory. Obstacle very close to the reference trajectory is a situation where tight maneuvers are necessary. This new control method is a modified approach of the strategy presented by kanayama et.al. [19] which is one of the widely used control methods for trajectory tracking control of a differential drive robot. The controller proposed by kanayama et.al is referred as 'standard controller' here onwards. The new controller can reduce the limitations of the standard controller such as the presence of cross-track error and unsuitability of the standard controller for waypoint trajectory tracking especially when sharp turns are there in the trajectory.

This paper is organized as follows: Section 2 explains the kinematic model of the robot and controller design. The optimization of the controller and the experimental result is discussed in Section 3. Section 4 presents the summary of the work and conclusions

II. Modeling of the Robot and Controller Design

Differential drive robot has two independently actuated wheels and one or more passive wheels to balance the robot. In Fig. 1, a waypoint tracking differential drive robot is shown. Here, the trajectory to be followed by mobile robot is specified as the path connecting the waypoints from a source ($q_{w0}=(x_{w0}, y_{w0})$) to a destination ($q_{wn}=(x_{wn}, y_{wn})$) through a set of intermediate waypoints($q_{wk}=(x_{wk}, y_{wk})$). Here k is an integer between zero and the total number of waypoints (n). The robot needs to tightly follow the line connecting the waypoints to achieve proper tracking of the trajectory.

The robot is considered to be moving on a surface where an inertial frame (X_I, Y_I) is defined. Posture (p_i) of the mobile robot at any given instant i with respect to the inertial coordinates is given in (1).The robot coordinates are assumed to be the coordinates of the perpendicular intercept to inertial frame axis from the center of the line connecting the actuated wheels centers. The locus of the points(x_i ,y_i)over the time is considered as the trajectory tracked by the mobile robot.

The differential drive robot is assumed to be under pure rolling. Hence it is possible to rely on the kinematic model of the robot given in (2). Here, v_i and ω_i are the linear and angular velocity of the mobile robot at i^{th} instance.$\dot{x}_i$, is the horizontal velocity component, $\dot{y}_i$ is the vertical velocity component and $\dot{\theta}_i$ is the angular velocity components.

$$p_i = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} \quad (1)$$

$$\dot{p}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \quad (2)$$

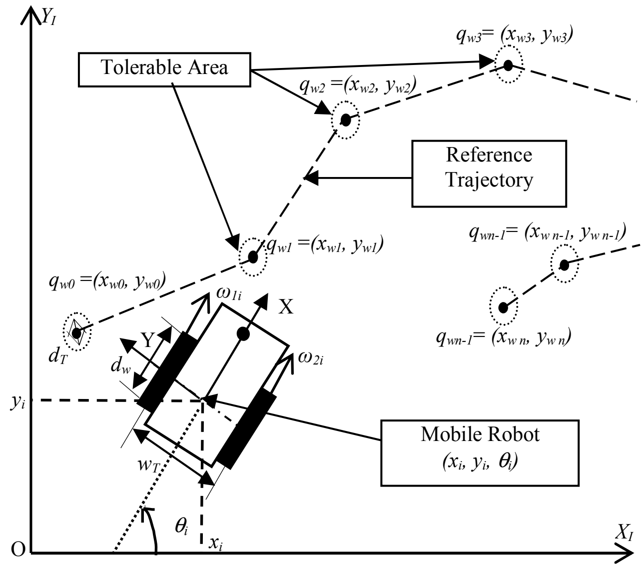

Figure 1 Differential drive robot tracking waypoint trajectory

For controlling the robot through the given trajectory, a controller has to generate linear control velocity and angular control velocity depending on the posture error of the robot with respect to the reference waypoint. Transformation of the posture error ($p_{ie}=[x_{ie}, y_{ie}, \theta_{ie}]^T$) of the mobile robot with respect to k^{th} waypoint posture ($q_{wk}=[x_{wk}, y_{wk,}\theta_{wk}]^T$) to the local coordinates X, Y, θ of the mobile robot is given in (3).This error is utilized as a feedback to the robot controller.

$$p_{ie} = \begin{bmatrix} x_{ie} \\ y_{ie} \\ \theta_{ie} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{wk}-x_i \\ y_{wk}-y_i \\ \theta_{wk}-\theta_i \end{bmatrix} \quad (3)$$

Where θ_{wk} is calculated as:

$$\theta_{wk} = \tan^{-1}\left(\frac{(y_{wk}-y_i)}{(x_{wk}-x_i)}\right) \quad (4)$$

In case of a differential drive robot, the linear velocity v_i and angular velocity ω_i of the robot can be written as a function of wheel velocities of the actuated left wheel ω_{1i} ,and right wheel ω_{2i} as given in (5). Here w_T is the wheel track width and $d_w$is the wheel diameter. For calculating the velocity, the wheels are assumed to have the same diameter.

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \frac{1}{4} \begin{bmatrix} d_w & d_w \\ \frac{2d_w}{w_T} & \frac{-2d_w}{w_T} \end{bmatrix} \begin{bmatrix} \omega_{1i} \\ \omega_{2i} \end{bmatrix} \quad (5)$$

The Controller proposed by Kanayama et.al. [11] is one of the most popular controllers in case of trajectories with continuous curvature. This control equation is given in (6). Here, v_{ci}, ω_{ci} are control velocities, v_{ri}, ω_{ri} are reference velocities of the mobile robot and k_1, k_2, k_3 are the control gains. The required motor control velocities to execute the calculated control action may be very large and may not be suitable for implementing in practical situations. Hence, the control velocity is limited with the help of soft limits before applying to the actual robot.

$$\begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix} = \begin{bmatrix} v_{ri} \cos\theta_{ie} + k_1 x_{ie} \\ \omega_{ri} + k_2 v_{ri} y_{ie} + k_3 v_{ri} \sin\theta_{ie} \end{bmatrix} \tag{6}$$

Equation (7) describing the transformation of velocities of the mobile robot into the wheel angular velocities can be obtained by taking the inverse transform of (5) and substituting v_{ci} and ω_{ci} instead of v_i and ω_i. The left motor and right motor of the robot has to run at an angular velocity of ω_{1ci} and ω_{2ci} to execute the control action. The wheel velocities can be further limited by soft limits to satisfy motor constraints.

$$\begin{bmatrix} \omega_{1ci} \\ \omega_{2ci} \end{bmatrix} = \frac{1}{d_w} \begin{bmatrix} 2 & w_T \\ 2 & -w_T \end{bmatrix} \begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix} \tag{7}$$

When the controller given by (6) is utilized for waypoint trajectory tracking, it was found to be failing in many situations. The controller was unable to make the robot follow the required trajectory connecting the waypoints. The controller performance is found to degrade as the angle of turn required to go from one waypoint to other increases. Fig. 2 shows three scenarios where the robot has to make a turn of л /4, л/2 and 3л /4 respectively to track the given trajectory. It can be observed that the controller is unable to make the robot to reach the required trajectory. Cross-track error increases as the turn angle increases. The results indicate the possibility of improvement to the existing controller.

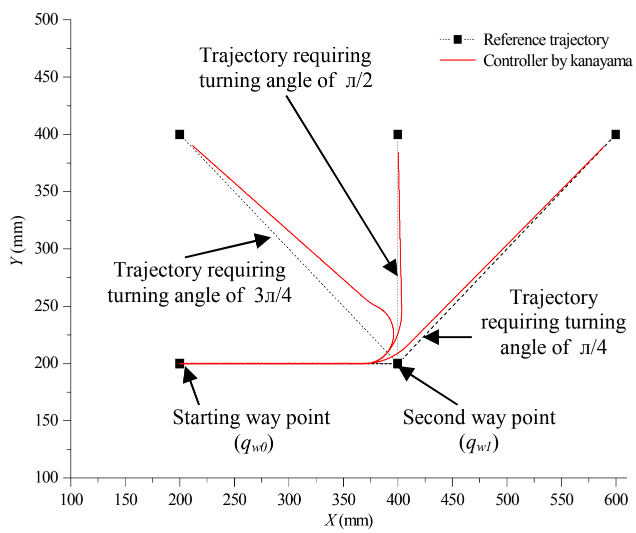

Figure 2 Trajectory tracked by the robot when the standard controller is utilized for waypoint trajectory tracking

In (6) The angular control velocity ω_{ci} depends on components of posture error y_{ie} ,θ_{ie}. The problem of cross-track error is arising due to components of posture error y_{ie} and θ_{ie} simultaneously becoming zero before the robot reaches the required trajectory. Whenever the heading direction of the robot is equal to the required heading direction, components of posture error become zero. From (3), (4) and (6), this possible pitfall of the control strategy can be predicted. Hence, when y_{ie},θ_{ie} become zero, ω_{ci} also become zero and will stay zero afterwards. The robot will move forward only. In Fig. 3 this condition is shown. The figure shows tracking the given waypoint trajectory using the standard controller proposed by kanayama et.al.

The robot starts from waypoint q_{w0} and go to the waypoint q_{w2} through the intermediate waypoint q_{w1}. For proper tracking of the trajectory, the robot needs to go through the line connecting q_{w0} and q_{w1}, q_{w1} and q_{w2}. The controller will always try to reorient the robot towards the waypoint even though there is a cross track-error. When the robot takes the turn at q_{w1} and orient itself facing waypoint q_{w2}, the components of posture error y_{ie}, θ_{ie} will both become zero even though there is a finite non-zero cross-track error. This results in robot going to the waypoint q_{w2} through a trajectory shown in fine dotted line. The figure also shows an obstacle near the reference trajectory. The robot will collide with the obstacle if it does not tightly track the trajectory.

The improved control equation for solving this problem should have a control parameter which will try to reduce the cross-track error. Adding this control parameter will make the robot move towards the required trajectory instead of orienting towards the waypoint. This in turn reduces the cross-track error. As the cross-track error decreases, the controller should make the robot reorient towards the waypoint, thus ensuring trajectory tracking as well as reaching the waypoint. The new control law is given in (8). Here, T_{ie} is the cross-track error as given in (9).

$$\begin{bmatrix} v_{ci} \\ \omega_{ci} \end{bmatrix} = \begin{bmatrix} v_{ri} \cos\theta_{ie} \\ \omega_{ri} + k_1 v_{ri} T_{ie} + k_2 v_{ri} \sin\theta_{ie} \end{bmatrix} \tag{8}$$

$$T_{ie} = \sin\phi(x_i - x_{wk}) - \cos\phi(y_i - y_{wk}) \tag{9}$$

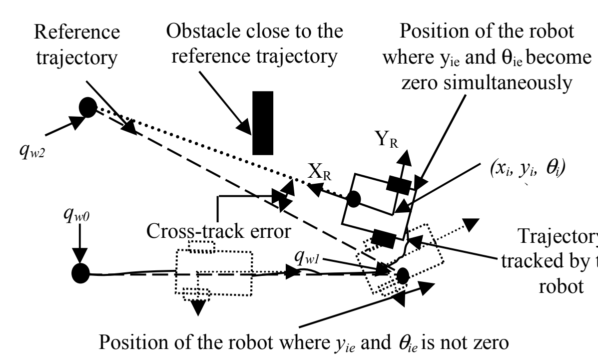

Figure 3 Condition where the component of posture error y_{ie} and θ_{ie} become zero

In (9), ϕ is defined as the angle made by the line connecting current waypoint with the previous waypoint given by (10). Here *wk-1* stand for *k-1*th waypoint.

$$\phi = tan^{-1}\left(\frac{y_{wk} - y_{wk-1}}{x_{wk} - x_{wk-1}}\right) \quad (10)$$

The reference waypoint is updated and changed to next waypoint based on (11), ie 'w*k*' is updated to '*wk+1*'when robot reaches a distance less than d_T from k^{th} waypoint. For example, reference waypoint will be changed from 3 to 4 if the robot reaches a point which is at a distance less than d_T from waypoint 3. Here d_T is the diameter of the tolerance area which is defined manually by the user based on the requirement. The robot is considered to be at the waypoint if it comes within this distance d_T from the given waypoint.

$$\sqrt{{x_{ie}}^2 + {y_{ie}}^2} \le d_T \quad (11)$$

The architecture of the waypoint tracking controller is shown in Fig. 4. It consists of a waypoint planner which gives a set of waypoints from q_{w0} to q_{wn} which defines the reference trajectory. The waypoint selector module changes reference waypoint to the next waypoint on satisfying the criteria given in (11). The error estimation block uses (3) to calculate trajectory tracking error. The waypoint tracking control unit utilizes (8). Motor controller module utilizes (7) to generate motor control commands. The posture estimation module calculates the new poster of the robot with the help of odometry or by using any other localization method. The new posture will act as the feedback to the error estimation block.

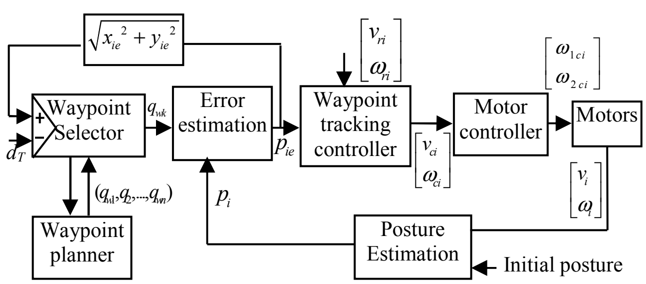

Figure 4 Architecture of waypoint tracking controller

III. Result and Discussion

The controller proposed in the current work is compared with the standard controller. Control gains of both the controllers are first optimized with the help of Multi-Objective Particle Swarm Optimization (MOPSO) technique to minimize cross-track error and velocity tracking error. The aim of the optimization is to find out the combination of control gain which can minimize the average cross-track error and average velocity tracking error. Average cross-track error and average velocity tracking error is calculated as given in (12) and (13) respectively.

$$\text{Average cross-track error} = \frac{\sum_{i=0}^{i=n} abs(T_{ie})}{n} \quad (12)$$

$$\text{Average velocitytracking error} = \frac{\sum_{i=0}^{i=n} abs(v_i - v_{ri})}{n} \quad (13)$$

Simulation-based optimization is carried out here. Each of the controllers is utilized to track a trajectory requiring turning angle of 3л/4. The trajectory followed by the robot and velocity of the robot when tracking the trajectory was calculated from this simulation. The control gains for these simulations were selected by the MOPSO. The simulation was carried out in MATLAB. The MOPSO utilized 5000 evaluation for optimizing each of the controllers. Each evaluation consists of a simulation of the controller using the control gains selected by MOPSO. The control gains selection carried out by the optimization depends on the results of the previous evaluation with an aim to reduce the tracking errors. The process when repeated over 5000 evaluations will find out control gains which can minimize the tracking error. The MOPSO provide Pareto front which consist of points corresponding to the set of possible values of the control parameters which can minimize the tracking errors. Fig. 5 (a) and (b) shows the optimization result as a Pareto front. Each point in the Pareto front represents a possible set of control gains which could minimize the available problem. Comparison of the Pareto front produced by the two controllers shows that new controller will perform better than the standard controller as the points in the Pareto front generated by the new controller is much closer the origin than any point in the Pareto front produced by the standard controller.

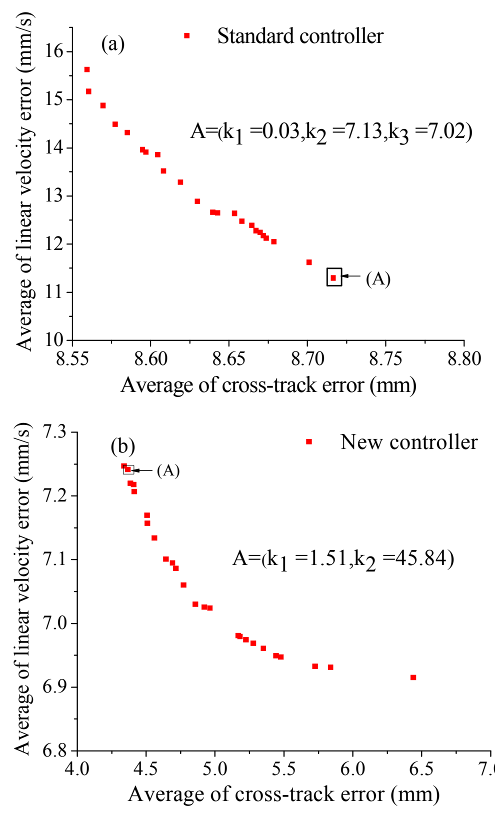

Figure 5 Pareto front produced with the help of multi objective particle swarm optimization method with an objective to minimize tracking error

The set marked as 'A' in the Pareto front have the least mean square error value. This point is considered as the point representing the solution of the multi-objective problem. The control gains corresponding to these points are utilized for further simulation of the controller. Control gains selected for the standard controller is k_1=0.03, k_2=7.13, k_3=7.02. Control gains selected for the new controller is k_1=1.51, k_2=45.84. The simulation is carried out with these control gains for tracking trajectories with different angle of turns. Fig. 6 shows the simulation result. It can be observed that the new controller is able to make the robot reach the given waypoint trajectory much before the standard controller. Table I shows the range in which the control gains were optimized. The parameters used for the simulation is given in Table II.

TABLE I. FIXED PARAMETERS USED FOR SIMULATION

Parameter	Range	
	Standard Controller	***New controller***
k_1	0-50	-
k_2	0-50	0-50
k_3	0-50	0-50

TABLE II. PARAMETER RANGE SELECTED FOR OPTIMIZATION

Parameter	Value
Wheel track width(w_T)	54 mm
Diameter of wheel(d_w)	40 mm
Reference linear velocity (v_{ri})	40 mm/s
Reference angular velocity (ω_{ri})	0 rad/s
Minimum linear velocity (v_{min})	0 mm/s
Maximum linear velocity (v_{max})	80 mm/s
Maximum angular velocity (ω_{max})	±0.75 rad/s

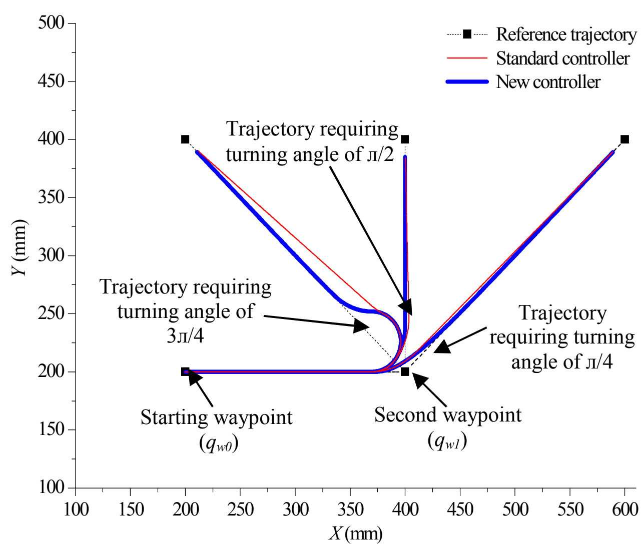

Figure 6 Trajectory tracked with the help of optimized controller

The optimized controller was tested for tracking a complex trajectory requiring sharp turning as well as regular turning to achieve trajectory tracking. Fig. 7 (a) shows the given waypoint trajectory and the trajectory tracked by the robot when using the standard controller as well as while using the new controller. The new controller could able to drive the robot closer to the reference trajectory than the standard controller. Thus the new controller is more suitable when tight maneuvers are required. Fig 7 (b) shows the cross-track error. Fig 7 (c) and (d) respectively shows the linear velocity control signal and angular velocity control signals.

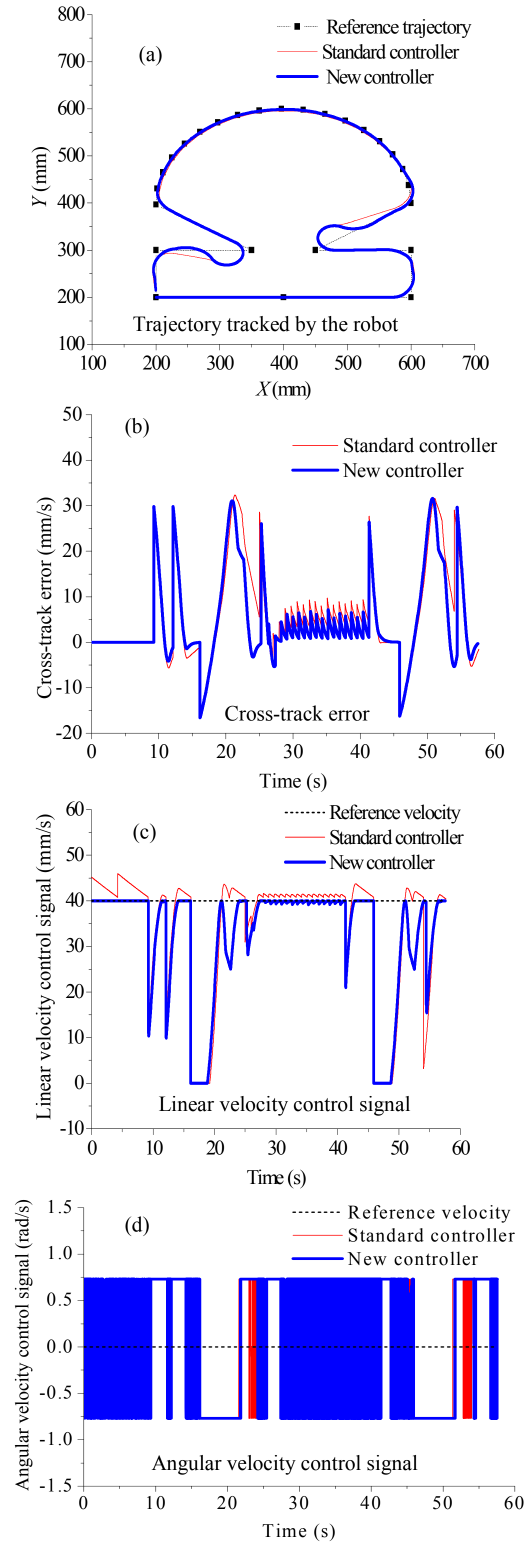

Figure 7 Result of tracking a complex trajectory

Table 3 shows the simulation results. From the table, it can be understood that the new controller could simultaneously reduce both average cross track error as well as average velocity tracking error while tracking different waypoint trajectories having sharp turns as well as regular turn. From the results, it can be concluded that the new controller performs better when compared to the standard controller in minimizing average cross track error and average velocity tracking error when waypoint trajectories requiring sharp turns have to be tracked. Also it can be observed from the circular arc section in Fig. 7(a) that the controller was performing well in a normal trajectory which does not require sharp turns. Hence, the new controller could be an alternative for the standard controller.

TABLE III. EXPERIMENTAL RESULTS

Sl. No	Type of waypoint trajectory	Average of Cross-track error (mm)		Average of velocity tracking error (mm/s)	
		Standard controller	***New controller***	***Standard controller***	***New controller***
1	Trajectory requiring turning angle of л /4	2.3	1.4	3.9	0.4
2	Trajectory requiring turning angle of л /2	3.6	2.8	4.9	2.9
3	Trajectory requiring turning angle of 3л /4	8.3	5.3	11.3	8.9
4	Complex trajectory	8.2	6.6	8.8	7.8

IV. SUMMARY AND CONCLUSIONS

A new control scheme for waypoint navigation of a differential drive robot is proposed. Following are the summary of the simulation result:

- The controller makes the robot move through the line connecting the waypoints.
- The new controller performs better in different test scenarios
- The controller reduces the cross-track error by 19.5 % in case of complex trajectory.
- The velocity tracking error was reduced by 11 %.

From the results, it can be concluded that the new controller performs better in terms of average cross track error and average velocity tracking error when waypoint trajectories requiring sharp turns have to be tracked. The controller was performing well in normal trajectories also. The ability of the controller to make the robot move through the line connecting the waypoints is suitable for tracking the path planner generated waypoint trajectory. So, the need for path smoothening algorithms can be avoided by using the developed controller. Hence this new controller can be an alternative for the standard controller proposed earlier by kanayama et.al. Particle swarm optimization based controller tuning is an interesting area where further investigation is required. In the present scenario, MPSO based optimization was found to be a useful tool for tuning the controller.

REFERENCES

[1] R. Mathew and S. S. Hiremath, “Trajectory Tracking and Control of Differential Drive Robot for Predefined Regular Geometrical Path,” *Procedia Technology*, vol. 25, no. Raerest, pp. 1273–1280, 2016.

[2] Á. Nagy, G. Csorvási, and D. Kiss, “Path planning and control of differential and car-like robots in narrow environments,” *SAMI 2015 - IEEE 13th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, pp. 103–108, 2015.

[3] S. T. Mitrović and Ž. M. Durović, “Fuzzy logic controller for bidirectional garaging of a differential drive mobile robot,” *Advanced Robotics*, vol. 24, no. 8–9, pp. 1291–1311, 2010.

[4] E. Maalouf, M. Saad, and H. Saliah, “A higher level path tracking controller for a four-wheel differentially steered mobile robot,” *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 23–33, 2006.

[5] O. Mohareri, R. Dhaouadi, and A. B. Rad, “Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks,” *Neurocomputing*, vol. 88, pp. 54–66, 2012.

[6] L. Valbuena and H. G. Tanner, “Hybrid potential field based control of differential drive mobile robots,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 68, no. 3–4, pp. 307–322, 2012.

[7] D. Chwa, “Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization,” *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 40, no. 6, pp. 1285–1295, 2010.

[8] B. Anoohya B and R. Padhi, “Trajectory Tracking of Autonomous Mobile Robots Using Nonlinear Dynamic Inversion,” *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 202–207, 2018.

[9] X. Liang, H. Wang, W. Chen, D. Guo, and T. Liu, “Adaptive Image-Based Trajectory Tracking Control of Wheeled Mobile Robots With an Uncalibrated Fixed Camera,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2266–2282, 2015.

[10] S. Blazįç and S. Blažič, “A novel trajectory-tracking control law for wheeled mobile robots,” *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 1001–1007, 2011.

[11] S. Veera Ragavan, S. G. Ponnambalam, and C. Sumero, “Waypoint-based path planner for mobile robot navigation using PSO and GA-AIS,” *IEEE Recent Advances in Intelligent Computational Systems, RAICS 2011*, pp. 756–760, 2011.

[12] J. Vilca, L. Adouane, and Y. Mezouar, “A novel safe and flexible control strategy based on target reaching for the navigation of urban vehicles,” *Robotics and Autonomous Systems*, vol. 70, pp. 215–226, 2015.

[13] M. A. Zakaria *et al.*, “Dynamic Curvature Path Tracking Control for Autonomous Vehicle : Experimental Results,” *3rd IEEE International Conference on Connected Vehicles and Expo (ICCVE 2014)*, pp. 1–6, 2014.

[14] G. M. Jaqueline, G. Liho, V. J. G. Villela, and R. Parkin, “A wheeled mobile robot with obstacle avoidance capability,” *Ingeniería Mecánica*, vol. 1, no. 5, pp. 159–166, 2004.

[15] S. Shair, J. H. Chandler, V. J. González-Villela, R. M. Parkin, and J. M. R. Jackson, “The use of aerial images and GPS for mobile robot waypoint navigation,” *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 6, pp. 692–699, 2008.

[16] R. Mathew and S. S. Hiremath, “Control of Velocity-Constrained Stepper Motor-Driven Hilare Robot for Waypoint Navigation,” *Engineering*, vol. 4, no. 4, pp. 491–499, 2018.

[17] V. D. Nguyen, G. S. Soh, S. Foong, and K. Wood, “De-coupled dynamics control of a spherical rolling robot for waypoint navigation,” *2017 IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2017 and IEEE Conference on Robotics, Automation and Mechatronics, RAM 2017 - Proceedings*, vol. 2018–Janua, pp. 562–567, 2018.

[18] P. Boucher, “Waypoints guidance of differential-drive mobile robots with kinematic and precision constraints,” *Robotica*, vol. 34, no. 4, pp. 876–899, 2016.

[19] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 384–389, 1990.