

The Impact of the Dynamic Model in Feedback Linearization Trajectory Tracking of a Mobile Robot

Welid Benchouche^{1*}, Rabah Mellah¹, Mohammed Salah Bennouna²

¹ Design and Drive of Production Systems Laboratory, Faculty of Electrical and Computing Engineering, University Mouloud Mammri of Tizi-Ouzou, P. O. B. 17 RP, 15000 Tizi-Ouzou, Algeria

² Department of Mechanical Engineering, University Kasdi Merbah Ouargla, P. O. B. 511, 30000 Ouargla, Algeria

* Corresponding author, e-mail: welidbenchouche@gmail.com

Received: 01 September 2020, Accepted: 14 October 2020, Published online: 03 September 2021

Abstract

This paper proposes the impact of the Dynamic model in Input-Output State Feedback Linearization (IO-SFL) technique for trajectory tracking of differential drive mobile robots, which has been restricted to using just the kinematics in most of the previous approaches. To simplify the control problem, this paper develops a novel control approach based on the velocity and position control strategy. To improve the results, the dynamics are taken into account. The objective of this paper is to illustrate the flaws unseen when adopting the kinematics-only controllers because the nonlinear kinematic model will suffice for control design only when the inner velocity (dynamic) loop is faster than the slower outer control loop. This is a big concern when using kinematic controllers to robots that don't have a low-level controller, Arduino robots for example. The control approach is verified using the Lyapunov stability analysis. MATLAB/SIMULINK is carried out to determine the impact of the proposed controller for the trajectory tracking problem, from the simulation, it was discovered that the proposed controller has an excellent dynamic characteristic, simple, rapid response, stable capability for trajectory-tracking, and ignorable tracking error. A comparison between the presence and absence of the dynamic model shows the error in tracking due to dynamic system that must be taken into account if our system doesn't come with a built-in one, thus, confirming the superiority of the proposed approach in terms of precision, with a neglectable difference in computations.

Keywords

mobile robots, dynamic modeling, trajectory tracking, lyapunov asymptotic stability, IO-SFL

1 Introduction

Wheeled Mobile Robots (WMRs) are becoming a necessary part of our daily lives because of the technology improvement, at first, they were limited to research purposes or military missions because of their high cost, nowadays; they are very affordable and available to general consumers. WMRs are the most important type of mobile robots, they are reliable, and easily built, they can move on almost every surface, many industrial and household applications employ them, for component transportation, painting, welding, inspecting, house cleaning, and many other applications. Due to all this, research on WMRs has been an immensely active field [1]. According to [2, 3], non-holonomic systems cannot be stabilized by continuously differentiable, time-invariant state feedback control law. This indicates that this problem is truly nonlinear; therefore, innovative techniques are highly needed.

Some robots have a built-in low lever controller (actuator controller), such as Khepera mobile robot, or a DrRobot I90 mobile robot, but they cost a lot, on the other hand, some Arduino or Raspberry robots for example don't have such a controller.

In this paper, we work on the non-holonomic class mobile robot that uses a differential drive, our objective is to develop a good dynamic model that eliminates kinematic constraints from the model, remove the maximum amount of nonlinearities, and represent the system in a symmetric way (number of inputs = number of outputs), after achieving that, our second objective is to control the resulting system using a kinematic controller that is asymptotically stable, and a dynamic controller based on feedback linearization, our last objective is to illustrate the impact of the dynamic control over the

kinematic-only control and we hope to achieve perfect tracking and neglectable errors.

We assume that wheel-ground contact is high to achieve pure rolling and no slipping. There are basically 3 varieties of motion command problems:

- path following,
- trajectory tracking,
- and point stabilization.

Point stabilization refers back to the stabilization of the robot into predefined Cartesian coordinates and orientation. Path following refers to moving a robotic system in a route in a time-independent manner. The trajectory tracking problem is similar to the path following's but in a predefined time. A regular motion control problem is trajectory-tracking, that's concerned with the design of control law that forces a mobile robot to reach and stay on, a time parameterized reference (i.e., a geometric course with an associated timing regulation) [4].

Trajectory tracking of mobile robots is one of the fundamental challenging problems in WMR motion control [5]. Because of the non-autonomous nature that comes from the time-varying reference [6]. However, the asymptotic stabilization has been proven to be obtained using the discontinuous, time-varying, or hybrid command laws [7–9], lots of research has been devoted to solving this problem using different techniques. Some of the researchers focused only on the kinematic model to design controllers [3–5, 10–24]. Sharma et al.'s work [4] is a comparison between two kinematic models for model predictive control, first one was Successive linear model and second one was an error based linear kinematic model, Kanayama et al. [5] used feedback linearization controller to control position; Nascimento et al. [10] used a new non-linear model predictive control approach and achieved better results with regards to the classical control approaches in which he assumed that each position and velocity given by the trajectory generator is the state of a virtual target to be tracked; Al Khatib et al. [11] show a comparison of a feedback linearization controller (Input-Output State Feedback Linearization: IO-SFL), Adaptive Proportional Controller (APC) and a Nussbaum function based Adaptive controller (NA); Merabti et al.'s work [12] is a comparison of metaheuristics (particle swarm optimization, ant colony optimization, and gravitational search algorithms) to solve an MPC problem; [13–15] used sliding mode control; [16–20] used adaptive control to eliminate the control

parameter problem; [20–24] used model predictive control techniques and had great success. However, as mentioned in [25], in mobile robotics, it is a good practice to include the dynamics of the system when designing trajectory tracking controllers to achieve stable motion control laws, as in [17, 26–39]. Martins et al. [26] proposed an adaptive controller with a parameterized dynamic model that is tuned online and showed good results; in [17, 27–35] were used adaptive control to compensate for the dynamics and control parameters; Tawfik et al. [36] used a Fractional-order (FOPID) that is tuned using an evolutionary algorithm to control the error position which feeds torques to the dynamic model and showed robustness to disturbances and uncertainties in the system model; Fierro and Lewis [37] used backstepping and fuzzy logic and designed a hybrid controller that satisfied a good position tracking performance; Dumitrascu et al. [38] used a backstepping controller and considered actuator (wheel) dynamics; Cherrotin and Boumehraz [39] used fuzzy logic and reinforcement learning-based approaches for the WMR navigation in an unknown environment.

The adaptive model predictive controller and the optimal model predictive controller seem to be very promising for this control problem, because the reference trajectory is known beforehand, and because it is an online optimization technique, also, its ability to handle constraints is easy and straightforward, they show amazing results and fewer errors. However, they suffer from the computational burden, especially when the system is fast evolving, and/or highly nonlinear, which hinders the successful application of them [40–43].

This paper addresses the impact of the dynamic modeling for the trajectory tracking problem which forces a differential drive WMRs considering both kinematics and dynamics to track a desired predefined trajectory. This means two controllers, a kinematic controller to make the position error asymptotically stable using a Lyapunov function. Then, designing a dynamic controller to compute the torque such that the velocities of the mobile robot converge to the given velocity inputs from the kinematic controller. In other words, we divide the controller design into two stages, the first one, the kinematic controller to control the position, and the dynamics controller to control the speed. Finally, we show a comparative study between the presence and absence of the dynamic model and the controller confirms the superiority of the proposed approach in terms of precision.

The rest of this paper is organized as follows. The kinematics and dynamics modeling for the wheeled mobile robot that is used in this paper are outlined in Section 2. After that, the design of the trajectory tracking controller for our system is illustrated in Section 3. Results for two reference trajectories (rectangular and a lemniscate), and a comparative study to show the clear difference between using only kinematics versus using kinematics and dynamics are depicted in Section 4. Finally, Section 5 presents a brief conclusion.

2 Mathematical modeling of a differential drive Wheeled Mobile Robot (WMR)

2.1 Kinematic modeling of the differential drive Wheeled Mobile Robot (WMR)

The derivations of the WMR's kinematics utilized in this paper are based on the assumptions [44] below:

1. Design assumptions:
 - The WMR does not contain flexible parts; it is considered a rigid body robot.
 - There is no steering link per wheel; wheels can either go forward or backward only.
2. Operational assumptions:
 - The WMR moves on a planar surface.
 - The translational friction at the point of contact between a wheel and the surface is large enough so that no translational slip may occur.

To start modeling the WMR, we provide some background information from [45]. In order to illustrate the position of the robot on the plane, we establish a relationship between the global reference frame on the plane and the local reference frame on the robot. These frames are shown in Fig. 1. The axes X_I and Y_I define the inertial global reference frame. The origin is O . To specify the position of the WMR, we choose a point P to represent the position reference point of the robot chassis; the pair $[X_R Y_R]^T$ represents

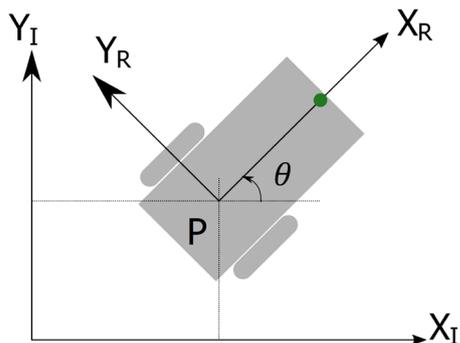


Fig. 1 The global reference frame and the robot local reference frame

the robot reference frame point, also referred to as the local frame point, this pair pass-through the point P and defines the WMR's local reference frame. The position of P in the global reference frame is specified by coordinates x and y , and the angular difference between the global and local reference frames (i.e. the angle between X_I and X_R) is given by θ . The position of the robot is then completely specified by the three variables x, y, θ . We define a 3 by 1 vector q describing the robot posture where q_I represents the global frame and q_R represents the local frame, we also define the orthogonal rotation matrix $R(\theta)$ as follows:

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

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

So $R(\theta)$ defines the relation between \dot{q}_I and, \dot{q}_R in Eq. (3):

$$\dot{q}_I = [\dot{x}, \dot{y}, \dot{\theta}]^T = f(l, r, \theta, \dot{\phi}_r, \dot{\phi}_l). \quad (3)$$

The differential drive robot has two wheels, each with a diameter r . Given a point P centered between two drive wheels, each wheel is at a distance l from P . Given r, l, θ and the spinning speed of each wheel $\dot{\phi}_r$ and $\dot{\phi}_l$, a forward kinematic model for the robot's overall speed in the global reference frame is:

$$\dot{q}_I = [\dot{x}, \dot{y}, \dot{\theta}]^T = f(l, r, \theta, \dot{\phi}_r, \dot{\phi}_l). \quad (4)$$

The speed of each wheel in the robot frame is $r\dot{\phi}$, therefore the translation speed in the robot frame is the average velocity:

$$v = \frac{r\dot{\phi}_r}{2} + \frac{r\dot{\phi}_l}{2} \quad (5)$$

$$w = \frac{r\dot{\phi}_r}{2l} - \frac{r\dot{\phi}_l}{2l}. \quad (6)$$

Before we present the derivation of the kinematic model of the WMR, two constraints will be presented for every fixed wheel. Fig. 2 shows a wheel A and describes its pose with respect to the local reference frame. Where α and β determine the orientation of the wheel with respect to the local frame. The first constraint enforces the concept of rolling contact i.e., that the wheel must roll when motion takes place in the appropriate direction, represented by Eq. (7). The second constraint enforces the concept of no lateral slippage (i.e., that the wheel must not slide orthogonal to the wheel plane), represented by Eq. (8).

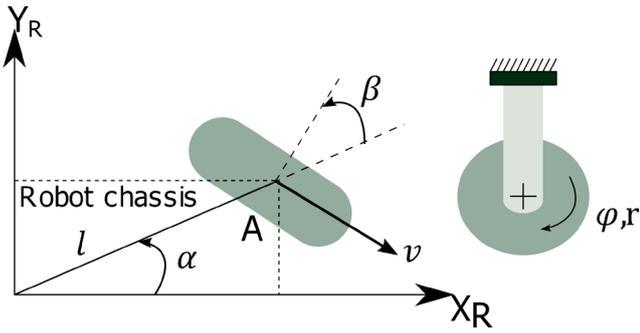


Fig. 2 A fixed standard wheel and its parameters [19]

$$\left[\left[\sin(\alpha + \beta) - \cos(\alpha + \beta)(-l) \cos \beta \right] R(\theta) \right] \dot{\mathbf{q}}_T - r \dot{\phi}_r = 0 \quad (7)$$

$$\left[\left[\cos(\alpha + \beta) \quad -\sin(\alpha + \beta) \quad l \sin \beta \right] R(\theta) \right] \dot{\mathbf{q}}_T = 0 \quad (8)$$

These constraints can be written in Eqs. (9), (10) [22]:

$$A(\mathbf{q}) \dot{\mathbf{q}} = 0 \quad (9)$$

$$A(\mathbf{q}) = \begin{bmatrix} -\sin \theta & \cos \theta & a & 0 & 0 \\ \cos \theta & \sin \theta & L & -R_a & 0 \\ \cos \theta & \sin \theta & -L & 0 & R_a \end{bmatrix} \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} \quad (10)$$

The above representation of the non-holonomic constraint is useful when we want to take the constraints into account in the dynamic modeling.

The combination of the wheel rolling and sliding constraints for all wheels of the robot describes the kinematic behavior and the resulting equation is the Eq. (11) [46]:

$$\dot{\mathbf{q}}_T = R(\theta)^{-1} \begin{bmatrix} \frac{r \dot{\phi}_r + r \dot{\phi}_l}{2} + \frac{r \dot{\phi}_l}{2} \\ 0 \\ \frac{r \dot{\phi}_r - r \dot{\phi}_l}{2l} - \frac{r \dot{\phi}_l}{2l} \end{bmatrix} \quad (11)$$

The linear velocity v of the robot is always heading in the X_R direction of the robot's reference frame due to the nonholonomic constraint and the angular velocity is w can be considered as the rotation speed of the local reference \mathbf{q}_R frame with respect to the global reference frame \mathbf{q}_T , by writing Eq. (11) in terms of v and w :

$$\dot{\mathbf{q}}_T = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{r \dot{\phi}_r + r \dot{\phi}_l}{2} + \frac{r \dot{\phi}_l}{2} \\ 0 \\ \frac{r \dot{\phi}_r - r \dot{\phi}_l}{2l} - \frac{r \dot{\phi}_l}{2l} \end{bmatrix} = \begin{bmatrix} v \cos \theta \\ v \sin \theta \\ w \end{bmatrix} \quad (12)$$

2.2 Dynamic modeling of the WMR

The dynamics of a robot is essential in designing a controller for the robot. By modeling forces that include energies and velocities associated with the motions. It is very important to study the dynamics because the kinematics study is geometrical, hence, doesn't consider forces affecting the robot.

In order to design a dynamic model for our robot, we provide some background information from [46].

A non-holonomic mobile robot with n -dimensional configuration space L with generalized coordinates (q_1, q_2, \dots, q_n) and subject to m constraints may be described by Eq. (13):

$$M(\mathbf{q}) \ddot{\mathbf{q}} + V(\mathbf{q}, \dot{\mathbf{q}}) + F(\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = B(\mathbf{q}) \boldsymbol{\tau} - A^T(\mathbf{q}) \boldsymbol{\lambda}, \quad (13)$$

where:

$M(\mathbf{q})$ is a symmetric positive definite inertia matrix,

$V(\mathbf{q}, \dot{\mathbf{q}})$ is the centripetal and Coriolis matrix,

$F(\dot{\mathbf{q}})$ is the surface friction matrix,

$\mathbf{G}(\mathbf{q})$ is the gravitational vector,

$B(\mathbf{q})$ is the input transformation matrix,

$\boldsymbol{\tau}$ is the input vector,

$A^T(\mathbf{q})$ is the kinematic constraint matrix,

$\boldsymbol{\lambda}$ is the Lagrange multipliers vector of forces.

To derive the dynamic model, we use the Euler-Lagrange approach that is:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}_i} \right) - \frac{\partial L}{\partial \mathbf{q}_i} = \sum_{j=1}^n \lambda_j a_{ji} + Q_i \quad i=1, 2, \dots, n, \quad (14)$$

where:

q_1, q_2, \dots, q_n are the generalized coordinates,

$L = T - V$ is the Lagrangian which is the difference between the systems kinetic energy T and potential energy V .

λ_j is the Lagrangian multiplier which relates the constraints to the constraint forces.

Q_i is the nonconservative forces in the system.

First, we calculate the kinetic and potential energies. Because the robot's movement is restricted to the ground, the potential energy is zero $V = 0$, therefore $L = T$. To calculate the kinetic energy, we introduce a new point in the robot c , and the kinetic energy function can be derived according to the velocities shown in Fig. 3.

To find the velocity of the point P and C :

$$x_c = x_p + a \cos \theta \quad (15)$$

$$y_c = y_p + a \sin \theta, \quad (16)$$

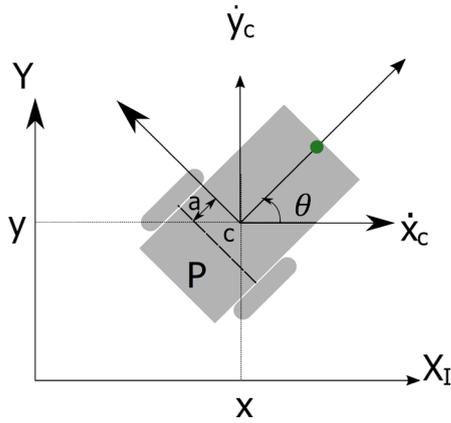


Fig. 3 The velocities of the robot

therefore

$$\dot{x}_c = \dot{x}_p - a\dot{\theta} \sin \theta \quad (17)$$

$$\dot{y}_c = \dot{y}_p + a\dot{\theta} \cos \theta. \quad (18)$$

By rearranging Eq. (17) and Eq. (18), the velocity of the robot center of rotation P is:

$$v_p = (\dot{x}_c + a\dot{\theta} \sin \theta)\vec{i} + (\dot{y}_c - a\dot{\theta} \cos \theta)\vec{j}. \quad (19)$$

The kinetic energy is:

$$T = \frac{1}{2}(m v_p^2 + I_p^2). \quad (20)$$

And from the parallel axis theorem we can say that:

$$I_p = I_c + ma^2. \quad (21)$$

By substituting Eq. (19) and Eq. (21) into Eq. (20), the kinetic energy becomes:

$$T = \frac{1}{2}m\dot{x}_c^2 + \frac{1}{2}m\dot{y}_c^2 + \frac{1}{2}I\dot{\theta}^2 + ma^2\dot{\theta}^2 + ma\dot{x}_c\dot{\theta} \sin \theta - ma\dot{y}_c\dot{\theta} \cos \theta. \quad (22)$$

From the kinetic energy T and knowing the potential energy V is zero, the Lagrangian will become

$$L = \frac{1}{2}m\dot{x}_c^2 + \frac{1}{2}m\dot{y}_c^2 + \frac{1}{2}I\dot{\theta}^2 + ma^2\dot{\theta}^2 + ma\dot{x}_c\dot{\theta} \sin \theta - ma\dot{y}_c\dot{\theta} \cos \theta. \quad (23)$$

The generalized coordinates to use in the Lagrange formulation are as follows:

$$\mathbf{q} = [x_c, y_c, \theta]^T \quad (24)$$

$$\dot{\mathbf{q}} = [\dot{x}_c, \dot{y}_c, \dot{\theta}]^T. \quad (25)$$

In order to find the dynamic equations using the above generalized coordinates Eq. (24) and Eq. (25) and the Lagrangian from Eq. (23), and then substituting in the Lagrange equation Eq. (14), we have:

$$m\ddot{x}_c + ma\ddot{\theta} \sin \theta + ma\dot{\theta}^2 \cos \theta = F_x + C_x \quad (26)$$

$$m\ddot{y}_c - ma\ddot{\theta} \cos \theta + ma\dot{\theta}^2 \sin \theta = F_y + C_y \quad (27)$$

$$(I_c + 2ma^2)\ddot{\theta} + ma\ddot{x}_c \sin \theta - ma\dot{y}_c \cos \theta = \tau + C_\theta, \quad (28)$$

where:

F_x, F_y are the actuator forces in the x -direction and y -direction respectively.

τ is the actuator rotational torque on the robot.

C_x, C_y and C_θ are the constraint forces in the x, y and θ direction.

F_x, F_y and τ are obtained and presented in Eq. (29):

$$\begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}. \quad (29)$$

According to the constraint equation shown in Eq. (10), the constraint forces will become:

$$\begin{bmatrix} C_x \\ C_y \\ C_\theta \end{bmatrix} = \begin{bmatrix} m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \sin \theta \\ -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \cos \theta \\ ma(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \end{bmatrix} = A^T(\mathbf{q})\lambda. \quad (30)$$

Now, Eqs. (26)–(28) can be presented in matrix form as follows:

$$\begin{bmatrix} m & 0 & ma \sin \theta \\ 0 & m & -ma \cos \theta \\ ma \sin \theta & -ma \cos \theta & I_c + 2ma^2 \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} ma\dot{\theta}^2 \cos \theta \\ ma\dot{\theta}^2 \sin \theta \\ 0 \end{bmatrix} \dot{\mathbf{q}} = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (31)$$

$$+ \begin{bmatrix} m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \sin \theta \\ -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \cos \theta \\ ma(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta} \end{bmatrix}.$$

Comparing the Eq. (31) with the general form for a robot dynamic equation Eq. (13), we have the following parameters for the differential drive mobile robot system.

$$M(\mathbf{q}) = \begin{bmatrix} m & 0 & ma \sin \theta \\ 0 & m & -ma \cos \theta \\ ma \sin \theta & -ma \cos \theta & I_c + 2ma^2 \end{bmatrix}$$

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} m a \dot{\theta}^2 \cos \theta \\ m a \dot{\theta}^2 \sin \theta \\ 0 \end{bmatrix}$$

$F(\dot{\mathbf{q}}) = 0$ assumed to be zero in this paper.

$\mathbf{G}(\mathbf{q}) = 0$ assumed to be zero in this paper.

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$B(\mathbf{q}) = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix}$$

$$A^T(\mathbf{q}) = \begin{bmatrix} -\sin \theta \\ +\cos \theta \\ -a \end{bmatrix}$$

$$\lambda = -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta)\dot{\theta}$$

The system shown in Eq. (31) can be transformed into a more proper representation for control. In this transformation, the system will become a square system (i.e., number of inputs equals the number of outputs), as well as, the constraint term will be eliminated from the equation. The following two matrices are defined to do this transformation:

$$\mathbf{v}_a(\mathbf{t}) = \begin{bmatrix} v \\ w \end{bmatrix} \quad (32)$$

$$S(\mathbf{q}) = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix}. \quad (33)$$

We can realize the in the modified forward kinematic matrix which has two velocity terms related to the distance between the robot centroid and the wheel axis. Therefore, we can write Eq. (34) for the system:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = S(\mathbf{q}) \mathbf{v}_a(\mathbf{t}) = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \quad (34)$$

It can easily be proved that $S(\mathbf{q})$ is in the null space of $A^T(\mathbf{q})$ matrix:

$$S^T(\mathbf{q}) A^T(\mathbf{q}) = 0. \quad (35)$$

By differentiating Eq. (34) we get:

$$\ddot{\mathbf{q}} = \dot{S}(\mathbf{q}) \mathbf{v}_a(\mathbf{t}) + S(\mathbf{q}) \dot{\mathbf{v}}_a(\mathbf{t}). \quad (36)$$

Substituting Eq. (34) and Eq. (36) in Eq. (31) and multiplying the resultant equation by $S^T(\mathbf{q})$, the result will be as follows:

$$\overline{M}(\mathbf{q}) \dot{\mathbf{v}}_a(\mathbf{t}) + \overline{V}(\mathbf{q}, \dot{\mathbf{q}}) \mathbf{v}_a(\mathbf{t}) = \overline{B}(\mathbf{q}) \boldsymbol{\tau}, \quad (37)$$

where

$$\overline{M}(\mathbf{q}) = S^T(\mathbf{q}) M(\mathbf{q}) S(\mathbf{q}) = \begin{bmatrix} m & 0 \\ 0 & I_c + m a^2 \end{bmatrix}$$

$$\overline{V}(\mathbf{q}, \dot{\mathbf{q}}) = S^T(\mathbf{q}) M(\mathbf{q}) \dot{S}(\mathbf{q}) + S^T(\mathbf{q}) M(\mathbf{q}) S(\mathbf{q}) \\ = \begin{bmatrix} 0 & -m a \dot{\theta} \\ m a \dot{\theta} & 0 \end{bmatrix}$$

$$\overline{B}(\mathbf{q}) = S^T(\mathbf{q}) B(\mathbf{q}) = \frac{1}{R_a} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}.$$

The new system in matrix form as shown in

$$\begin{bmatrix} m & 0 \\ 0 & I_c + m a^2 \end{bmatrix} \dot{\mathbf{v}}_a(\mathbf{t}) + \begin{bmatrix} 0 & -m a \dot{\theta} \\ m a \dot{\theta} & 0 \end{bmatrix} \mathbf{v}_a(\mathbf{t}) \\ = \frac{1}{R_a} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix} \boldsymbol{\tau}. \quad (38)$$

3 Controller design

In general, the trajectory tracking problem returns into designing a dynamic error model controller, therefore the following steps should be taken.

First, design the kinematic controller to make the position error asymptotically stable. Then, design a dynamic controller to compute the torque such that the mobile robot's velocities converge to the given velocity inputs from the kinematic controller. In other words, we divide the controller design into two stages. The first one, the kinematic controller to control the position, and the dynamics controller to control the speed (i.e., the output velocity of the kinematic controller is the input velocity to the dynamic controller), therefore the control scheme looks like Fig. 4.

3.1 Kinematic controller design

The kinematic controller used here uses two postures for the robot: the reference posture $\mathbf{q}_r = [x_r, y_r, \theta_r]^T$ and the current position of the robot as $\mathbf{q}_c = [x_c, y_c, \theta_c]^T$ we define then the tracking error model in the basis of the frame linked to the mobile platform or the local frame as follows:

$$\mathbf{q}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = R(\theta)(\mathbf{q}_r - \mathbf{q}_c) = R(\theta) \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta_c \end{bmatrix}. \quad (39)$$

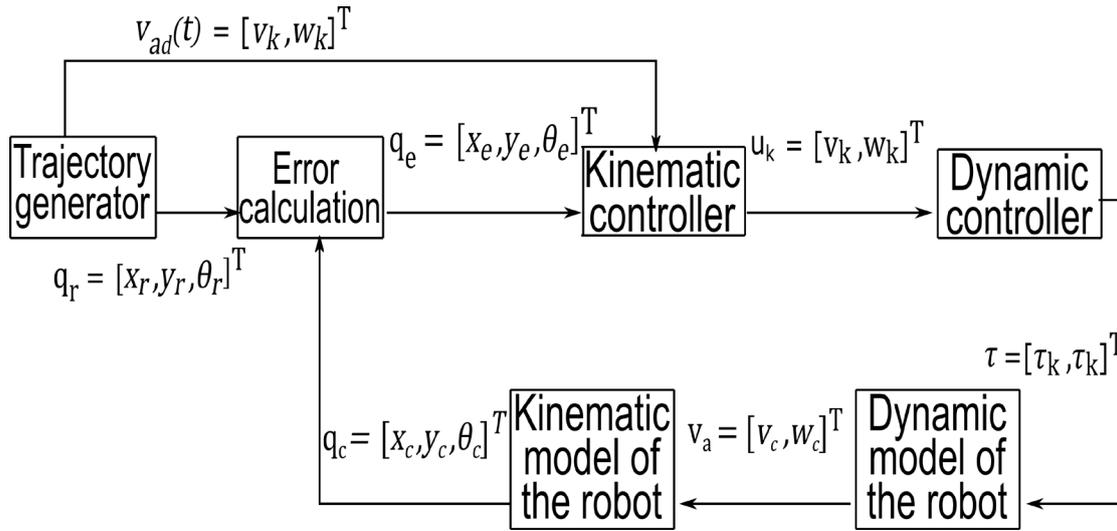


Fig. 4 Control system architecture

We differentiate Eq. (39) with respect to time; we get the error dynamics for the tracking problem which can be written as follows [5]:

$$\dot{x}_e = w y_e - v + v_r \cos \theta_e \quad (40)$$

$$\dot{y}_e = -w x_e + v_r \sin \theta_e \quad (41)$$

$$\dot{\theta}_e = w_d - w. \quad (42)$$

Proof of equations Eqs. (40)–(42) can be found in [22].

Clearly, $R(\theta)$ is invertible and nonsingular, thus:

$$\lim_{x \rightarrow \infty} (|x_e| + |y_e| + |\theta_e|) = 0 \quad (43)$$

leads to:

$$\lim_{x \rightarrow \infty} x_c = x_r \quad (44)$$

$$\lim_{x \rightarrow \infty} y_c = y_r \quad (45)$$

$$\lim_{x \rightarrow \infty} \theta_c = \theta_r. \quad (46)$$

From [47], it is shown and proved that if v_r and w_r are continuous and bounded, k_x, k_y, k_θ are bounded, \dot{v}_r and \dot{w}_r are small sufficiently and $v_r > 0$, then using the control law v_k will grant asymptotic stability in the sense of Lyapunov using the Lyapunov function V , where

$$v_k = \begin{bmatrix} v_r \cos e_\theta + k_x e_x \\ w_r + k_y v_r e_y + k_\theta v_r \sin e_\theta \end{bmatrix} \quad (47)$$

$$V = \frac{1}{2}(x_e^2 + y_e^2) + \frac{1}{k_y}(1 - \cos e_\theta). \quad (48)$$

Also, k_x, k_y, k_θ are positive constants, the stability of this control law was proved using the Lyapunov stability method in [22].

3.2 Dynamic controller design

The previous section was about selecting a velocity control $v(t)$ defined in Eq. (47) for the kinematic model of the robot. In this section, we illustrate our contribution to the literature which is converting this control velocity into a torque control τ for the actual physical robot.

Now, we consider the equations of motion of the non-holonomic mobile robot platform that is given by Eq. (34) and Eq. (38) and they are shown below:

$$\dot{q} = S(q)v_a(t) \quad (49)$$

$$\bar{M}(q)\dot{v}_a(t) + \bar{V}(q, \dot{q})v_a(t) = \bar{B}(q)\tau. \quad (50)$$

If we choose τ as follows:

$$\tau = \bar{B}^{-1}(q)[\bar{M}(q)\dot{v}_a(t) + \bar{V}(q, \dot{q})v_a(t)]. \quad (51)$$

Then, Eq. (50) becomes:

$$\dot{v}_a(t) = v. \quad (52)$$

Now we can choose $v = v_k$ from the kinematic controller.

4 Results and discussions

In this section, we illustrate the performance of the controller proposed through some simulation results for the tracking of a lemniscate trajectory. The proposed controller is simulated using MATLAB/SIMULINK, the parameters for the physical robot and DC motors are adopted

from [48] and [49]. Many simulations were executed, with different parameters each time, the ones we depict here are the ones that we were satisfied with.

The Feedback linearization-based controller is tested with perfect knowledge of the parameters; the initial robot pose was $q_0 = [0, 1, 0]^T$ and the initial reference position was $q_0^{ref} = [0, 0.75, \pi/4]^T$. The Polygon trajectories are represented using parametric equations like the following formulation:

$$traj = [(x(0) \ y(0)), (x(1) \ y(1)), \dots, (x(t) \ y(t))].$$

Where t is the simulation time, and the lemniscates trajectory was simulated with constant velocity $v_d = 0.4$ m/s, and was described by Eqs. (53)–(55):

$$x = 0 + \sin\left(\frac{t}{20}\right) \tag{53}$$

$$y = 0.8 + 2 \sin\left(\frac{t}{10}\right) \tag{54}$$

$$\theta_d = a \tan 2(y, x). \tag{55}$$

The results for the rectangle trajectory gotten by the proposed IOSFL controller is shown in Figs. 5–8, starting off with trajectory tracking in Fig. 5, then, tracking error of x , y and θ in Fig. 6, followed by the linear and angular velocity of the mobile robot and the desired velocity in Fig. 7. The same order for the lemniscates trajectory results from Figs. 8–10.

We conducted some experiments to ensure that the control system is stable, these experiments include changing the controller parameters, and the best values we found are in Table 1, the control parameters need to be sufficiently small to avoid oscillatory behavior and eventually instability.

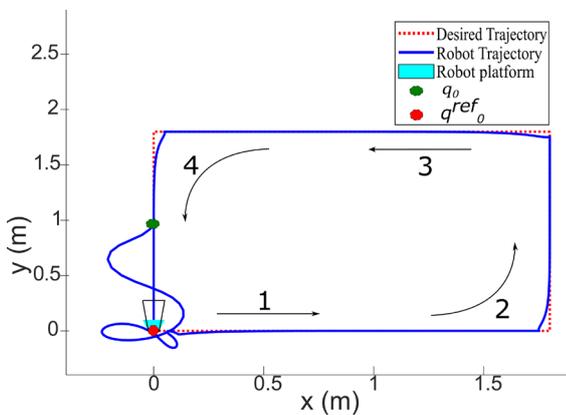


Fig. 5 $[x(t), y(t)]$ Plot for a rectangular trajectory

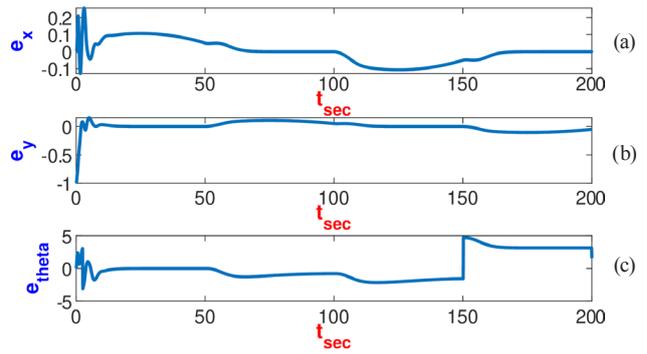


Fig. 6 Robot error results: (a) Robot longitudinal error e_x , (b) Robot lateral error e_y , (c) Robot heading error e_θ

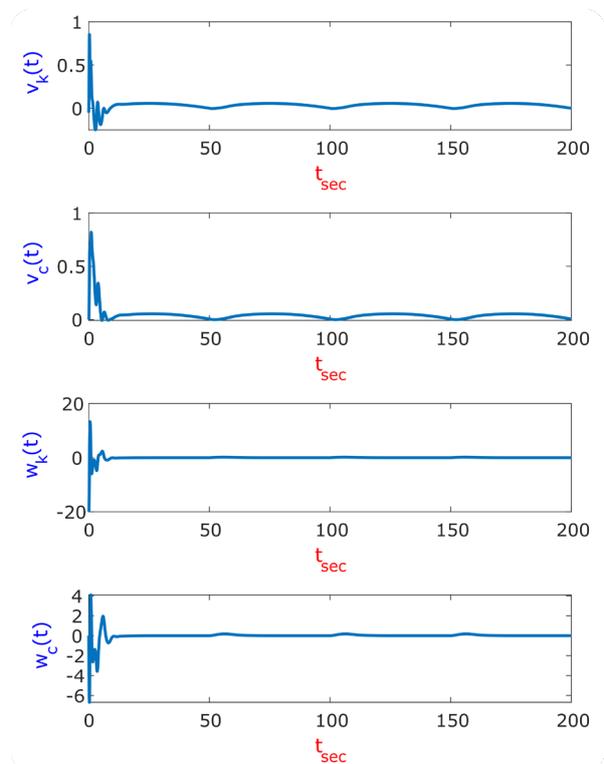


Fig. 7 Linear and angular velocities of the controller and the robot

4.1 First scenario: rectangular shape

The first simulation trajectory is a rectangular shape that has the configuration

$$traj1 = [(0 \ 0), (1.8 \ 0), (1.8 \ 1.8), (0 \ 1.8), (0 \ 0)].$$

The robot trajectory is represented by a continuous line and the reference trajectory is given with dots. The movement direction is indicated by arrows and successive numbering. The green circle indicates the initial position of the robot q_0 and the red circle indicates the initial position of the reference trajectory q_0^{ref} . Pay attention to the high position error at the beginning, which is because the reference

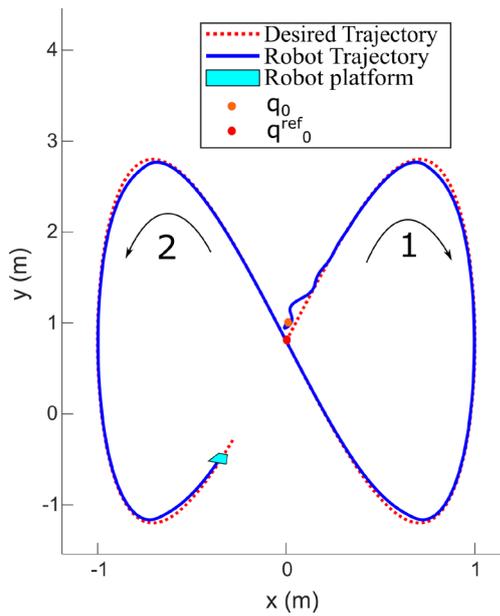


Fig. 8 $[x(t), y(t)]$ Result for a lemniscates trajectory

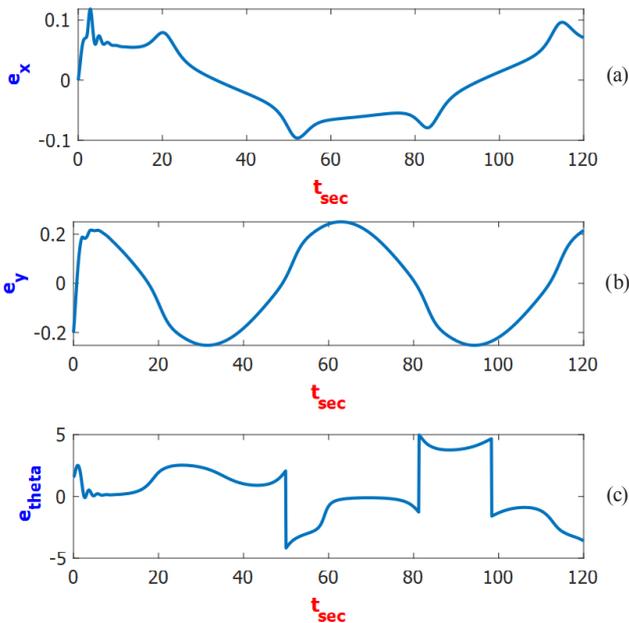


Fig. 9 Robot error results: (a) Robot longitudinal error e_x , (b) Robot lateral error e_y , (c) Robot heading error e_θ

trajectory starts at a point far from the initial position, after about the position error is kept in acceptable limits by the feedback linearization controller and the robot can achieve the objective in small time, roughly 8.5 sec, after the 10th second, the robot was already on the track, and stood there for the whole remaining 190 sec of the simulation, another observation to be noticed is the small error when the robot is turning, which can be reduced by reducing the turning speed thereby giving time for the robot to respond to the control signal, Next the three error components are depicted.

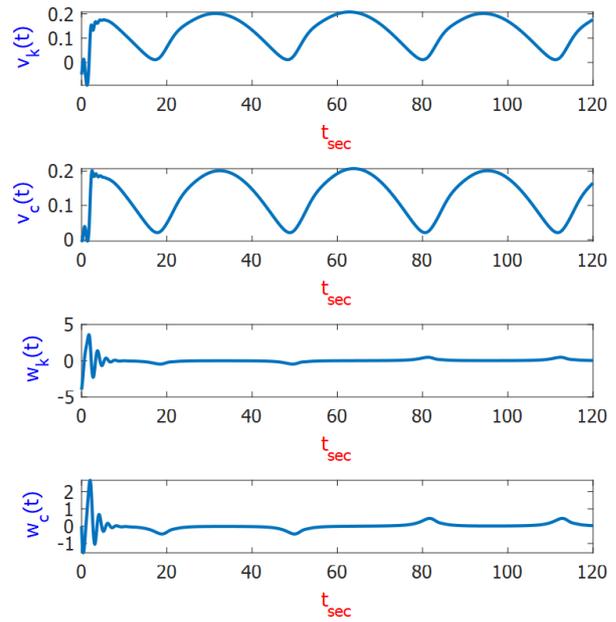


Fig. 10 Linear and angular velocities of the controller and the robot

Table 1 Simulation parameters

Parameter	Value	Unit
b : The distance between the centre of mass and wheels axis	0.05	[M]
R_a : The radius of the wheels	0.05	[M]
m : The robot mass	10	[Kg]
L : The lateral distance of the wheels to the center	0.5	[M]
J : The robot moment of inertia	5	[Kg M ²]
k_p 1 st gain of the dynamic controller (for linear velocity error)	10	Unitless
k_p 2 nd gain of the dynamic controller (for angular velocity error)	10	Unitless
k_x : The gain of the kinematic controller	10	1/S
k_y : The gain of the kinematic controller	0.0064	1/Cm
k_θ : The gain of the kinematic controller	0.16	1/Cm

As one can see from Fig. 6, the error convergence is properly ensured by the controller. The longitudinal and lateral errors are less than 0.1 m. Also, as we can see in the second plot (e_y), the error starts from -1 which is the difference between the starting point of the robot ($y = 1$) and the starting point of the reference trajectory ($y = 0$) that goes to zero after a short time and stays very close to it for the whole simulation.

The above Fig. 7 illustrates the desired linear velocity, the robot linear velocity, the desired angular velocity, and the robot current angular velocity respectively, as we can see in the results above, the current and desired velocities are almost exactly the same for all simulations, which means the controller is doing perfect tracking. In the interval $(]0, 45[\cup]55, 95[\cup]105, 145[\cup]145, 195[)$, the linear velocity is nonzero and the angular velocity is zero,

that's where the robot is moving in straight, in the interval $(]45,55[\cup]95,105[\cup]145,155[\cup]195,200[)$, the linear velocities v_k and v_c are zero, and the angular velocities w_k and w_c are nonzero, that's where the robot is rotating.

4.2 Second scenario: lemniscates trajectory

The robot trajectory is represented by a continuous line and the reference trajectory is given with dots. The movement direction is indicated by arrows and successive numbering. The orange circle indicates the initial position of the robot and the red circle indicates the initial position of the reference trajectory. As we can see, the position error is kept in acceptable limits by the feedback linearization controller (0.1 m) and the robot can achieve the objective in a small-time, somewhere around 4.5 sec, the reason why it was faster than the other scenario is that the robot's initial state was very close the reference's initial state. After the 5th second, the robot was already on the track and stood there for the whole remaining 105.5 sec of the simulation. Next, the three error components are depicted.

As we can see from Fig. 9, the error convergence is properly ensured by the controller. An observation to be noticed is the difference in errors when the reference is a polygon, and when it is a lemniscate, the errors in a polygon are only noticeable in turns, contrarily, in the lemniscates, the errors are always nonzero because the linear and angular velocities are both nonzero. The longitudinal and lateral errors are less than 0.1 m for the whole simulation after the robot gets on the trajectory (i.e., after 4.5 sec). As for the heading error, it is less than 4 rad since the beginning. Next, the linear and angular velocities results are depicted.

Fig. 10 illustrates the desired linear velocity on the top (first one), the robot linear velocity (the second one), the desired angular velocity (the third one), and the robot current angular velocity (fourth and last one), as we can see, the linear velocities are almost exactly the same for all simulations, same for the angular velocities, which means the controller is doing perfect tracking, the linear velocities v_k and v_c are positive, and the angular velocities w_k and w_c are zero, that means the robot is moving in a straight line, in the interval $(]0,18[\cup]22,48[\cup]52,78[\cup]82,111[\cup]113,120[)$, on the other hand, the linear velocities v_k and v_c are almost zero, and the angular velocities w_k and w_c are nonzero, that means the robot is turning, in the interval $(]18,22[\cup]48,52[\cup]79,83[\cup]109,111[)$.

One can also see that, the angular velocities are negative in $]18,22[$ and $]48,52[$, and that is because the robot is turning against theta's positive direction,

which is clockwise, however, the angular velocities in $]79,83[$ and $]109,111[$ are positive, because the robot is turning with theta's positive direction (i.e., counter-clockwise). Next, Figs. 11, 12 show a comparison between the proposed controller (on the left-hand side) and the same controller without the dynamics controller and model (on the right-hand side) for a rectangle reference trajectory and a lemniscates reference trajectory respectively, it is quite obvious that the dynamic system and dynamic controller take into account the system nonlinearities or when the motors aren't strong enough to allow for neglecting the dynamics, and gives a closer result to the real robot, especially at the first 10 sec of the simulation.

4.3 Comparative study between position control only and position and velocity control

To test the performance of the proposed controllers, we will show the difference between two control schemes, first one is when we use only the kinematics controller, the second one is when we use both controllers (kinematics and dynamics) to show the difference between the two, first we will start with the trajectory tracking plots, then tracking errors, and finally velocities (linear and angular).

As shown in Figs. 11, 12, the same reference trajectories for a rectangle and a lemniscate, respectively, are used for both simulations in both scenarios, represented by dashed lines, on the left, the simulation of the WMR using both controllers is shown and on the right, the simulation of the WMR using only the kinematic controller is shown, it is very obvious that in the start of the simulation, the kinematic controller doesn't show the error in tracking, whereas, when considering dynamics, it is taken into account the take-off of the car, which is due to the nonlinearities, the controller parameters, the saturations, this is very important when experimental implementations take place.

Figs. 13, 14 depict a comparison between the tracking error of the two controllers for both scenarios is made, on the left, the tracking error using the (kinematic and dynamic) controller, on the right, the tracking error using the (kinematic only) controller is shown, again, when the robot is taking off, the difference in errors is very clear. Finally, the linear and angular velocities plots will be depicted.

In Figs. 15, 16, we have the desired and robot's linear and desired and robot's angular velocities, in the top and bottom for the rectangle and lemniscates reference trajectories respectively, the results for the proposed controller are on the (left) and the for the kinematic controller are on the (right).

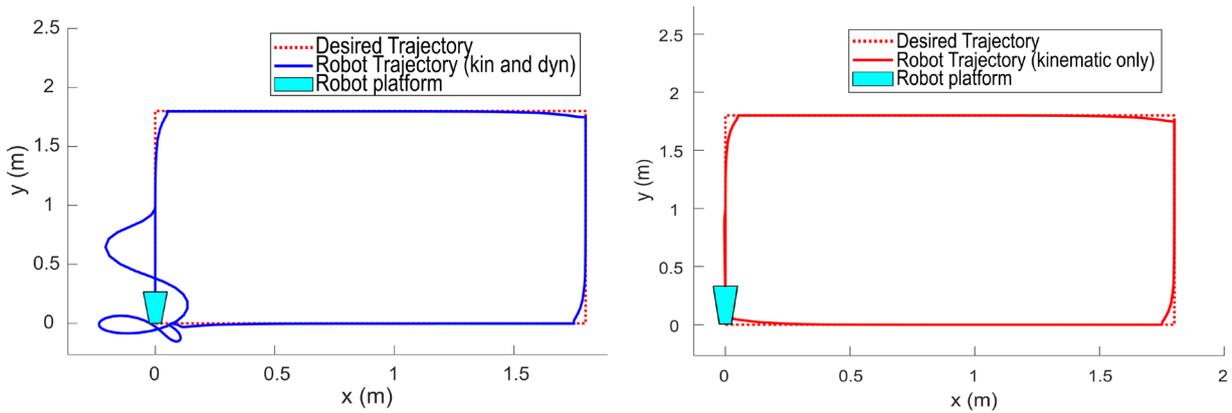


Fig. 11 Trajectory tracking behavior for a (dynamic and kinematic (blue)) controller on the left versus a (kinematic only (red)) controller on the right for the same rectangle reference trajectory

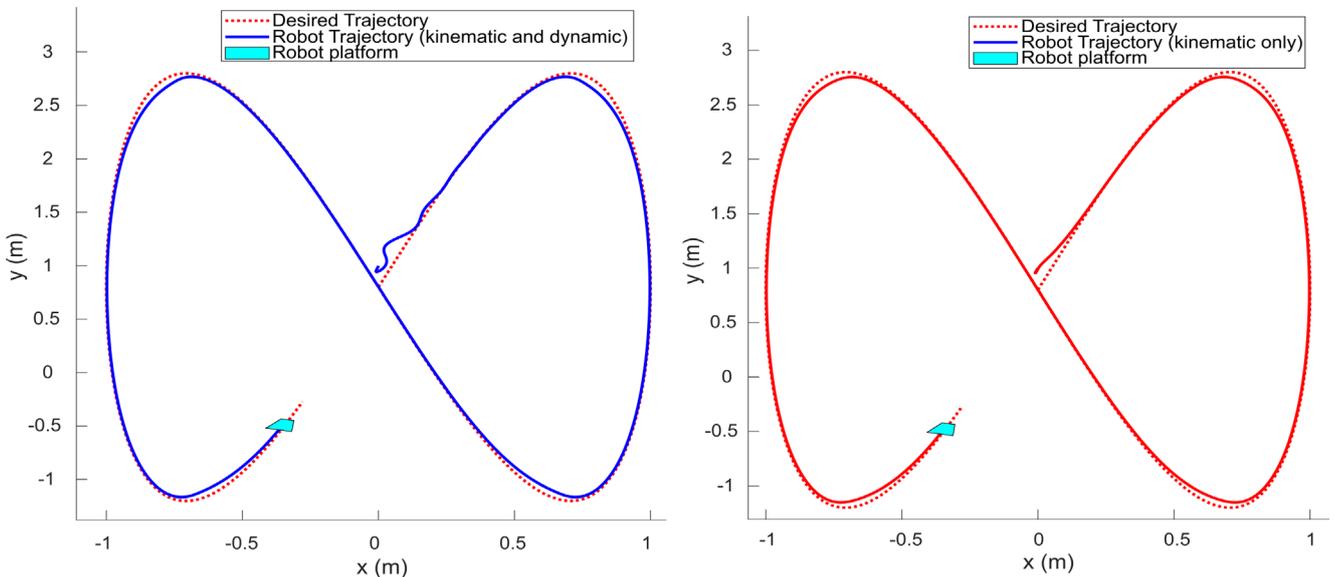


Fig. 12 Trajectory tracking behavior for a (dynamic and kinematic (blue)) controller on the left versus a (kinematic only (red)) the controller on the right for the same lemniscates reference trajectory

It is quite obvious that the proposed controller takes the velocities into account, which is an advantage with regards to the kinematic controller that has no influence on it.

5 Conclusion

Trajectory tracking and control of differential drive robots along different regular trajectories were introduced. The proposed controller uses a high-level controller which computes the linear and angular velocities then send them to a low-level controller that controls the motors. The IOSFL controller has been tested with perfect knowledge of the robot's parameters and dynamics and without any noise or disturbances. It is very sensitive to control parameters (this is one of its limitations), it is suitable for non-autonomous control systems such as trajectory

tracking of mobile robots [5]. Errors produced at turns are due to the turning radius of the robot, this error is found to be the main limitation of the current control strategy, however, this error at turns can be reduced by reducing the turning speed thereby giving time for the robot to respond to the control signal.

Considering the robot's dynamics make it more suitable for real-time experiments since it takes into account the internal forces of the system, these nonlinearities can't be neglected.

The future work includes penalizing the error in the start of the simulation as well as the development of a control algorithm to reduce the error in tracking when turning as seen in this paper which will be more realistic in the present scenario.

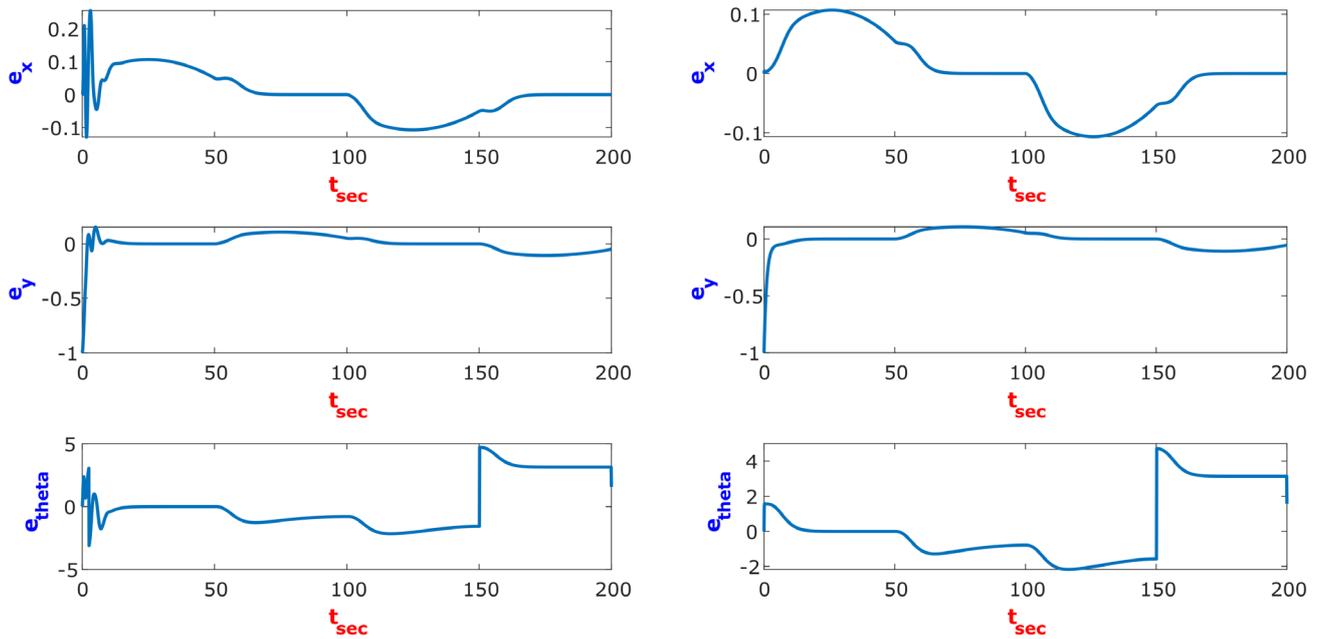


Fig. 13 A comparison between the tracking errors of the two controllers (proposed on the left and kinematic only on the right) for the rectangle scenario

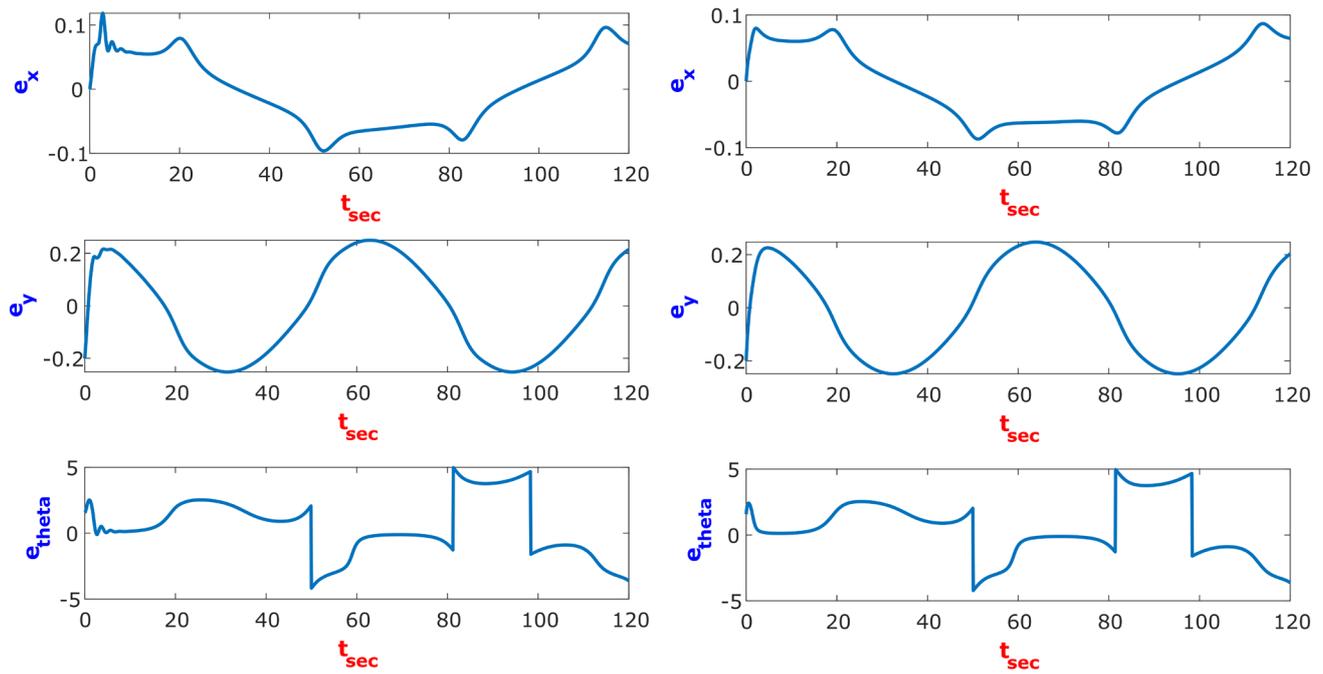


Fig. 14 A comparison between the tracking errors of the two controllers (proposed on the left and kinematic only on the right) for the lemniscates scenario

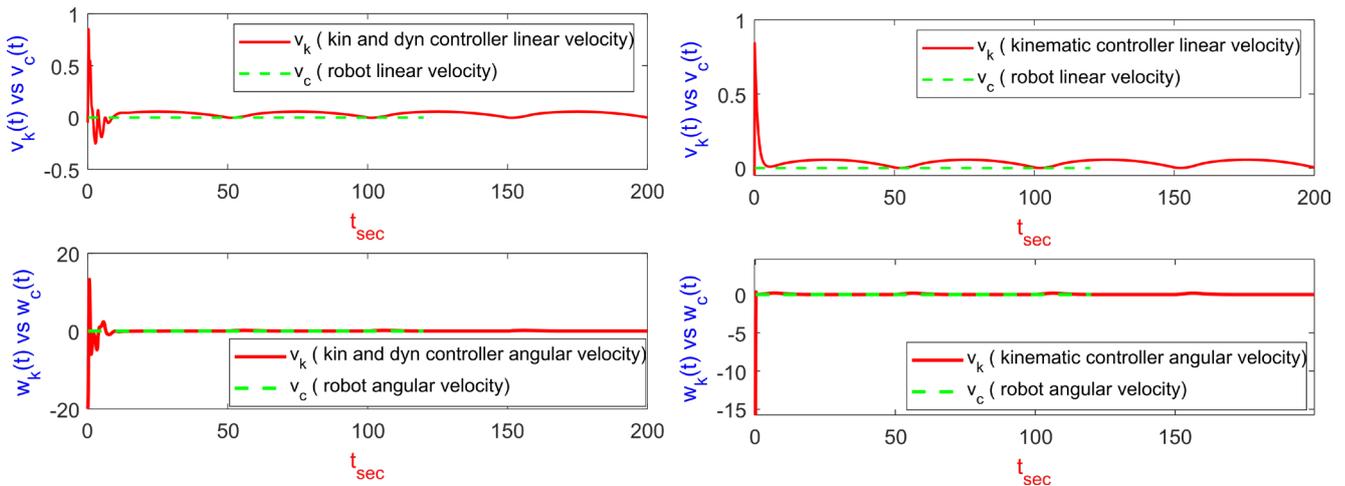


Fig. 15 A comparison between the linear velocities (top) and angular velocities (bottom) of the two controllers (proposed on the left and kinematic only on the right) for the rectangle scenario

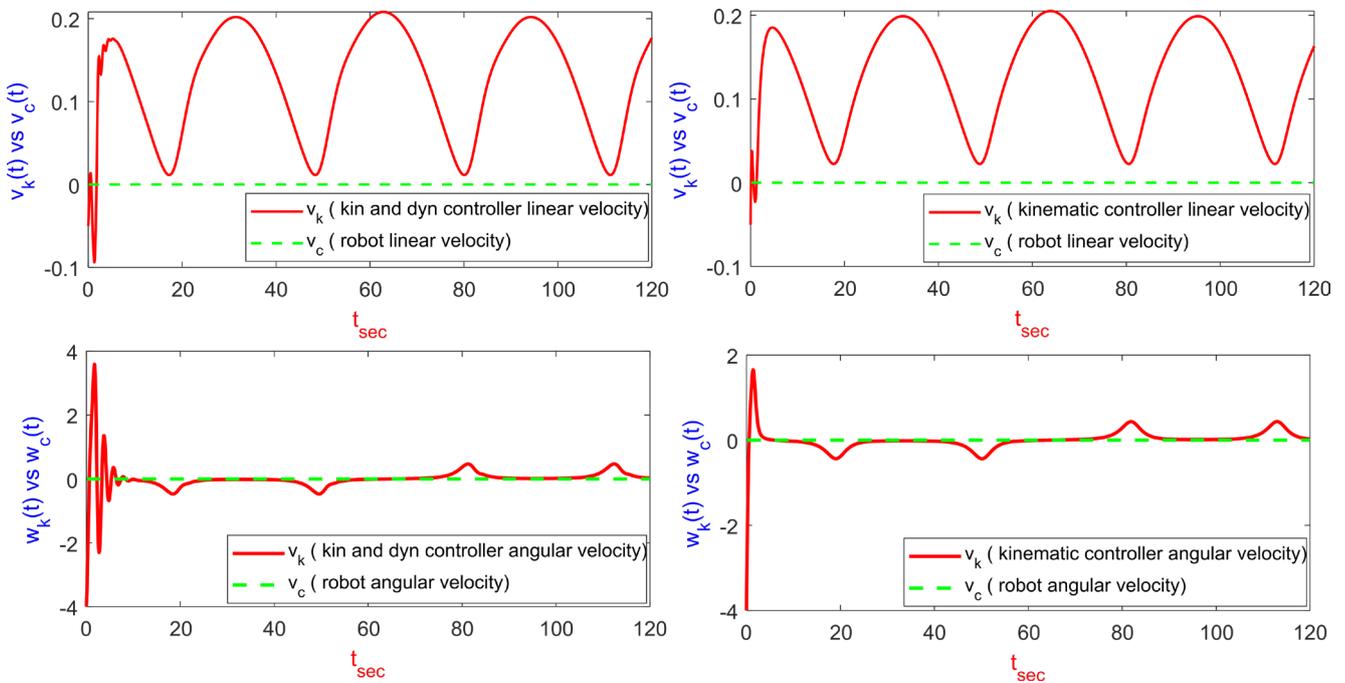


Fig. 16 A comparison between the linear velocities (top) and angular velocities (bottom) of the two controllers (proposed on the left and kinematic only on the right) for the lemniscates scenario

References

[1] Han, G. X., Chen, H., Ma, M. M., Zhao, H. Y. "约束非完整移动机器人轨迹跟踪的非线性预测控制" (Nonlinear Model Predictive Control of Trajectory Tracking for Non-holonomic Mobile Robots with Constraints), *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)*, 39(1), pp. 177–181, 2009. (in Chinese)
<https://doi.org/10.13229/j.cnki.jdxbgxb2009.01.048>

[2] Brockett, R. W. "Asymptotic Stability and Feedback Stabilization", In: Brockett, R. W., Millman, R. S., Sussmann, H. J. (eds.) *Differential Geometric Control Theory*, Birkhauser, Boston, MA, USA, 1983, pp. 181–208. [online] Available at: <http://hrl.harvard.edu/publications/brockett83asymptotic.pdf> [Accessed: 10 May 2020]

[3] Oriolo, G., De Luca, A., Vendittelli, M. "WMR control via dynamic feedback linearization: design, implementation, and experimental validation", *IEEE Transactions on Control Systems Technology*, 10(6), pp. 835–852, 2002.
<https://doi.org/10.1109/TCST.2002.804116>

[4] Sharma, K. R., Dušek, F., Honc, D. "Comparative study of predictive controllers for trajectory tracking of non-holonomic mobile robot", In: *2017 21st International Conference on Process Control*, Strbske Pleso, Slovakia, 2017, pp. 197–203.
<https://doi.org/10.1109/PC.2017.7976213>

- [5] Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T. "A stable tracking control method for an autonomous mobile robot", In: IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 1990, pp. 384–389.
<https://doi.org/10.1109/ROBOT.1990.126006>
- [6] Khalil, H. K. "Nonlinear Systems", Prentice Hall, Upper Saddle River, NJ, USA, 2002.
- [7] Kolmanovsky, I., McClamroch, N. H. "Developments in nonholonomic control problems", IEEE Control Systems Magazine, 15(6), pp. 20–36, 1995.
<https://doi.org/10.1109/37.476384>
- [8] Sarkar, N., Yun, X., Kumar, V. "Control of Mechanical Systems With Rolling Constraints: Application to Dynamic Control of Mobile Robots", The International Journal of Robotics Research, 13(1), pp. 55–69, 1994.
<https://doi.org/10.1177/027836499401300104>
- [9] Luca, A. D., Oriolo, G. "Modelling and Control of Nonholonomic Mechanical Systems", In: Angeles, J., Kecskeméthy, A. (eds.) Kinematics and Dynamics of Multi-Body Systems, Springer, Vienna, Austria, 1995, pp. 277–342.
https://doi.org/10.1007/978-3-7091-4362-9_7
- [10] Nascimento, T. P., Dorea, C. E. T., Gonçalves, L. M. G. "Nonlinear model predictive control for trajectory tracking of non-holonomic mobile robots: A modified approach", International Journal of Advanced Robotic Systems, 15(1), Article number: 1729881418760461, 2018.
<https://doi.org/10.1177/1729881418760461>
- [11] Al Khatib, E. I., Al-Masri, W. M. F., Mukhopadhyay, S., Jaradat, M. A., Abdel-Hafez, M. "A comparison of adaptive trajectory tracking controllers for wheeled mobile robots", In: 2015 10th International Symposium on Mechatronics and its Applications (ISMA), Sharjah, United Arab Emirates, 2015, pp. 1–6.
<https://doi.org/10.1109/ISMA.2015.7373485>
- [12] Merabti, H., Belarbi, K., Bouchemal, B. "Nonlinear predictive control of a mobile robot: a solution using metaheuristics", Journal of the Chinese Institute of Engineers. 39(3), pp. 282–290, 2016.
<https://doi.org/10.1080/02533839.2015.1091276>
- [13] Xue, J., Dong, S., Yan, J. "Sliding Mode Variable Structure Based Path Following Control of Agricultural Robots with Differential Drive", IFAC-PapersOnLine, 51(17), pp. 455–459, 2018.
<https://doi.org/10.1016/j.ifacol.2018.08.173>
- [14] Axenie, C., Solea, R. "Real time control design for mobile robot fault tolerant control. Introducing the ARTEMIC powered mobile robot", In: IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Qingdao, China, 2010, pp. 7–13.
<https://doi.org/10.1109/MESA.2010.5552026>
- [15] Goswami, N. K., Padhy, P. K. "Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance", Computers & Electrical Engineering, 72, pp. 307–323, 2018.
<https://doi.org/10.1016/j.compeleceng.2018.09.021>
- [16] Yang, E., Gu, D., Mita, T., Hu, H. "Nonlinear tracking control of a car-like mobile robot via dynamic feedback linearization", In: Control 2004, Bath, UK, 2004, Paper number: ID-218. [online] Available at: <http://ukacc.group.shef.ac.uk/proceedings/control2004/Papers/218.pdf> [Accessed: 10 May 2020]
- [17] Caracciolo, L., de Luca, A., Iannitti, S. "Trajectory tracking control of a four-wheel differentially driven mobile robot", In: 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), Detroit, MI, USA, 1999, pp. 2632–2638.
<https://doi.org/10.1109/ROBOT.1999.773994>
- [18] Bendtsen, J. D., Andersen, P., Pedersen, T. S. "Robust Feedback Linearization-based Control Design for a Wheeled Mobile Robot", presented at 6th International Symposium on Advanced Vehicle Control (AVEC'02), Hiroshima, Japan, Sept., 9-13, 2002.
- [19] d'Andréa-Novel, B., Campion, G., Bastin, G. "Control of Nonholonomic Wheeled Mobile Robots by State Feedback Linearization", The International Journal of Robotics Research, 14(6), pp. 543–559, 1995.
<https://doi.org/10.1177/027836499501400602>
- [20] Kühne, F., da Silva Jr, J. M. G., Lages, W. F. "Mobile Robot Trajectory Tracking Using Model Predictive Control", presented at II IEEE Latin-American Robotics Symposium, Sao Luis, Brazil, Sept., 18-23, 2005.
- [21] Li, Z., Deng, J., Lu, R., Xu, Y., Bai, J., Su, C.-Y. "Trajectory-Tracking Control of Mobile Robot Systems Incorporating Neural-Dynamic Optimized Model Predictive Approach", IEEE Transactions on Systems, Man, and Cybernetics: Systems, 46(6), pp. 740–749, 2016.
<https://doi.org/10.1109/TSMC.2015.2465352>
- [22] Klančar, G., Škrjanc, I. "Tracking-error model-based predictive control for mobile robots in real time", Robotics and Autonomous Systems, 55(6), pp. 460–469, 2007.
<https://doi.org/10.1016/j.robot.2007.01.002>
- [23] Wang, D., Wei, W., Yeboah, Y., Li, Y., Gao, Y. "A Robust Model Predictive Control Strategy for Trajectory Tracking of Omnidirectional Mobile Robots", Journal of Intelligent & Robotic Systems, 98(2), pp. 439–453, 2020.
<https://doi.org/10.1007/s10846-019-01083-1>
- [24] Shi, J., Sun, D., Qin, D., Hu, M., Kan, Y., Ma, K., Chen, R. "Planning the trajectory of an autonomous wheel loader and tracking its trajectory via adaptive model predictive control", Robotics and Autonomous Systems, 131, Article number: 103570, 2020.
<https://doi.org/10.1016/j.robot.2020.103570>
- [25] Velagic, J., Lacevic, B., Osmic, N. "Nonlinear Motion Control of Mobile Robot Dynamic Model", In: Jing, X.-J. (ed.) Motion Planning, InTech, Rijeka, Croatia, 2008, pp. 529–550.
<https://doi.org/10.5772/5997>
- [26] Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., Bastos-Filho, T. F. "An adaptive dynamic controller for autonomous mobile robot trajectory tracking", Control Engineering Practice, 16(11), pp. 1354–1363, 2008.
<https://doi.org/10.1016/j.conengprac.2008.03.004>
- [27] Yi, J., Song, D., Zhang, J., Goodwin, Z. "Adaptive Trajectory Tracking Control of Skid-Steered Mobile Robots", In: 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 2605–2610.
<https://doi.org/10.1109/ROBOT.2007.363858>
- [28] Shojaei, K., Shahri, A. M., Tarakameh, A., Tabibian, B. "Adaptive trajectory tracking control of a differential drive wheeled mobile robot", Robotica, 29(3), pp. 391–402, 2011.
<https://doi.org/10.1017/S0263574710000202>

- [29] Matraji, I., Al-Durra, A., Haryono, A., Al-Wahedi, K., Abou-Khousa, M. "Trajectory tracking control of Skid-Steered Mobile Robot based on adaptive Second Order Sliding Mode control", *Control Engineering Practice*, 72, pp. 167–176, 2018. <https://doi.org/10.1016/j.conengprac.2017.11.009>
- [30] Binh, N. T., Tung, N. A., Nam, D. P., Quang, N. H. "An Adaptive Backstepping Trajectory Tracking Control of a Tractor Trailer Wheeled Mobile Robot", *International Journal of Control, Automation and Systems*, 17, pp. 465–473, 2019. <https://doi.org/10.1007/s12555-017-0711-0>
- [31] Mohareri, O., Dhaouadi, R., Rad, A. B. "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks", *Neurocomputing*, 88, pp. 54–66, 2012. <https://doi.org/10.1016/j.neucom.2011.06.035>
- [32] Asif, M., Khan, M. J., Cai, N. "Adaptive sliding mode dynamic controller with integrator in the loop for nonholonomic wheeled mobile robot trajectory tracking", *International Journal of Control*, 87(5), pp. 964–975, 2014. <https://doi.org/10.1080/00207179.2013.862597>
- [33] Park, B. S., Yoo, S. J., Park, J. B., Choi, Y. H. "A Simple Adaptive Control Approach for Trajectory Tracking of Electrically Driven Nonholonomic Mobile Robots", *IEEE Transactions on Control Systems Technology*, 18(5), pp. 1199–1206, 2010. <https://doi.org/10.1109/TCST.2009.2034639>
- [34] Shojaei, K., Shahri, A. M., Tarakameh, A. "Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties", *Robotics and Computer-Integrated Manufacturing*, 27(1), pp. 194–204, 2011. <https://doi.org/10.1016/j.rcim.2010.07.007>
- [35] Bai, G., Liu, L., Meng, Y., Luo, W., Gu, Q., Wang, J. "Path Tracking of Wheeled Mobile Robots Based on Dynamic Prediction Model", *IEEE Access*, 7, pp. 39690–39701, 2019. <https://doi.org/10.1109/ACCESS.2019.2903934>
- [36] Tawfik, M. A., Abdulwahb, E. N., Swadi, S. M. "Trajectory Tracking Control for a Wheeled Mobile Robot Using Fractional Order PI^D Controller", *Al-Khwarizmi Engineering Journal*, 10(3), pp. 39–52, 2014. [online] Available at: <https://alkej.uobaghdad.edu.iq/index.php/alkej/article/view/459> [Accessed: 15 May 2020]
- [37] Fierro, R., Lewis, F. L. "Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics", *Journal of Robotic Systems*, 14(3), pp. 149–163, 1997. [https://doi.org/10.1002/\(SICI\)1097-4563\(199703\)14:3<149::AID-ROBI>3.0.CO;2-R](https://doi.org/10.1002/(SICI)1097-4563(199703)14:3<149::AID-ROBI>3.0.CO;2-R)
- [38] Dumitrascu, B., Filipescu, A., Minzu, V., Filipescu Jr, A. "Backstepping Control of Wheeled Mobile Robots", presented at 15th International Conference on System Theory, Control and Computing, Sinaia, Romania, Oct., 14–16, 2011.
- [39] Cherrotin, L., Boumehraz, M. "Fuzzy Logic and Reinforcement Learning based Approaches for Mobile Robot Navigation in Unknown Environment", *The Mediterranean Journal of Measurement and Control*, 9(3), pp. 109–117, 2013. [online] Available at: <http://archives.univ-biskra.dz/handle/123456789/7511> [Accessed: 15 May 2020]
- [40] Maurović, I., Baotić, M., Petrović, I. "Explicit Model Predictive Control for trajectory tracking with mobile robots", In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, Hungary, 2011, pp. 712–717. <https://doi.org/10.1109/AIM.2011.6027140>
- [41] Sharma, K. R., Honc, D., Dusek, F. "Predictive Control of Differential Drive Mobile Robot Considering Dynamics and Kinematics", In: 30th European Conference on Modelling and Simulation, Regensburg, Germany, 2016, pp. 354–360. <https://doi.org/10.7148/2016-0354>
- [42] Kühne, F., Lages, W. F., da Silva Jr, J. M. G. "Model Predictive Control of a Mobile Robot Using Linearization", In: International Conference on Mechatronics and Robotics, Aachen, Germany, 2004, pp. 525–530. [online] Available at: http://www.ece.ufrgs.br/~fetter/mechrob04_553.pdf [Accessed: 10 May 2020]
- [43] Lages, W. F., Alves, J. A. V. "Real-time Control of a Mobile Robot Using Linearized Model Predictive Control", *IFAC Proceedings Volumes*, 39(16), pp. 968–973, 2006. <https://doi.org/10.3182/20060912-3-DE-2911.00166>
- [44] Dhaouadi, R., Hatab, A. A. "Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework", *Advances in Robotics & Automation*, 2(2), Article number: 107, 2013. <https://doi.org/10.4172/2168-9695.1000107>
- [45] Muir, P. F., Neuman, C. P. "Kinematic modeling of wheeled mobile robots", *Journal of Robotic Systems*, 4(2), pp. 281–340, 1987. <https://doi.org/10.1002/rob.4620040209>
- [46] Siegwart, R., Nourbakhsh, I. R., Scaramuzza, D. "Mobile Robot Kinematics", In: *Introduction to Autonomous Mobile Robots*, The MIT Press, Cambridge, MA, USA, 2011, pp. 57–99.
- [47] Jiang, Z.-P., Nijmeijer, H. "Tracking Control of Mobile Robots: A Case Study in Backstepping", *Automatica*, 33(7), pp. 1393–1399, 1997. [https://doi.org/10.1016/S0005-1098\(97\)00055-1](https://doi.org/10.1016/S0005-1098(97)00055-1)
- [48] Corradini, M. L., Orlando, G. "Control of mobile robots with uncertainties in the dynamical model: a discrete time sliding model approach with experimental results", *Control Engineering Practice*, 10(1), pp. 23–34, 2002. [https://doi.org/10.1016/S0967-0661\(01\)00109-5](https://doi.org/10.1016/S0967-0661(01)00109-5)
- [49] Das, T., Kar, I. N. "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots", *IEEE Transactions on Control Systems Technology*, 14(3), pp. 501–510, 2006. <https://doi.org/10.1109/TCST.2006.872536>