

Real Time Trajectory Tracking Controller based on Lyapunov Function for Mobile Robot

Mohammed Elsayed

Faculty of Engineering at Shoubra, Benha University, Egypt
108 Shoubra, Cairo Governorate, Egypt

Ashraf Hafez

Faculty of Engineering at Shoubra, Benha University, Egypt
108 Shoubra, Cairo Governorate, Egypt

Abdallah Hammad

Faculty of Engineering at Shoubra, Benha University, Egypt
108 Shoubra, Cairo Governorate, Egypt

Hala Mansour

Faculty of Engineering at Shoubra, Benha University, Egypt
108 Shoubra, Cairo Governorate, Egypt

ABSTRACT

An important issue in robotics research is path tracking control where the robot is required to follow a certain path. The error between the desired path and the actual path is to converge to zero. This problem is more complicated when the robot dynamics is considered. This paper proposes a real time trajectory tracking control for a differential drive wheeled mobile robot (DDWMR) in obstacle-free environment. The robot is guided to follow certain reference path with a pre-calculated velocity profile. The controller design and analysis of the system stability are guaranteed using Lyapunov stability theory. The dynamic model of real DDWMR is derived and used in the LabVIEW simulation environment for testing the validity of designed controller. The obtained simulation results illustrate the success of the proposed controller. Also to Test the effectiveness of proposed controller, a comparison study with a widely used backstepping based controller is performed.

Keywords

Trajectory tracking, nonholonomic robots, dynamic Modeling, differential drive, Lyapunov stability, mobile robot

1. INTRODUCTION

The problem of Real-time trajectory tracking control of a mobile robot is essential in the public arena today. Autonomous mobile robots have different applications in the military field, industry and in all areas of life. The issue of autonomous trajectory tracking control of wheeled mobile robots have occupied a huge part of research interest in the robotics field.

The path tracking problem of any mobile robot is mainly how to generate the required linear and angular velocity that enables the robot to follow a predefined path, i.e., the error between the desired path and the actual path is to converge to zero. In recent years there have been many researches on tracking control of a mobile robot. Mostly include the kinematic model of DDWMR while very few include dynamic model due to the complexity of the model and high non-linearity degree.

Mainly the trajectory tracking control algorithms can be classified

into on of six categories [1]: (1) backstepping [2], [3]; (2) linearization [4]; (3) sliding mode [5]; (4) fuzzy systems [6], [7]; (5) neural networks [8]; and (6) neurofuzzy systems [9].

In this paper, the kinematics and the dynamic model of a differential drive wheeled mobile robot (DDWMR) is presented. This model itself is used as a motion controller in a closed loop control scheme. In the absence of workspace obstacles, the basic motion tasks assigned to the mobile robots may be reduced to moving between two robot postures and following a given trajectory. Also at the end of the paper a brief comparison with the backstepping controller will be introduced.

This paper is organized as follows: in section II the basic structure, kinematic and dynamic modeling of differential drive wheeled mobile robot (DDWMR) are exposed. The proposed control architecture is introduced in Section III. Simulation results are shown in section IV. In section V we will compare our obtained results with backstepping based controller. Finally, we conclude the paper in Section VI.

2. KINEMATIC AND DYNAMIC MODELING OF DDWMR

In this section the kinematic and dynamic models of a real DDWMR are derived. The dynamic model is based on lagrange approach.

2.1 Coordinate System

A top view of the DDWMR used in this work, is shown in Fig. 1. The robot configuration consists of the main frame with two wheels fixed on one axis with two identical motors and a front free wheel (castor). The mobile robot motion is controlled by changing the relative angular speed of the driving wheels.

In order to describe the state of the robot we will define two coordinate systems, the reference frame $\{x_r, y_r\}$ and vehicle frame $\{x_v, y_v\}$. The state of the robot in a reference frame is represented by the position of point b which is located at the center of the wheel axle and the robot heading angle θ . The robot configuration in the

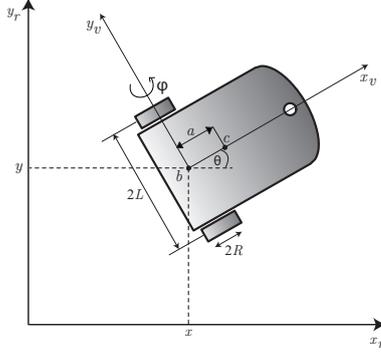


Fig. 1. Differential Drive wheeled Mobile Robot

reference frame is symbolized as

$$p_r = [x \ y \ \theta]^T, \quad (1)$$

and in vehicle frame as

$$p_v = [x_v \ y_v \ \theta_v]^T. \quad (2)$$

The transformation of robot configuration from vehicle frame to reference frame can be done through the following transformation

$$p_r = R(\theta)p_v, \quad (3)$$

where:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

2.2 Nonholonomic Constraints

Some assumptions and constraints must be declared Before proceeding with kinematic and dynamic model of DDWMR

- (1) The robot motion is on a flat surface, meaning zero potential energy,
- (2) No lateral slip movement: This means that the robot center point velocity will be in the direction of the x-axis of vehicle frame and that the motion in the y-axis of vehicle frame will be zero. Since the motion in the y-axis of vehicle frame will be zero, then

$$\dot{y}_v = 0. \quad (5)$$

Using the transformation matrix to represent this constraints in the reference frame, so

$$\begin{bmatrix} \dot{x}_v \\ \dot{y}_v \\ \dot{\theta}_v \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}. \quad (6)$$

So we can represent this constraint in reference frame as following

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0. \quad (7)$$

- (3) The second constraint is pure rolling without slipping, then each wheel touches the ground in single point. The pure rolling constraint is formulated as [10]

$$\dot{x} \cos \theta + \dot{y} \sin \theta + L\dot{\theta} = R\omega_R, \quad (8)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta - L\dot{\theta} = R\omega_L. \quad (9)$$

Equations 7 - 9 can be written in matrix form

$$\begin{bmatrix} -\sin \theta & \cos \theta & 0 & 0 & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = 0, \quad (10)$$

or

$$A(p)\dot{p} = 0, \quad (11)$$

where A_p is the constraints matrix and it will be used in the derivation of the dynamic model.

2.3 Kinematic Model

For the DDWMR, the primary motivation behind kinematic is to obtain the robot speeds as a function of the driving wheels speeds. The linear and angular speeds of the DDWMR in the vehicle frame can be calculated as

$$v = \frac{v_R + v_L}{2} = R \frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2}, \quad (12)$$

and

$$\omega = \frac{v_R - v_L}{2L} = R \frac{(\dot{\varphi}_R - \dot{\varphi}_L)}{2L}. \quad (13)$$

Then the linear and the angular speeds of the DDWMR in the reference frame can be calculated as follows

$$\dot{x} = v \cos \theta = R \frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \cos \theta, \quad (14)$$

$$\dot{y} = v \sin \theta = R \frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \sin \theta, \quad (15)$$

$$\dot{\theta} = \omega = R \frac{(\dot{\varphi}_R - \dot{\varphi}_L)}{2L}. \quad (16)$$

Now consider a DDWMR at certain position (x, y, θ) which has a non-zero distance with the desired position (x_d, y_d, δ) as shown in Fig.2.

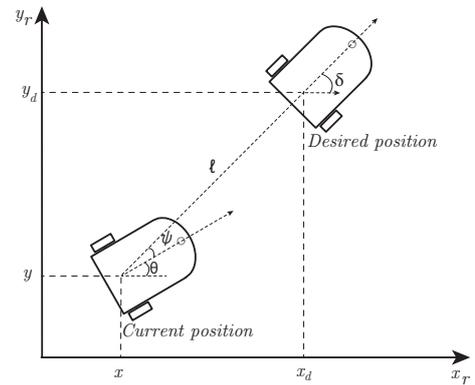


Fig. 2. Current and Goal Positions of robot

The robot can be represented in polar coordinates involving error distance

$$l = \sqrt{(x_d - x)^2 + (y_d - y)^2}, \quad (17)$$

and heading angle error

$$\psi = \delta - \theta, \quad (18)$$

as in the following equation

$$\begin{bmatrix} \dot{\ell} \\ \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} -\cos(\psi) & 0 \\ \sin(\psi)/\ell & -1 \\ \sin(\psi)/\ell & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (19)$$

2.4 Dynamic Modeling

The dynamic model of the DDWMR is very important for simulation of different control algorithms. In this work the lagrangian approach is used to drive the equations of motion of DDWMR. The Lagrangian equation in general is

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{p}_i} \right) + \left(\frac{\partial L}{\partial p_i} \right) = F - A^T(p)\lambda, \quad (20)$$

where the Lagrangian function L is the difference between the kinetic energy and the potential energy of the robot. According to the above assumptions the potential energy equals zero, then the Lagrangian function equals the the kinetic energy only. F is the force vector, A_p is the constraints matrix and λ is the Lagrange multipliers for the robot constraints. The total robot kinetic energy is the summation of robot frame kinetic energy and the wheels kinetic energy. Then

$$L = T = T_f + T_w, \quad (21)$$

where

$$T_f = \frac{1}{2} m_f v_c^2 + \frac{1}{2} I_c \dot{\theta}^2, \quad (22)$$

$$T_w = \frac{1}{2} m_w (v_R^2 + v_L^2) + I_m \dot{\theta}^2 + \frac{1}{2} I_w (\dot{\varphi}_R^2 + \dot{\varphi}_L^2). \quad (23)$$

Where m_f is the robot mass excluding motors and wheels, m_w is the mass of motors and wheels together. I_c is the inertial moment of the robot about the vertical axis passes through the center of mass, I_m and I_w are the inertial moments of the motors and wheels about the vertical axis passes through the center of mass and the wheel axis respectively. Representing the velocities as function of reference coordinates using the relation $v_i^2 = \dot{x}_i^2 + \dot{y}_i^2$, we get

$$v_R^2 + v_L^2 = 2(\dot{x}^2 + \dot{y}^2 + 2(L\dot{\theta})^2), \quad (24)$$

$$v_c^2 = \dot{x}^2 + \dot{y}^2 + 2a\dot{\theta}(\dot{y} \cos \theta - \dot{x} \sin \theta) + (a\dot{\theta})^2, \quad (25)$$

where a is the distance between the center of mass c and mid-point on the axis between the two wheels b . Then

$$L = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2) + m_f a \dot{\theta}(\dot{y} \cos \theta - \dot{x} \sin \theta) + \frac{1}{2} I_w (\dot{\varphi}_R^2 + \dot{\varphi}_L^2) + \frac{1}{2} I \dot{\theta}^2, \quad (26)$$

where

$$m = m_f + 2m_w, \quad (27)$$

$$I = I_c + m_f a^2 + 2m_w L^2 + 2I_m. \quad (28)$$

Applying the lagrangian equation 20 on the obtained lagrangian function, easily we can see that

$$m\ddot{x} - m_f a(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) = C_1, \quad (29)$$

$$m\ddot{y} + m_f a(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = C_2, \quad (30)$$

$$I\ddot{\theta} + m_f a(\ddot{y} \cos \theta - \ddot{x} \sin \theta) = C_3, \quad (31)$$

$$I_w \ddot{\varphi}_R = \tau_R + C_4, \quad (32)$$

$$I_w \ddot{\varphi}_L = \tau_L + C_5. \quad (33)$$

These equations can be written in form of the general equations of motion [11]

$$M(p)\ddot{p} + V(p, \dot{p})\dot{p} = B(p)\tau - A^T(p)\lambda, \quad (34)$$

where $M(p)$ is the symmetric positive definite inertia matrix, $V(p, \dot{p})$ is the centripetal and coriolis matrix, $B(p)$ is the input matrix, $\tau^T = [\tau_R \ \tau_L]$ is the input vector, $A^T(p)$ is the matrix associated with the kinematic constraints, and λ is the Lagrange multipliers vector. We can see that

$$M(p) = \begin{bmatrix} m & 0 & -m_f a \sin \theta & 0 & 0 \\ 0 & m & m_f a \cos \theta & 0 & 0 \\ -m_f a \sin \theta & m_f a \cos \theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}, \quad (35)$$

$$V(p, \dot{p}) = \begin{bmatrix} 0 & 0 & -m_f a \dot{\theta} \cos \theta & 0 & 0 \\ 0 & 0 & -m_f a \dot{\theta} \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (36)$$

$$B(p) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (37)$$

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix}. \quad (38)$$

Back to the kinematic model, the model can be written in the following form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R \cos \theta & R \cos \theta \\ R \sin \theta & R \sin \theta \\ R/L & -R/L \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix}, \quad (39)$$

or

$$\dot{p} = Q(p)\eta. \quad (40)$$

It is Straightforward to show that $Q^T(p)A^T(p) = 0$. Differentiating Equation 40, substituting the expression for \dot{p} and \ddot{p} into 34, and premultiplying by $Q^T(p)$ to convert the motion equation to the unconstrained form, then we have

$$\bar{M}(p)\dot{\eta} + \bar{V}(p, \dot{p})\eta = \bar{B}(p)\tau, \quad (41)$$

where:

$$\bar{M}(p) = Q^T(p)M(p)Q(p) \quad (42)$$

$$= \begin{bmatrix} I_w + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_w + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix},$$

$$\bar{V}(p, \dot{p}) = Q^T(p)M(p)\dot{Q}(p) + Q^T(p)V(p, \dot{p})Q(p) \quad (43)$$

$$= \begin{bmatrix} 0 & \frac{R^2}{2L}m_f a \dot{\theta} \\ -\frac{R^2}{2L}m_f a \dot{\theta} & 0 \end{bmatrix},$$

$$\bar{B}(p) = Q^T(p)B(p) \quad (44)$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Or we can write the equation in terms of v and ω as in the following form

$$(m + \frac{2}{R^2}I_w)\dot{v} - am_f\omega^2 = \frac{(\tau_R + \tau_L)}{R}, \quad (45)$$

$$(I + \frac{2L^2}{R^2}I_w)\dot{\omega} + am_f v\omega = \frac{L(\tau_R - \tau_L)}{R}. \quad (46)$$

Equations 45 and 46 are used to simulate our real robot whose dynamic parameters are listed in Table 1.

Table 1. Robot dynamic model parameters

Parameter	Value (unit)	Description
L	10 cm	The distance between the drive wheel and the axis of symmetry.
R	7.5 cm	Wheel radius.
a	5 cm	The distance between the center of mass and drive wheel axis.
m	6 kg	Total robot weigh with onboard load.
m_f	4 kg	Robot frame weigh.
I	5 kg.m ²	Mass moment of inertia about the center of mass c .
I_w	0.005 kg.m ²	Inertial moment of the motors and wheels about the wheel axis.

3. TRACKING PROBLEM AND CONTROL ALGORITHM

In this section the control algorithm that we will use in this paper is derived. The role of the control algorithm is to enable the robot to go to the desired goal and follow the target trajectory. That is achieved by using the posture error to provide the robot with the reference linear and angular velocities. The description of tracking problem can be described as follows. Given the desired robot posture $p_d = [x_d \ y_d \ \delta]^T$ while the robot currently is at $p = [x \ y \ \theta]^T$ and the controller must generate the linear and the angular velocities such that $\lim_{t \rightarrow \infty} p = p_d$. A general structure for the tracking control system is presented in Fig. 3.

To grantee the convergence and the stability of our controller, a Lyapunov function candidate is chosen as

$$V = V_1 + V_2 = \frac{\ell^2}{2} + \frac{\psi^2}{2} \quad (47)$$

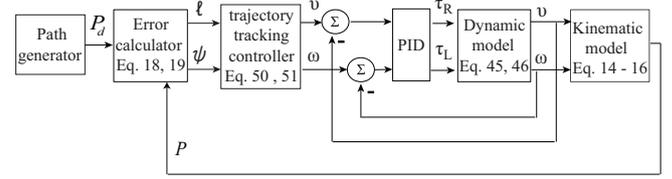


Fig. 3. Tracking controller system architecture.

It is clear that $V > 0$ and that $V = 0$ only if there is no error in robot posture, then V is appositive definite function. By taking the time derivative of Lyapunov function, we have

$$\dot{V} = \ell \dot{\ell} + \psi \dot{\psi} \quad (48)$$

If we substitute for $\dot{\ell}$ and $\dot{\psi}$ from model equation (19)

$$\dot{V} = \ell(-v \cos(\psi)) + \psi(\frac{v \sin(\psi)}{\ell} - \omega) \quad (49)$$

Then we can choose the driving velocities to be

$$v = k_\ell \ell \cos(\psi) \quad (50)$$

$$\omega = k_\ell \sin(\psi) \cos(\psi) + k_\psi \tanh(k_\psi \psi) \quad (51)$$

Where k_ℓ and k_ψ are positive numbers, then the derivative of the Lyapunov function will be in the form of

$$\dot{V} = -k_\ell \ell^2 \cos^2(\psi) - k_\psi \psi \tanh(k_\psi \psi) \quad (52)$$

It is clear that the derivative of the Lyapunov function V is a negative function, then the point $e_P = 0$ is asymptotically stable.

4. SIMULATION RESULTS

In this section the performance of proposed controller is tested using NI-LabVIEW control design and simulation toolkit. In the following subsections we will show some of the results that prove the stability and convergence of used controller. The controller parameters are chosen as: $k_\ell = 2$, $k_\psi = 3$. The sampling time is set to 0.001 second.

4.1 Go to Goal

The desired goal for the robot is described as $x_d = 3$ and $y_d = 4$. Goal point is sent to controller which directs the vehicle towards the goal. The robot starts at $(-1, 0, 0)$. Then the error at $t = 0$ is $e_P = [4 \ 4 \ \pi/4]^T$. Fig. 4 shows the result of go to goal simulation. Fig. 4.A shows the tracking performance of the robot in xy -plan. Fig. 4.B shows the tracking errors in x , y directions and in orientation. Fig. 4.C shows the controller output (linear and angular velocities).

4.2 Path Tracking

4.2.1 Circular Path Tracking. The desired goal for the robot is to follow the path described as $x_d = 1 + \sin t$ and $y_d = 1 - \cos t$, or in general $(x_d - 1)^2 + (y_d - 1)^2 = 1$. The robot starts at $(-1, -1, 0)$ with initial error $e_P = [2 \ 1 \ 0]^T$. Fig. 5 shows the result of circular path tracking simulation. Fig. 5.A shows the tracking performance of the robot in xy -plan. Fig. 5.B shows the controller output (linear and angular velocities). Fig. 5.C shows the desired and the actual values of x . Fig. 5.D shows the desired and the actual values of y .

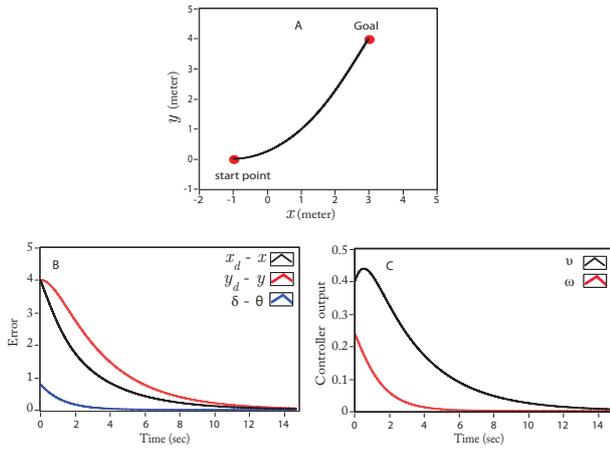


Fig. 4. Go to goal. A: The tracking performance; B: The tracking errors in x and y directions, and in orientation; C: The linear and angular velocities.

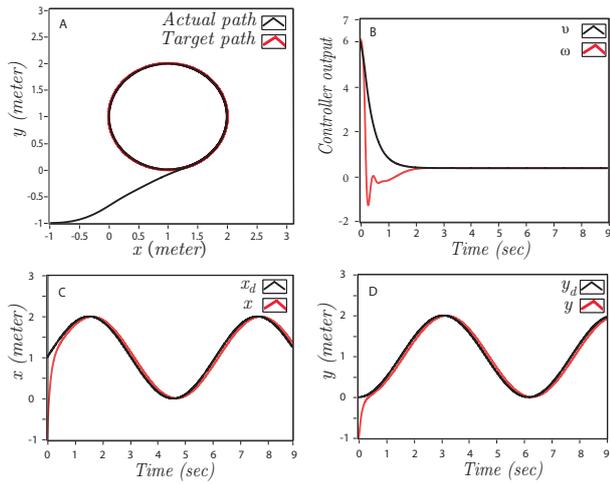


Fig. 5. Simulation tracking for circular path. A: The tracking performance; B: The linear and angular velocities; C, D: The desired and the actual values of x and y respectively.

4.2.2 Figure-eight Path Tracking. The desired goal for the robot is to follow the path described as $x_d = 1 + \sin 2t$ and $y_d = 1 - \cos t$. The robot starts at $(-1, -1, 0)$ with initial error $e_P = [2 \ 1 \ 0]^T$. Fig. 6 shows the result of Figure-eight path tracking simulation. Fig. 6.A shows the tracking performance of the robot in xy -plan. Fig. 6.B shows the controller output (linear and angular velocities). Fig. 6.C shows the desired and the actual values of x . Fig. 6.D shows the desired and the actual values of y .

4.2.3 Ellipsoidal Path Tracking. The desired goal for the robot is to follow the path described as $x_d = 1 + \sin t$ and $y_d = 1 - \cos(t + \pi/4)$, Where $f = 2\pi Hz$. The robot starts at $(-1, -1, 0)$ with initial error $e_P = [2 \ 1.29 \ 0.576]^T$. Fig. 7 shows the result of ellipsoidal path tracking simulation. Fig. 7.A shows the tracking performance of the robot in xy -plan. Fig. 7.B shows the controller output (linear and angular velocities). Fig. 7.C shows the desired and the actual values of x . Fig. 7.D shows the desired and the actual values of y .

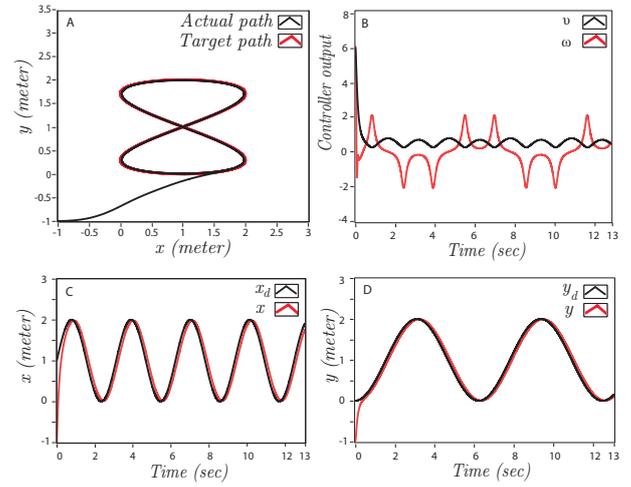


Fig. 6. Simulation tracking for Figure eight path. A: The tracking performance; B: The linear and angular velocities; C, D: The desired and the actual values of x and y respectively.

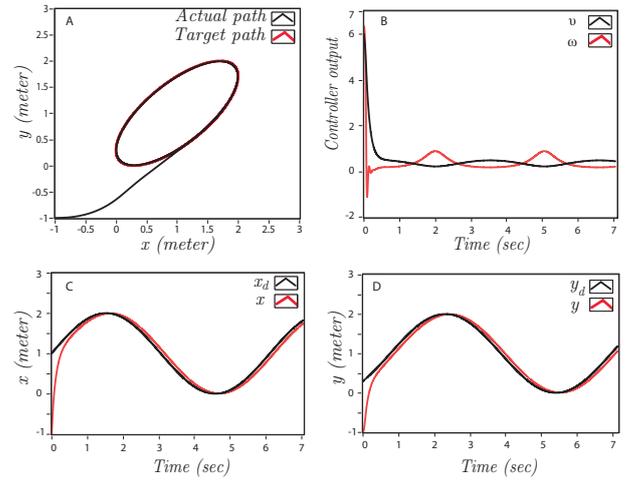


Fig. 7. Simulation tracking for ellipsoidal path. A: The tracking performance; B: The linear and angular velocities; C, D: The desired and the actual values of x and y respectively.

5. COMPARISON TO BACKSTEPPING BASED CONTROLLER

In this section, the backstepping algorithm based control is compared to our Lyapunov based tracking controller proposed in this work. The backstepping based tracking control technique used infig:Circle-sim [12] is given by the following equations

$$v = v_d \cos e_\theta + k_x e_x, \quad (53)$$

and

$$\omega = \omega_d + k_y v_d e_y + k_\theta v_d \sin e_\theta, \quad (54)$$

where k_x , k_y , and k_θ are the controller parameters, and

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix}. \quad (55)$$

The backstepping tracking control defined in equation (53) and (54) is simulated here under the same conditions as the proposed controller. The Fig. 8 Shows the tracking performance of both controller for a circular path. As we can see that the proposed controller is more faster in reaching the reference trajectory than the backstepping based controller.

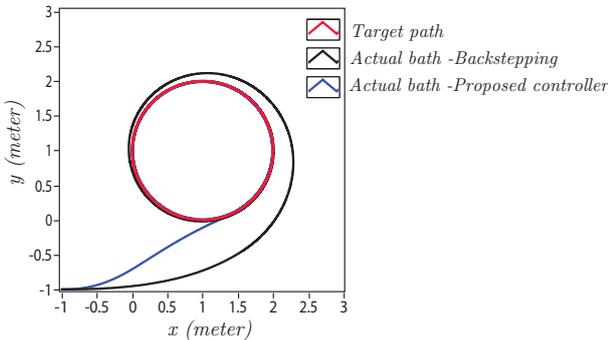


Fig. 8. Tracking performance comparison between backstepping controller and proposed controller.

Also, we can see in Fig. 9 the large difference between the startup linear and angular velocities in both controller. The proposed controller velocity profile is more practical than that produced from back stepping controller.

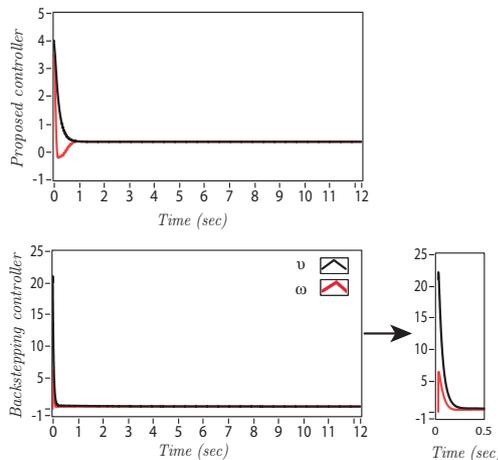


Fig. 9. Controller output for backstepping controller and proposed controller.

6. CONCLUSION

In this paper, the kinematic and dynamic models of DDWMR are introduced. Non-linear dynamics based tracking controller is suggested, which is capable of producing continuous and smooth velocity commands. The tracking controller simulation shows that the system is asymptotically stable, and that the tracking errors are ensured to move toward to zeros. Also we compared the results obtained using the proposed controller to the result of backstepping based controller.

7. REFERENCES

- [1] G. Yuan, S. Yang, and G. Mittal, Tracking control of a mobile robot using a neural dynamics based approach, Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164).
- [2] T. Dierks and S. Jagannathan, Control of Nonholonomic Mobile Robot Formations: Backstepping Kinematics into Dynamics, 2007 IEEE International Conference on Control Applications, 2007.
- [3] I. Benaoumeur, B. Laredj, H. E. A. Reda, and A.-F. Zoubir, Backstepping Approach for Autonomous Mobile Robot Trajectory Tracking, Indonesian Journal of Electrical Engineering and Computer Science, vol. 2, no. 3, p. 478, Jan. 2016.
- [4] D.-H. Kim and J.-H. Oh, Tracking control of a two-wheeled mobile robot using inputoutput linearization, Control Engineering Practice, vol. 7, no. 3, pp. 369373, 1999.
- [5] J.-M. Yang and J.-H. Kim, Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots, IEEE Transactions on Robotics and Automation, vol. 15, no. 3, pp. 578587, 1999.
- [6] A. Ollero, A. G. Cerezo, and J. V. Martinez, Fuzzy supervisory path tracking of mobile robots, Control Engineering Practice, vol. 2, no. 1, p. 160, 1994.
- [7] A. Pandey and D. R. Parhi, Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm, Defence Technology, vol. 13, no. 1, pp. 4758, 2017.
- [8] J. Velagic, N. Osmic, and B. Lacevic, Design of Neural Network Mobile Robot Motion Controller, New Trends in Technologies, Jan. 2010.
- [9] M. K. Singh and D. R. Parhi, Intelligent neuro-controller for navigation of mobile robot, Proceedings of the International Conference on Advances in Computing, Communication and Control - ICAC3 09, 2009. Vol. 4 No. 1 2012 ISSN: 0975-3176 pp. 19-26
- [10] Y. Tian and N. Sarkar, Control of a Mobile Robot Subject to Wheel Slip, Journal of Intelligent and Robotic Systems, vol. 74, no. 3-4, pp. 915929, Sep. 2013.
- [11] N. Sarkar, X. Yun, and V. Kumar, Control of Mechanical Systems With Rolling Constraints, The International Journal of Robotics Research, vol. 13, no. 1, pp. 5569, 1994.
- [12] R. Fierro and F. Lewis, Control of a nonholonomic mobile robot: backstepping kinematics into dynamics, Proceedings of 1995 34th IEEE Conference on Decision and Control.