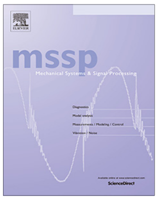
Mechanical Systems and Signal Processing 134 (2019) 106319

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp



A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots

Pouya Panahandeh, Khalil Alipour*, Bahram Tarvirdizadeh, Alireza Hadi

Advanced Service Robots (ASR) Laboratory, Department of Mechatronics Engineering, Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:
Received 3 June 2019
Received in revised form 17 August 2019
Accepted 20 August 2019
Available online 30 August 2019

Keywords:
Mobile robot
Differentially driven wheeled robot
Posture stabilization
Robot kinematics

ABSTRACT

In this paper, two non-smooth kinematic control strategies are proposed for the posture stabilization of a differentially driven wheeled mobile robot. The developed approach is based on kinematic coordinate transformation and Lyapunov-like stability technique. The presented control laws can stabilize the system asymptotically at a desired target and offer two different manoeuvers. Further, the response of the suggested algorithms is compared against recent previous studies. Next, the control laws are modified to tackle the obstacle avoidance problem while the robot approaches the final desired posture. The obtained simulation results as well as experimental tests prove the effectiveness of the presented techniques.

1. Introduction

Due to the today's applications of the wheeled vehicles in human life including industry, transportation and security, a future with wide usage of Wheeled Mobile Robots (WMRs) can be predicted. This foresight attracted a lot of researches and investments to the challenges of WMRs in last two decades [1,2]. The main problems in the field of motion control of WMRs can be classified into three main categories. Trajectory tracking, path following and point stabilization are common issues that can be considered for all types of WMRs such as differentially driven robots, car-like robots and tractor-trailers [3,4].

In the point or posture (orientation and position) regulation problems, in the absence of obstacles, the aim is to move the robot to a desired configuration, starting from an initial posture. Considering condition of perfect rolling on the robot wheels causes a special limitation for WMRs in their kinematics that categorizes them as typical nonholonomic systems. This restriction makes the problem of point stabilization of WMRs challenging because one cannot create a differentiable or even continuous pure state feedback control to stabilize the robot at a desired target [5].

The first step to achieve a control law in many researches is transformation of coordinate or system states. This conversion is mainly used to make the process of reaching to the control law easy and possible. Chained form is one of the well-known approaches to convert the system inputs and states. This method is also efficient in the cases of car-like robot and tractor-trailer one. Different feedback strategies have been proposed to stabilize nonholonomic systems in the chained form [6–9]. Another familiar form of the transformation includes the change of kinematic or dynamic coordinate from Cartesian coordinate to Polar coordinate. Utilizing this conversion modifies the response of the system and produces optimum results [10–13], but in some cases, the created control laws have singularities [13].

* Corresponding author.
E-mail address: k.alipour@ut.ac.ir (K. Alipour).

https://doi.org/10.1016/j.ymssp.2019.106319
0888-3270/© 2019 Elsevier Ltd. All rights reserved.

In the current study, a transformation in Cartesian coordinate will be proposed. This transformation introduces the position and orientation of desired postures in the coordinate that attaches to the robot. This conversion previously has been employed for the trajectory tracking problem of the WMRs [14–17]. After transformation of the coordinate, various methods have been utilized to obtain the appropriate control law, in the previous studies. Time-varying stabilization [18,19], hybrid stabilization [9] and discontinuous time-invariant stabilization [20] are three main strategies which were applied to the systems. However, most of the previous studies suffer from the lack of simultaneous simplicity and good performance, which motivates the authors to search for new methods. To solve this issue, a novel lower bounded function and Barbalat's lemma are developed in this study.

In order to examine the ability of the presented methods of this paper, two works are selected. One of them is a finite time control rule, which was presented in Ref. [21]. Its control law has two levels. In the first phase, one of the position error and orientation error should be vanished and the other error should be eliminated in the second phase. Therefore, such a two-phase algorithm leads the traveled path to get away from its optimality conditions, and also discontinuity occurs in switching time. A similar finite time controller also can be found in Ref. [22]. The second algorithm, which is selected for the comparison, is based on the work in Ref. [23]. In Ref. [23], A Vector-Field-Orientation feedback control was designed for point stabilization. The produced path of mentioned method for point stabilization problem is near optimal. However, the control law is complex. On the other hand, a feedback control laws was published in Ref. [13]. Despite the fact that it has a good performance, its control rule has discontinuity in the origin. In addition, some papers suggested a mixed controller or a unified framework for point stabilization and tracking control of the differential drive robots [24,25]. Ref. [24] proposed a control strategy based on Pointwise Angle Minimization method and Ref. [25] applied the integration of backstepping technique and neural dynamics. The problem of point stabilization was also investigated for tractor-trailer wheeled mobile robots in Ref. [26]. However, the proposed two-stage controller cannot attain the desired posture using a single maneuver.

In this paper, we address the posture stabilization problem of the WMRs involving geometric transformation in Cartesian coordinates. The main objective is to achieve two independent control rules, allowing the robot to move from a given initial location to a final desired posture including specific position and orientation. The existence of two solutions for each stabilization puts an additional degree of freedom into our developed solution. This privilege seems to be necessary in the presence of the obstacles and cooperating of robots. Moreover, the function of controller in the presence of obstacle is briefly explained.

This paper is organized as follows. In Section 2, the model of the robot is introduced. Section 3 contains the process of obtaining two control laws for stabilization of the robot. Section 4 provides numerical simulations and comparison of the obtained results with two latest methods in point stabilization. In Section 5, two experiments will be planned to evaluate the suggested control laws capability in practice. The problem of obstacle avoidance is explained in Section 6. Finally, Section 7 concludes this work.

2. Mobile robot model

The position and orientation of a mobile robot, which is located on a 2D plane, can be defined with a vector. This vector possesses three parameters in Cartesian coordinate. Two parameters specify the position of the robot and another parameter defines the orientation of the robot. In the point stabilization problem, two sets of configurations are important. The first one is the actual posture of the vehicle at the current moment, $\mathbf{p}_\mathrm{c} = [x_\mathrm{c}, y_\mathrm{c}, \theta_\mathrm{c}]^\mathrm{T}$, and the second one is the goal posture of the vehicle, $\mathbf{p}_\mathrm{r} = [x_\mathrm{r}, y_\mathrm{r}, \theta_\mathrm{r}]^\mathrm{T}$, which controller tries to carry the robot to that point.

From the motion control perspective, a mobile robot has two control inputs as u and ω, which u is the forward velocity and ω is angular velocity of the mobile robot, which are noted by q in Eq. 1 as follows.

$$\mathbf{q} = \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{1}$$

These kinematic inputs are related to angular velocities of the robot actuated wheels as,

$$\begin{aligned} u &= \tfrac{r}{2}(\dot{\alpha}_\mathrm{r} + \dot{\alpha}_\mathrm{l}) \\ \omega &= \tfrac{r}{2b}(\dot{\alpha}_\mathrm{r} - \dot{\alpha}_\mathrm{l}) \end{aligned} \tag{2}$$

where r is the radius of the wheels, $2b$ is the distance between robot actuated wheels (see Fig. 1), $\dot{\alpha}_r$ and $\dot{\alpha}_l$ are the angular velocities of the right and the left wheels, respectively. Linear and angular velocities of the robot can completely control the movement of the vehicle. Therefore, it is necessary to find the relation between these two control inputs and posture's variation of the robot in Cartesian coordinate. This relation shapes the vehicle's kinematic [14]. The kinematic of the robot can be defined by a Jacobian matrix J in Eq. (3), as follow.

$$\begin{bmatrix} \dot{x}_\mathrm{c} \\ \dot{y}_\mathrm{c} \\ \dot{\theta}_\mathrm{c} \end{bmatrix} = \dot{\mathbf{p}}_\mathrm{c} = \mathbf{Jq} = \begin{bmatrix} \cos\theta_\mathrm{c} & 0 \\ \sin\theta_\mathrm{c} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{3}$$

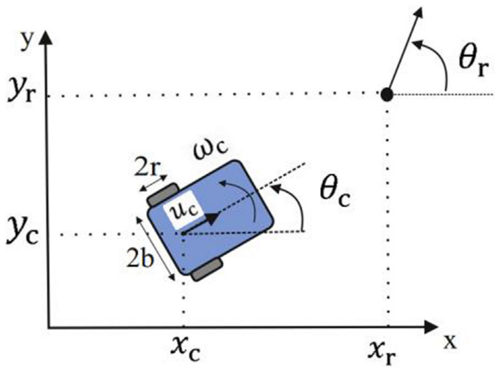


Fig. 1. Differentially driven mobile robot. Goal posture and Current posture.

The pure rolling assumption and non-slip condition of vehicle's wheels affect the vehicle's kinematics and omit one degree of freedom of the robot motion. This assumption causes a relation as follows.

$$\dot{y}_c \cos\theta_c - \dot{x}_c \sin\theta_c = 0 \tag{4}$$

ω_c and u_c, which are used in the Fig. 1, are angular and linear velocities of the robot. These two inputs are chosen by means of the control law and then they are fed to the robot.

Now an error posture, $\mathbf{p}_e$, can be defined, which is a description of the goal posture, $\mathbf{p}_r$, in a local coordinate system with origin of (x_c, y_c) and an X-axis in the direction of θ_c. The aforementioned control inputs, ω_c and u_c, try to minimize the error posture and drive the robot to goal posture.

$$\mathbf{p_e} = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} (\mathbf{p}_r - \mathbf{p}_c) \tag{5}$$

and the derivative of the error can be calculated in Eq. (6), as follows:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \dot{\mathbf{p}}_e \tag{6}$$

In the below calculations, the non-slip relation in (4) is employed and we should notice that $\dot{x}_r$, $\dot{y}_r$ and $\dot{\theta}_r$ are all zero because x_r, y_r and θ_r are constant.

$$\dot{x}_e = (\dot{x}_r - \dot{x}_c)\cos\theta_c + (\dot{y}_r - \dot{y}_c)\sin\theta_c - (x_r - x_c)\dot{\theta}_c\sin\theta_c + (y_r - y_c)\dot{\theta}_c\cos\theta_c = y_e\omega_c - u_c \tag{7}$$

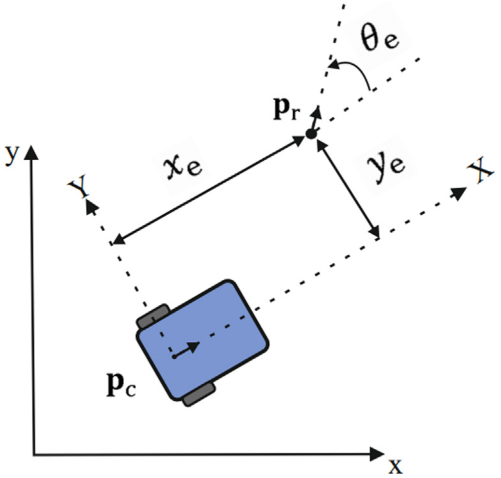


Fig. 2. Robot posture error coordinate.

$$\dot{y}_e = -(\dot{x}_r - \dot{x}_c)\sin\theta_c + (\dot{y}_r - \dot{y}_c)\cos\theta_c - (x_r - x_c)\dot{\theta}_c\cos\theta_c - (y_r - y_c)\dot{\theta}_c\sin\theta_c = -x_e\omega_c + \dot{x}_c\sin\theta_c - \dot{y}_c\cos\theta_c = -x_e\omega_c \tag{8}$$

$$\dot{\theta}_e = -\dot{\theta}_c = -\omega_c \tag{9}$$

To derivate the Lyapunov-like function in the next section, the computation of the above terms is essential.

3. Control law design

In this section, control laws are developed and the proof of stabilization of robot at the desired configuration will be studied. To this end, a Lyapunov-like function is defined as,

$$L = \frac{1}{2}\left({x_e}^2 + {y_e}^2\right) \tag{10.a}$$

The derivative of the above function can be calculated as,

$$\dot{L} = \dot{x}_e x_e + \dot{y}_e y_e \tag{10.b}$$

Considering the dynamic of the position error in Eqs. (7) and (8), $\dot{L}$ can be written as,

$$\dot{L} = (y_e\omega_c - u_c)x_e + (-x_e\omega_c)y_e = -u_c x_e \tag{10.c}$$

If it is assumed that ω_c is bounded and the linear velocity of robot has been selected as $u_c = k_x x_e$, where $k_x > 0$, then the derivative of the Lyapunov-like function is:

$$\dot{L} = -k_x {x_e}^2 \tag{10.d}$$

Considering the fact that $\dot{L} \le 0$, hence $L \le L(0)$ and it can be concluded that x_e and y_e are bounded. Now it is essential to derive $\ddot{L}$.

$$\ddot{L} = -2k_x x_e \dot{x}_e = -2k_x x_e(y_e\omega_c - u_c) = -2k_x x_e(y_e\omega_c - k_x x_e) \tag{10.e}$$

L is a lower bounded function, $\dot{L}$ is negative semi-definite and $\ddot{L}$ is bounded, then due to the Barbalat's lemma [27], it can be concluded that $\dot{L} \to 0$, which means $x_e \to 0$. Suppose that x_e smoothly converges to zero, then $\dot{x}_e \to 0$. This implies that:

$$\dot{x}_e = y_e\omega_c - u_c = y_e\omega_c - k_x x_e = y_e\omega_c = 0 \tag{10.f}$$

Suppose that ω_c is a nonzero and bounded variable, this assumption results in $y_e \to 0$. Now it is only needed to find the appropriate angular velocity, which should satisfy its boundedness constraint. Generally, the orientation of robot, θ, is a periodic variable with 2π periodicity. Therefore, this orientation can be limited in $[0, 2\pi)$ interval without making any changes to the whole problem. This modification should be done for both actual and desired orientation. This idea comes from the fact that in the experimental tests, Atan2(.) function, which is bounded in the domain of $(-\pi,\pi]$, is used to evaluate the orientation of robot. Therefore, the actual orientation of robot can be described easily in $[0, 2\pi)$ interval in practice. The below relations map the orientation of robot to the aforementioned bound.

$$\begin{array}{l} \text{IF } \theta \ge 2\pi \\ \text{THEN } \theta = \theta - 2\pi \\ \text{ELSEIF } \theta < 0 \\ \text{THEN } \theta = \theta + 2\pi \\ \text{ELSE } \theta = \theta \end{array} \tag{11.a}$$

Alike this transformation can also be seen in Ref. [23]. According to Eq. (5) and mapping in Eq. (11.a), error of direction, θ_e, is limited in $(-2\pi,2\pi)$ interval. Since the error of orientation, θ_e, same as the orientation of robot, θ, is a periodic variable with 2π periodicity, it can be described in $[0, 2\pi)$ interval. This transformation can be conducted by the following relation:

$$\begin{array}{r} \text{IF } \theta_e < 0 \\ \text{THEN } \theta_e = \theta_e + 2\pi \\ \text{ELSE } \theta_e = \theta_e \end{array} \tag{11.b}$$

In order to find the suitable angular velocity, a Lyapunov-like function is defined in Eq. (12.a), which is lower bounded and while the configuration errors are in the domain of $-\infty < x_e < +\infty$, $-\infty < y_e < +\infty$ and $0 \le \theta_e < 2\pi$, the infimum of function is one.

$$V = \frac{1}{2}\left({x_e}^2 + {y_e}^2\right) + e^{\theta_e} \tag{12.a}$$

The derivative of the aforementioned function is defined in Eq. (12.b):

$$\dot{V} = \dot{x}_\mathrm{e}x_\mathrm{e} + \dot{y}_\mathrm{e}y_\mathrm{e} + \dot{\theta}_\mathrm{e}\mathrm{e}^{\theta_\mathrm{e}} \tag{12.b}$$

Considering Eqs. (7), (8), (9) and $u_\mathrm{c} = k_\mathrm{x}x_\mathrm{e}$, the derivative of the Lyapunov-like function can be written as Eq. (12.c),

$$\dot{V} = (y_\mathrm{e}\omega_\mathrm{c} - u_\mathrm{c})x_\mathrm{e} + (-x_\mathrm{e}\omega_\mathrm{c})y_\mathrm{e} - \omega_\mathrm{c}\mathrm{e}^{\theta_\mathrm{e}} = -u_\mathrm{c}x_\mathrm{e} - \omega_\mathrm{c}\mathrm{e}^{\theta_\mathrm{e}} = -k_\mathrm{x}{x_\mathrm{e}}^2 - \omega_\mathrm{c}\mathrm{e}^{\theta_\mathrm{e}} \tag{12.c}$$

If the angular velocity is selected as follows:

$$\omega_\mathrm{c} = k_\mathrm{y}{y_\mathrm{e}}^2(\mathrm{e}^{-\theta_\mathrm{e}}) + k_\theta{\theta_\mathrm{e}}^2(\mathrm{e}^{-\theta_\mathrm{e}}) \tag{13}$$

where $k_\mathrm{y},\ k_\theta > 0$, then $\dot{V}$ can be computed as Eq. 14,

$$\dot{V} = -k_\mathrm{x}{x_\mathrm{e}}^2 - k_\mathrm{y}{y_\mathrm{e}}^2 - k_\theta{\theta_\mathrm{e}}^2 \tag{14}$$

This implies that $\dot{V} \leq 0$ and considering the possible jump in θ_e, which has been explained in Eq. (11.b), it can be deduced that V is limited as follows:

$$1 \leq V < \left[\frac{1}{2}(x_e^2 + y_e^2) + e^{\theta_e}\right]_{x_e = x_e(0),\ y_e = y_e(0),\ \theta_e = 2\pi} \tag{15}$$

As a result, x_e and y_e are bounded and according to the constraints (11), which have been applied on θ_e, ω_c is also bounded. Therefore, the boundedness constraint on ω_c has been satisfied and consequently based on Eqs. (10), it can be concluded that $x_e \to 0$ and $y_e \to 0$. In addition, the following constraint can be obtained for the selected angular velocity.

$$\dot{\theta}_\mathrm{e} = -\omega_\mathrm{c} \leq 0 \tag{16}$$

Hence, according to (9), (11), (13) and (16), as long as $y_e \neq 0$ and $\theta_\mathrm{e} \neq 0$, θ_e goes from 2π to zero. As a conclusion, θ_e definitely can be located in some vicinity around zero. Finally, choosing control inputs as follows drives the robot to the desired configuration.

$$\begin{aligned} u_\mathrm{c} &= k_\mathrm{x}x_\mathrm{e} \\ \omega_\mathrm{c} &= k_\mathrm{y}{y_\mathrm{e}}^2(\mathrm{e}^{-\theta_\mathrm{e}}) + k_\theta{\theta_\mathrm{e}}^2(\mathrm{e}^{-\theta_\mathrm{e}}) \end{aligned} \tag{17}$$

Now the possible jump in orientation error and its influence on angular velocity will be investigated. Considering the transformation in Eq. (11.b) for θ_e, it is expected to have a single jump for the angular velocity in Eq. (17). The maximum jump will be occurred when $\theta_\mathrm{e} = 0$ is replaced by $\theta_\mathrm{e} = 2\pi$. The values of angular velocities for these two cases can be obtained as follows:

$$\omega_\mathrm{c}|_{\theta_\mathrm{e} = 2\pi} = k_\mathrm{y}{y_\mathrm{e}}^2(\mathrm{e}^{(-2\pi)}) + k_\theta(-2\pi)^2(\mathrm{e}^{(-2\pi)}) = k_\mathrm{y}{y_\mathrm{e}}^2(0.0019) + k_\theta(0.0737) \tag{18.a}$$

$$\omega_\mathrm{c}|_{\theta_\mathrm{e} = 0} = k_\mathrm{y}{y_\mathrm{e}}^2(\mathrm{e}^{(0)}) + k_\theta(0)^2(\mathrm{e}^{(0)}) = k_\mathrm{y}{y_\mathrm{e}}^2 \tag{18.b}$$

It is assumed that in the resetting instance, the value of y_e remains constant. The maximum value of jump will be obtained from the difference between these two velocities, (18.a) and (18.b), as follows.

$$\left|(\omega_\mathrm{c}|_{\theta_\mathrm{e} = 2\pi}) - (\omega_\mathrm{c}|_{\theta_\mathrm{e} = 0})\right| = \left|-k_\mathrm{y}{y_\mathrm{e}}^2(0.9981) + k_\theta(0.0737)\right| \tag{19}$$

It is worth mentioning that the described discontinuity in Eqs. (18) and (19) is general, which likely happened depending on the values of the selected gains (i.e. k_x, k_y and k_θ) and initial condition of robot. As a result, this jump may occur in some cases or may not happen in other cases based on the aforementioned factors. Furthermore, this jump originates theoretically from Brocket's condition, therefore, controller (17) is a non-smooth feedback control law. However, the value of the happened jump can be controlled by selecting appropriate values for k_y and k_θ based on the motors' characteristics.

Another set of control inputs which leads to posture stabilization is as,

$$\begin{aligned} u_\mathrm{c} &= k_\mathrm{x}x_\mathrm{e} \\ \omega_\mathrm{c} &= -k_\mathrm{y}{y_\mathrm{e}}^2(\mathrm{e}^{\theta_\mathrm{e}}) - k_\theta{\theta_\mathrm{e}}^2(\mathrm{e}^{\theta_\mathrm{e}}) \end{aligned} \tag{20}$$

To prove the stability of this controller, one can follow the similar procedure presented in Eq. (10.a) to Eq. (11.a). In this section, the error of orientation, θ_e, has been mapped to the interval $-2\pi < \theta_\mathrm{e} \leq 0$. This transformation can be done by adopting the following relations:

$$\begin{aligned} &\text{IF } \theta_\mathrm{e} > 0 \\ &\text{THEN } \theta_\mathrm{e} = \theta_\mathrm{e} - 2\pi \\ &\text{ELSE } \theta_\mathrm{e} = \theta_\mathrm{e} \end{aligned} \tag{21}$$

Considering the bounds $-\infty < x_e < +\infty$, $-\infty < y_e < +\infty$ and $-2\pi < \theta_\mathrm{e} \leq 0$, the new Lyapunov-like function can be considered as,

$$P = \frac{1}{2}\left({x_e}^2 + {y_e}^2\right) + e^{-\theta_e} \tag{22}$$

Selecting the control inputs (20) makes $\dot{P}$ to be nonpositive. This fact can be shown as,

$$\dot{P} = \dot{x}_e x_e + \dot{y}_e y_e - \dot{\theta}_e e^{-\theta_e} = (y_e\omega_c - u_c)x_e + (-x_e\omega_c)y_e + \omega_c e^{-\theta_e} = -u_c x_e + \omega_c e^{-\theta_e} = -k_x {x_e}^2 - k_y {y_e}^2 - k_\theta {\theta_e}^2 \tag{23.a}$$

Hence, the Lyapunov-like function P can be limited in the following domain.

$$1 \le P < \left[\frac{1}{2}\left(x_e^2 + y_e^2\right) + e^{-\theta_e}\right]_{x_e = x_e(0),\ y_e = y_e(0),\ \theta_e = -2\pi} \tag{23.b}$$

By adopting the similar argument presented for the previous controller, it can be deduced that x_e and y_e converge to zero. According to Eq. (20), the following constraint can be introduced for angular velocity.

$$\dot{\theta}_e = -\omega_c \ge 0 \tag{24}$$

Therefore, considering Eqs. (9), (20), (21) and (24), as far as $y_e \ne 0$ and $\theta_e \ne 0$, θ_e goes from -2π to zero. Consequently, θ_e can be located in the neighborhood of zero.

Comparing the suggested control laws with the previous researches, two important points can be concluded. Firstly, the usage of the suggested Lyapunov-like function obviously simplifies the steps, which should be passed to obtain the control laws and significantly reduces the complexity, which can be found in other works. Another advantage is the existence of two separate solutions that drive the robot to the desired posture. This subject will be clearly visible in the practical experiments and numerical simulations, in the next sections. Furthermore, according to angular velocities in control inputs in (17) and (20), we find out that one of the angular velocity is always clockwise while the other one is always counterclockwise. In other words, adopting controller (17) leads the robot to tend to the goal posture counterclockwise and on the other hand, controller (20) drives the robot clockwise to the desired posture. Therefore, robot travels a separate path in each controller and arrives to the goal posture.

4. Numerical simulations

To examine the presented control laws, firstly a simulation with different initial and final postures is tested by the control methods (17) and (20). Thereafter, two previously presented methods for the point stabilization in recent years, [21] and [23], are compared with our work in three different cases and the results are detailed. In the first scenario, we employ both of the control laws to drive the robot from the initial position and orientation of $\mathbf{p}_0 = \left[0, 4, \frac{\pi}{2}\right]$ to the final state of $\mathbf{p}_r = \left[5, -5, -\frac{\pi}{2}\right]$. In the first simulation we only focus on our methods and investigate the ability of them in the problem of point stabilization. The results contain the traveled path of the robot, angular and linear velocities and errors of the robot's position and orientation on the path. The sampling time is also set to 0.01 (s) for all computer simulations. These results are shown in Figs. 3–8 for both controllers. In order to clearly demonstration of the robot's path and orientation, the robot's traveled path for each controller is depicted separately. The controller's coefficients are listed in Table 1. It should be noted that in all of the conducted simulations in this section for all of the controllers, including the suggested controllers or the others', the associated controller gains are selected based on the trial and error approach. To this end, the controller gains were

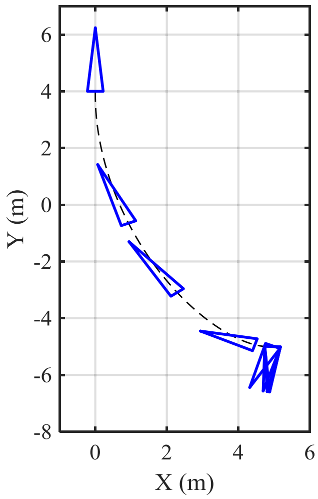


Fig. 3. The robot's path-controller (17). The robot is depicted by triangles.

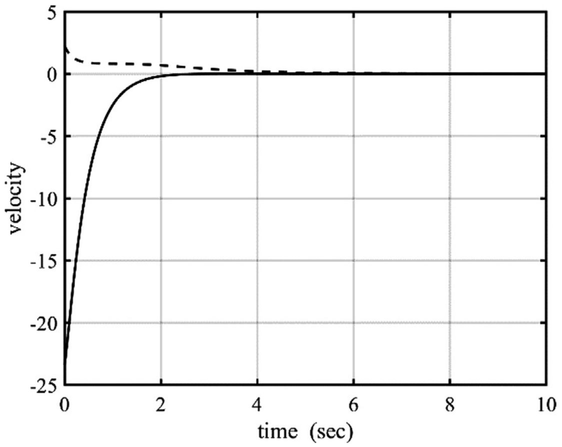


Fig. 4. Control signals of the controller (17). ω_c (- -) (rad/s). u_c (-) (m/s).

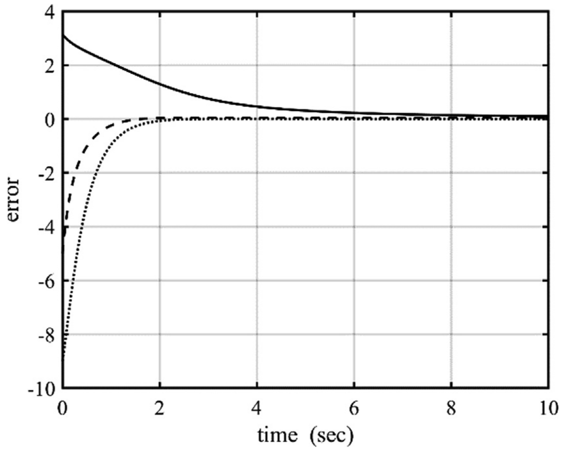


Fig. 5. Stabilization's error of the controller (17). x_e (...) (m). y_e(- -) (m). θ_e (-) (rad).

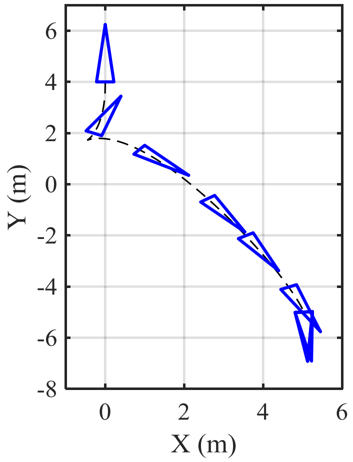


Fig. 6. The robot's path-controller (20). The robot is depicted by triangles.

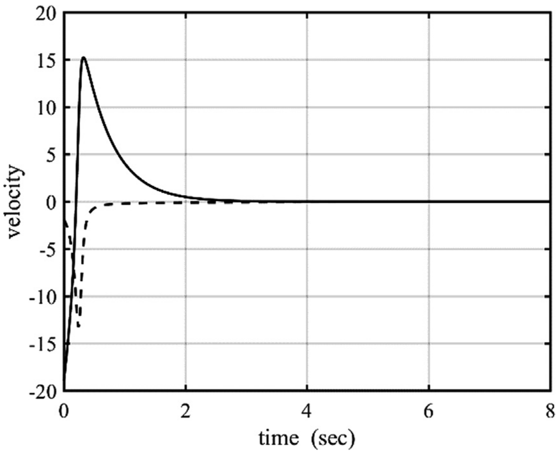


Fig. 7. Control signals of the controller (20). ω_c (- -) (rad/s). u_c (-) (m/s).

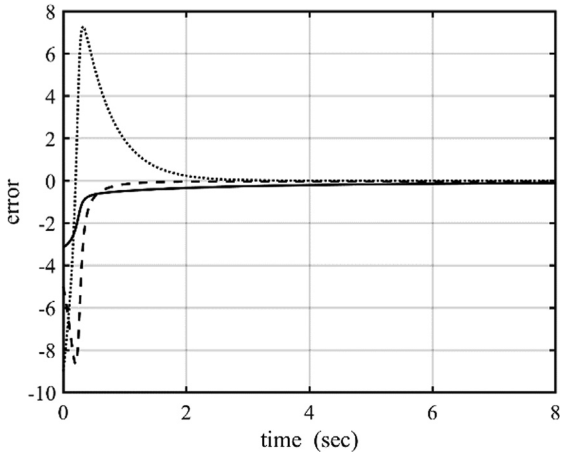


Fig. 8. Stabilization's error of the controller (20). x_e (...) (m). y_e (- -) (m). θ_e (-) (rad).

Table 1
The coefficients of the controllers in the first simulation scenario.

Controller	k_x	k_y	k_θ
(17)	2.6	1.5	1.5
(20)	2.1	1.25	1.25

varied randomly and the related simulations results were obtained. After that, the gains for each controller which resulted to good performance, based on the short traveled distance and low control effort, were selected and reported as the final gains of each controller. The control effort will be defined later on Eq. (25).

As can be observed in the figures, the existence of two controllers provides two distinct maneuvers that can be selected due to our expectations from controllers. This fact is important in some situation like presence of obstacles and cooperative mobile robots.

Another conclusion that can be obtained from the results is that in the controller (20), during approaching to the goal posture, the robot always rotates clockwise and in the other one, controller (17), robot rotates counterclockwise. Therefore, the angular velocity in (17) is positive and in (20) is negative. This information about the behavior of the robot in special situation like the object transportation might be useful. The variation of velocities in (17) is more smoothly than in (20), so lower forces exert on the platform of the robot.

In the following numerical simulations, three comparative cases are planned to evaluate the performance of the currently proposed control laws and control laws in Refs. [21,23]. In Ref. [23], a Vector-Field-Orientation feedback control (VFO) was presented for both trajectory tracking and point stabilization problems. The VFO controller travels an optimal path. However, the control law only presented a unique solution. In Ref. [21], a Finite Time (FT) controller is suggested that the singularity in orientation of robot in 90° and instability at the time of switching (for more information see Ref. [21]) are its main weaknesses.

In order to compare various methods, three measures including the control effort, which is defined in (25), traveled distance along the path and the time of operation are considered.

$$C.E \triangleq \int_{0}^{t} [|u_c(\tau)| + |\omega_c(\tau)|] d\tau \tag{25}$$

In the first comparative case, we employ four controllers in a scenario of point stabilization, which was planned in Ref. [23]. In this occasion, robot should be prepared to move from initial location of $\mathbf{p}_0 = [0, 2, 0]$ to final location of $\mathbf{p}_r = [2, -1, \frac{\pi}{4}]$. The coefficients of VFO controller, [23], are same as what have been selected in the paper and are listed in the first row of Table 2. The coefficients of FT controller, [21], are obtained by trial and error and are reported in the first row of Table 3. The coefficients of controllers (20) and (17) are also stated in Table 4. The traveled path of the robot in Cartesian coordinate and angular and linear velocities of control algorithms in comparative case I are depicted in Figs. 9–14. The performance mea-

Table 2
The coefficients of VFO controller [23] in comparative cases.

Comparative case	k_1	k_p	η
I	10	5	3
II	1	4	0.1
III	1	8	1

Table 3
The coefficients of FT controller [21] in comparative cases.

Comparative case	K_1	K_2	k_1	k_2	α_1	α_2	T
I	2	2	2	2	1.02	$\frac{1}{2}$	2.52
II	1.5	1.1	3	2	$\frac{2}{3}$	$\frac{1}{2}$	2
III	3	2	3	2	0.9	$\frac{1}{2}$	2.1

Table 4
The coefficients of controller (20) and (17) in comparative case I.

Controller	k_x	k_y	k_θ
(20)	6.9	25	29
(17)	3	1.4	13

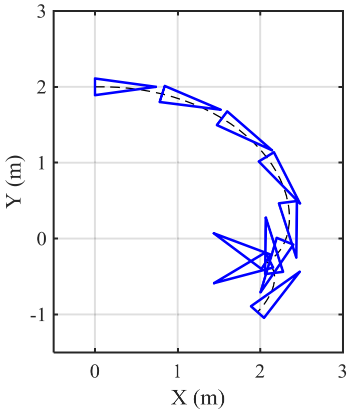


Fig. 9. The robot's path-controller (20). Comparative case I. The robot is depicted by triangles.

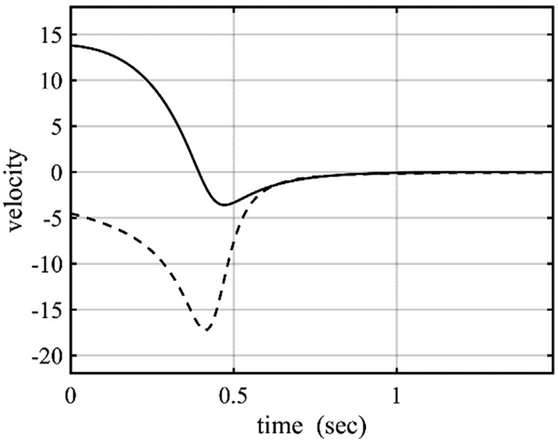


Fig. 10. Control signals of the controller (20) in comparative case I. ω_c (- -) (rad/s). u_c (-) (m/s).

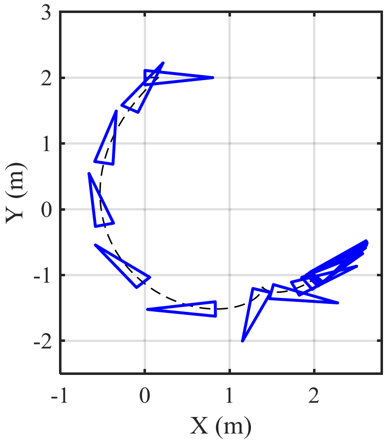


Fig. 11. The robot's path-controller (17). Comparative case I. The robot is depicted by triangles.

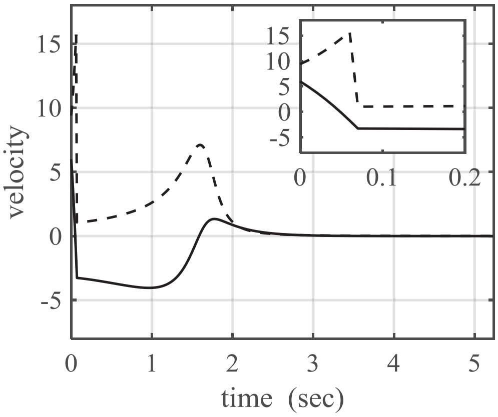


Fig. 12. Control signals of the controller (17) in comparative case I. ω_c (- -) (rad/s). u_c (-) (m/s).

sures are calculated and are shown in Table 5. The mentioned jump in the angular velocity of robot, which has been described in Eqs. (18) and (19), has been occurred in comparative case I for controller (17). The moment, which the jump happened, has been magnified in Fig. 12.

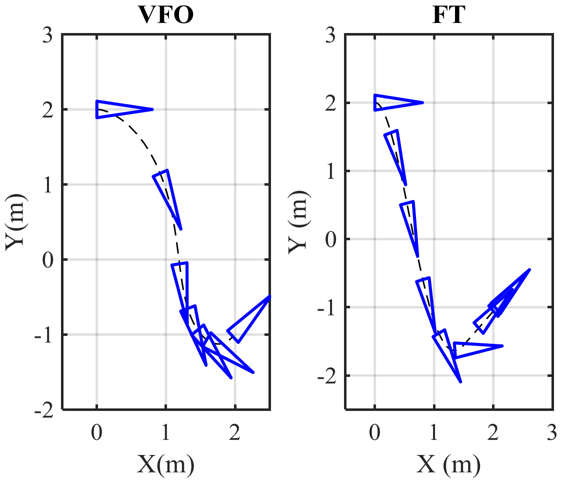


Fig. 13. The robot's path in comparative case I. The robot is depicted by triangles.

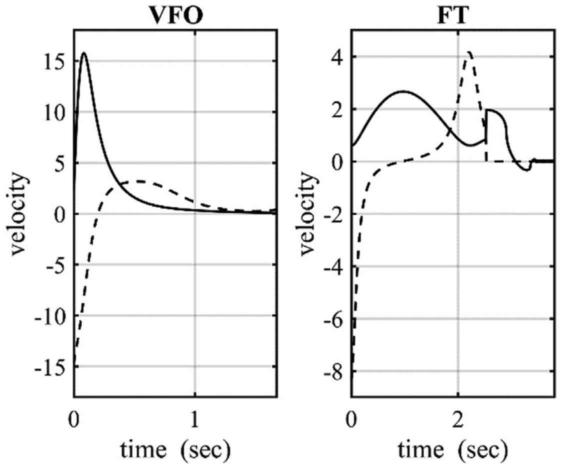


Fig. 14. Control signals in comparative case I. ω_c (- -) (rad/s). u_c (-) (m/s).

Table 5
Performance measures of the control algorithms in comparative simulation I.

Controller	Control effort	Traveled distance (m)	Time (s)
VFO	7.7404	4.083	1.68
FT	8.5589	5.0882	3.81
This work (20)	10.1677	4.6889	1.48
This work (17)	13.0940	6.0454	5.24

The results show that VFO and FT controllers have similar behavior. The obtained values demonstrate that the both controllers of this paper need more control effort and move longer distance than VFO and FT. In this simulation, VFO controller has the best performance. Among the controllers, controller (20) does this operation in less time. The simplicity and additional degree of freedom of the proposed controllers in this paper justify utilizing them despite a bit more control effort and traveled distance. Consider a condition in which the robot wants to be parked on the foresaid location and a hedge is located downward of the considered desired posture in the first comparative simulation. In this case, VFO and FT controllers lose their ability to carry out the mission. The dichotomy of our controllers assists to solve this issue. In this specific condition, we can employ controller (20).

In the second comparative simulation, four methods are examined in a scenario, which was designed in Ref. [21]. In this case, the robot is programmed to move from the initial configuration of $\mathbf{p}_0 = [4, 1, -0.2]$ to the origin. The coefficients of controllers and their performance indices are shown in Tables 2, 3, 6 and 7. The results also are depicted in Figs. 15–20.

Table 6
The coefficients of controller (20) and (17) in comparative case II.

Controller	k_x	k_y	k_θ
(20)	3.5	7	5
(17)	4	1.4	13

Table 7
Performance measures of control algorithms in comparative simulation II.

Controller	Control effort	Traveled distance (m)	Time (s)
VFO	14.5271	6.5393	3.23
FT	12.8195	12.1738	4.06
This work (20)	13.7620	7.6986	8.37
This work (17)	11.0650	4.6017	5.95

Table 8
The coefficients of controller (20) and (17) in comparative case III.

Controller	k_x	k_y	k_θ
(20)	2	6	6
(17)	0.6	2.9	2.4

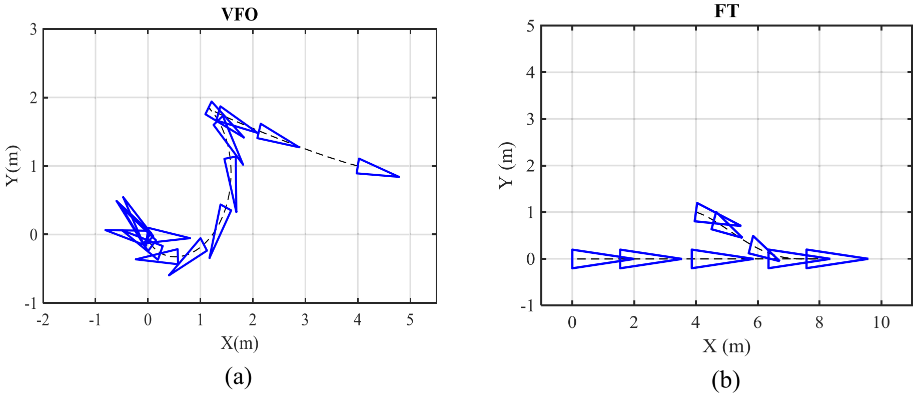


Fig. 15. The robot's path in comparative case II. The robot is depicted by triangles.

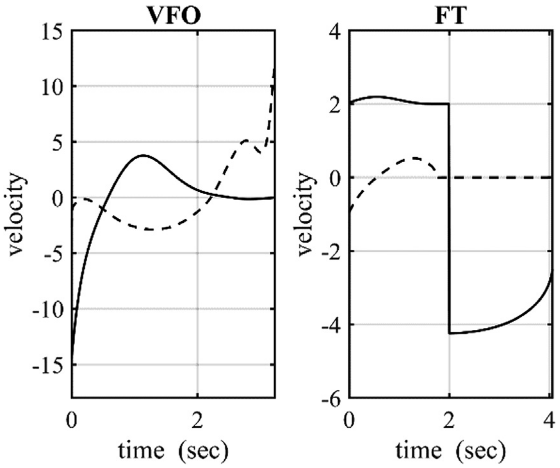


Fig. 16. Control signals in comparative case II. ω_c (- -) (rad/s). u_c (-) (m/s).

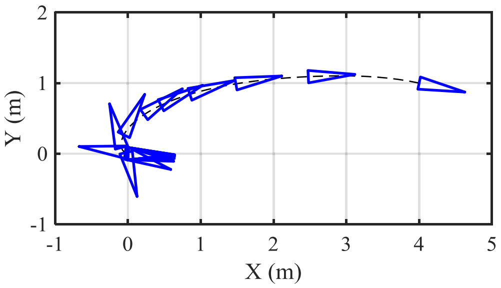


Fig. 17. The robot's path-controller (17). Comparative case II. The robot is depicted by triangles.

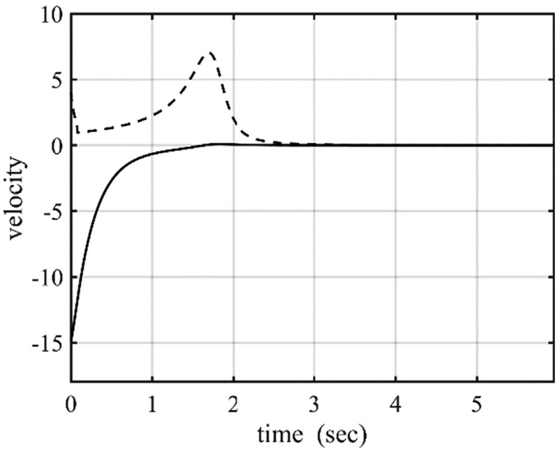


Fig. 18. Control signals of controller (17) in comparative case II. ω_c (- -) (rad/s). u_c (-) (m/s).

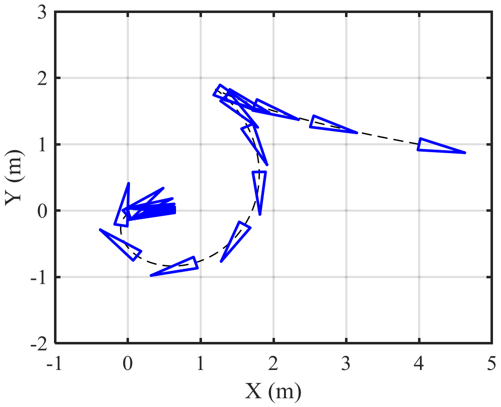


Fig. 19. The robot's path-controller (20). Comparative case II. The robot is depicted by triangles.

The performance of the controllers in second comparative case confirms the superiority of the controller (17) especially in the measures like amount of traveled path and control effort, although the control efforts are close to each other. In fact, the performance of the controller (17) is really near the optimal. The VFO controller does this simulation in minimum time but consumes more energy. The existence of discontinuity in linear velocity of robot in switching time is a critical point of FT controller. Two-step nature of FT controller for converging to the origin also extends its traveled path.

In the last comparative simulation, we want to drive the robot from $\mathbf{p}_0 = [0, 2, 0]$ to the final posture of $\mathbf{p}_r = [0, -2, 0]$. This case is also known as parallel parking. In this scenario, the performance measures only are calculated and reported in Table 9. The coefficients of controllers are listed in Tables 2, 3 and 8.

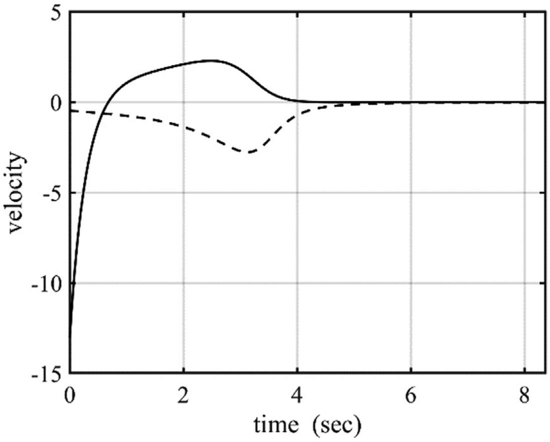


Fig. 20. Control signals of controller (20) in comparative case II. ω_c (- -) (rad/s). u_c (-) (m/s).

The performance indices for the third simulation represent the excellence of the suggested controllers in this paper against VFO and FT controllers. VFO controller has the minimum time and controller (20) has the minimum traveled distance and control effort. However, the aforementioned metrics of controller (17) have closed values to those of controller (20).

5. Experiment

In this section, we implement our controllers to a differentially driven robot to show the effectiveness of the proposed algorithms in practice (see Fig. 25). The basic parameters of the robot are explained in Table 10. The active wheels of the robot are driven by two 24-V dc motors. The control signals, which are fed to the robot, are angular velocities of the right and the left wheels. There are relations between these inputs and the linear and the angular velocities of robot as were mentioned earlier in Eq. (2). The position and orientation of the robot are obtained by using the camera of an android mobile phone and IP Webcam application. The mobile phone is located on the top of the test site. Image processing toolbox of the MATLAB software is used to analyze the acquired pictures, which are sent to the host computer by mobile phone. The test field is as a $2.2 \times 1.9\,(\mathrm{m}^2)$ rectangle area on a tile surface. During the experimental tests, the sampling time is about 0.1 (s).

The parallel parking is a practical problem in the domain of the wheeled vehicle. Hence, this challenging maneuver is selected to assess the ability of the proposed controllers. The initial condition is started from $\mathbf{p}_0 = [0, -0.5, 0]$ and the goal

Table 9
Performance measures of control algorithms in comparative simulation III.

Controller	Control effort	Traveled distance (m)	Time (s)
VFO	13.8588	7.6685	1.68
FT	19.4322	18.0394	3.37
This work (20)	10.8320	4.6480	4.11
This work (17)	11.6335	5.4498	10.93

Table 10
Basic parameters of differentially driven robot.

Parameter	Description	Value	Units
r	Driving wheel radius	0.05	[m]
b	Half of distance between robot actuated wheels	0.135	[m]

Table 11
The coefficients of the controllers in parallel parking.

Controller	k_x	k_y	k_θ
(17)	0.57	1.8	2.4
(20)	0.3	1.3	1.55

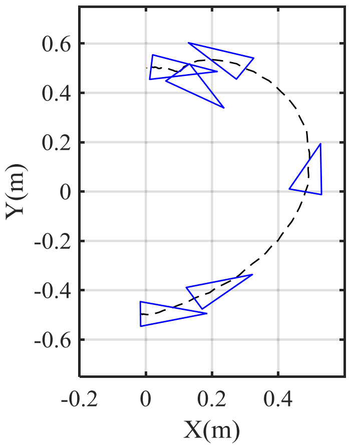


Fig. 21. Vehicle's path during parallel parking. Using controller (17).

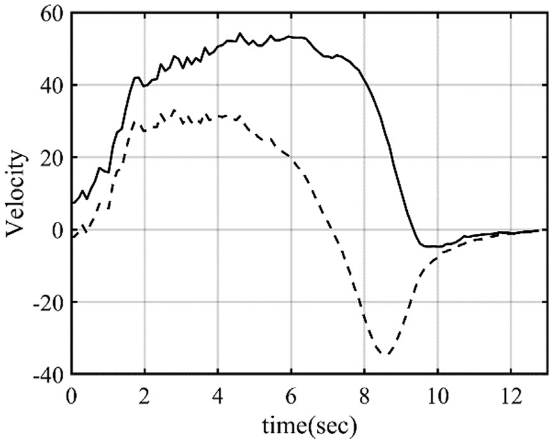


Fig. 22. Control signals of the controller (17) in parallel parking. α_r (-) (rpm). α_l (- -) (rpm).

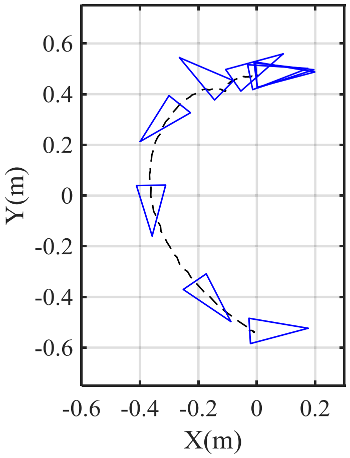


Fig. 23. Vehicle's path during parallel parking. Using controller (20).

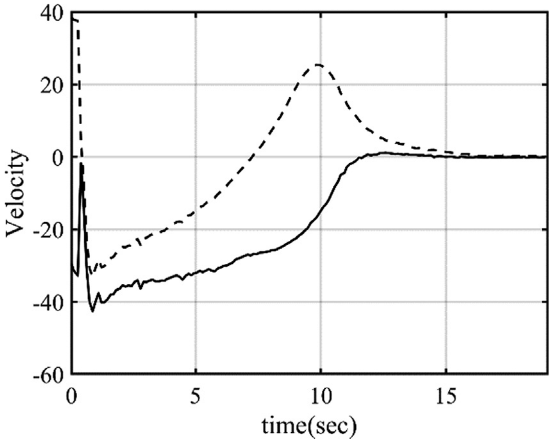


Fig. 24. Control signals of the controller (20) in parallel parking. α_r (-) (rpm). α_l (- -) (rpm).

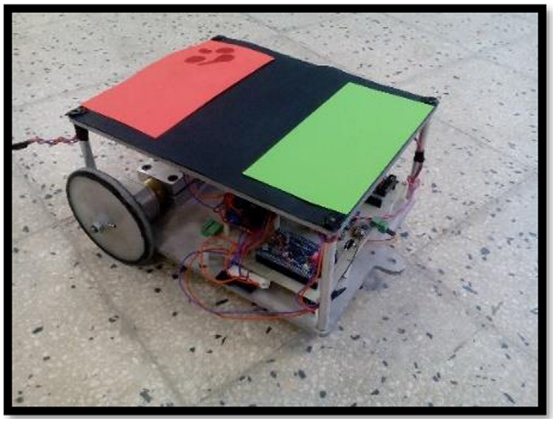


Fig. 25. Differentially driven robot.

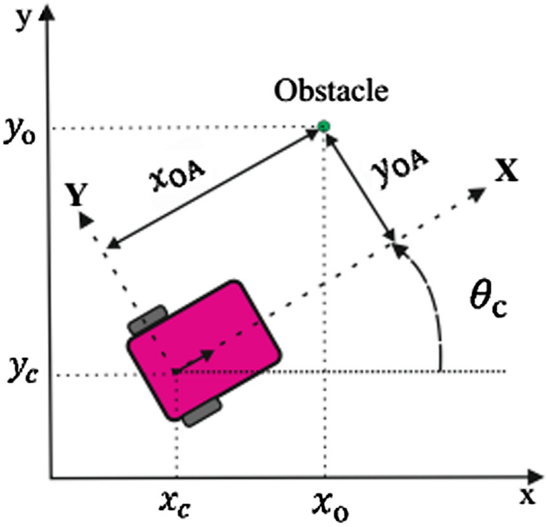


Fig. 26. Representation of obstacle in local coordinate.

posture is $\mathbf{p}_r = [0, 0.5, 0]$. The coefficients of the controllers are reported in Table 11. The results, including the traveled path of the robot and the angular velocities of the left and the right wheels of robot, are depicted in Figs. 21–24.

As seen, the path of the controller (17) is completely located on the right plane of the Cartesian coordinate and robot always rotates counterclockwise during the experiment. Therefore, the right wheel's angular velocity is always greater than the left wheel's angular velocity. However, the path of the controller (20) is located on the left plane of the Cartesian coordinate and the robot always rotates clockwise during the experiment. Hence, the left wheel's angular velocity is always greater than the right wheel's angular velocity. Consequently, based on the requirements and conditions each of the controllers can be selected. For example, consider a situation, in which we have some limitations that prevent the robot to travel in one of the right or the left plane. The duality of the controllers makes this mission possible in the parallel parking.

6. Stabilization with obstacle avoidance

In the current section, the application of the suggested method in the presence of obstacles is briefly explained for the controller (20). To avoid obstacles, some modifications should be applied on the controller. The similar modifications can be considered for controller (17).

The simple structure of the proposed controller causes its capability in real environment applications in the presence of obstacles. According to Eq. (20), the controller has two parts. One of the controller's command is the linear velocity of the robot. By this input, controller tries to reduce the longitudinal distance between the actual and the desired position of robot, x_e, which is described in the local coordinate system of the robot (see Fig. 2). If x_e has positive value then the corresponding linear velocity is positive too. This linear velocity causes the robot to get closer to the desired position. On the other hand, if x_e has negative value then the linear velocity is negative and this velocity also makes x_e smaller.

If the controller acts vice versa, the robot will get away from the desired position. It means that if x_e has positive value, then the controller considers $-k_x x_e$ as the linear velocity. It is clear that the linear velocity has negative value and this velocity makes x_e more and more. This scenario can be conducted for the obstacle avoidance. Therefore, if the longitudinal distance between the actual position of a robot and a point with opposite sign is fed to the robot as its linear velocity, then the robot will move away from that point.

Assume that an obstacle is located in the task space in the position of $\mathbf{p}_o = [x_o, y_o]$ in the Cartesian coordinate (see Fig. 26). Hence, in the first step, the position of the obstacle should be described in the robot attached local coordinate frame as Eq. 26,

$$\mathbf{p_{OA}} = \begin{bmatrix} x_{OA} \\ y_{OA} \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c \\ -\sin\theta_c & \cos\theta_c \end{bmatrix} \begin{bmatrix} x_o - x_c \\ y_o - y_c \end{bmatrix} \tag{26}$$

Now, if the term $-K_{avoid} x_{OA}$, where K_{avoid} is a positive constant, is added to the linear velocity in the control law, it will keep the robot away from the obstacle. However, as the robot gets closer to the obstacle, the dissuasive term "$-K_{avoid} x_{OA}$" is getting smaller. Therefore, the term $-K_{avoid} x_{OA}$ is replaced by $\frac{-K_{avoid}}{x_{OA}}$ to guarantee collision avoidance between the robot and the obstacle. Notice that as the robot gets closer to the obstacle the dissuasive term "$\frac{-K_{avoid}}{x_{OA}}$" is getting more and more. Consequently, the new controller can be written as Eq. 27,

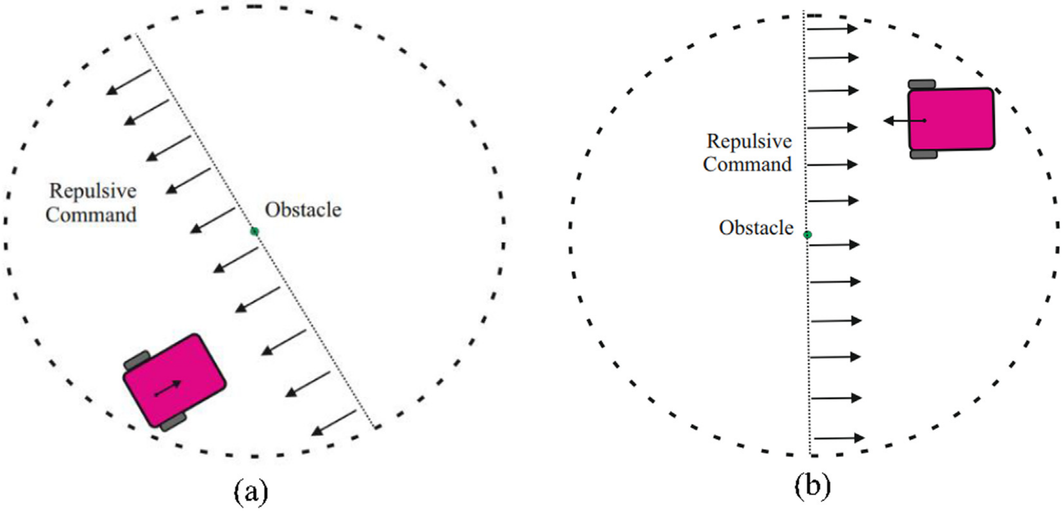


Fig. 27. Repulsive commands for robot in the influence zone. (a) Forward motion of robot. (b) Backward motion of robot.

$$\begin{aligned} u_c &= k_x x_e + \frac{-K_{\text{avoid}}}{x_{OA}} \\ \omega_c &= -k_y {y_e}^2 (e^{\theta_e}) - k_\theta {\theta_e}^2 (e^{\theta_e}) \end{aligned} \tag{27}$$

To not change the general rule of stabilization, a circular area is considered around each obstacle as its influence zone, which affects the control rule. In fact, if the robot enters into this zone, the control law will be changed from (20) to (27). It should be mentioned that the desired posture should be out of all obstacles' influence zone. By this assumption, the proof of stabilization of the control law is valid since if the robot enters the influence zone of obstacle, the new controller pushes it toward out of the influence zone. Then, the main controller drives the robot to the desired posture which is out of the influence zones. It is pointed out that out of the influence zones, the control rule is the same as before. Because the presented approach is intuitive, therefore the effectiveness of the aforementioned algorithm for obstacle avoidance is shown through computer simulations, similar to what has been done in Refs. [28,29]. The effect of the modified control method is depicted in Fig. 27. It is worth mentioning that the radius of influence for each obstacle is selected based on the obstacle dimensions and the robot performance.

The conditions in the last comparative case in Section 4 are chosen to examine the ability of the aforementioned method. Two static obstacles are located in the robot workspace. In each simulation, one of the obstacle's location is fixed and the

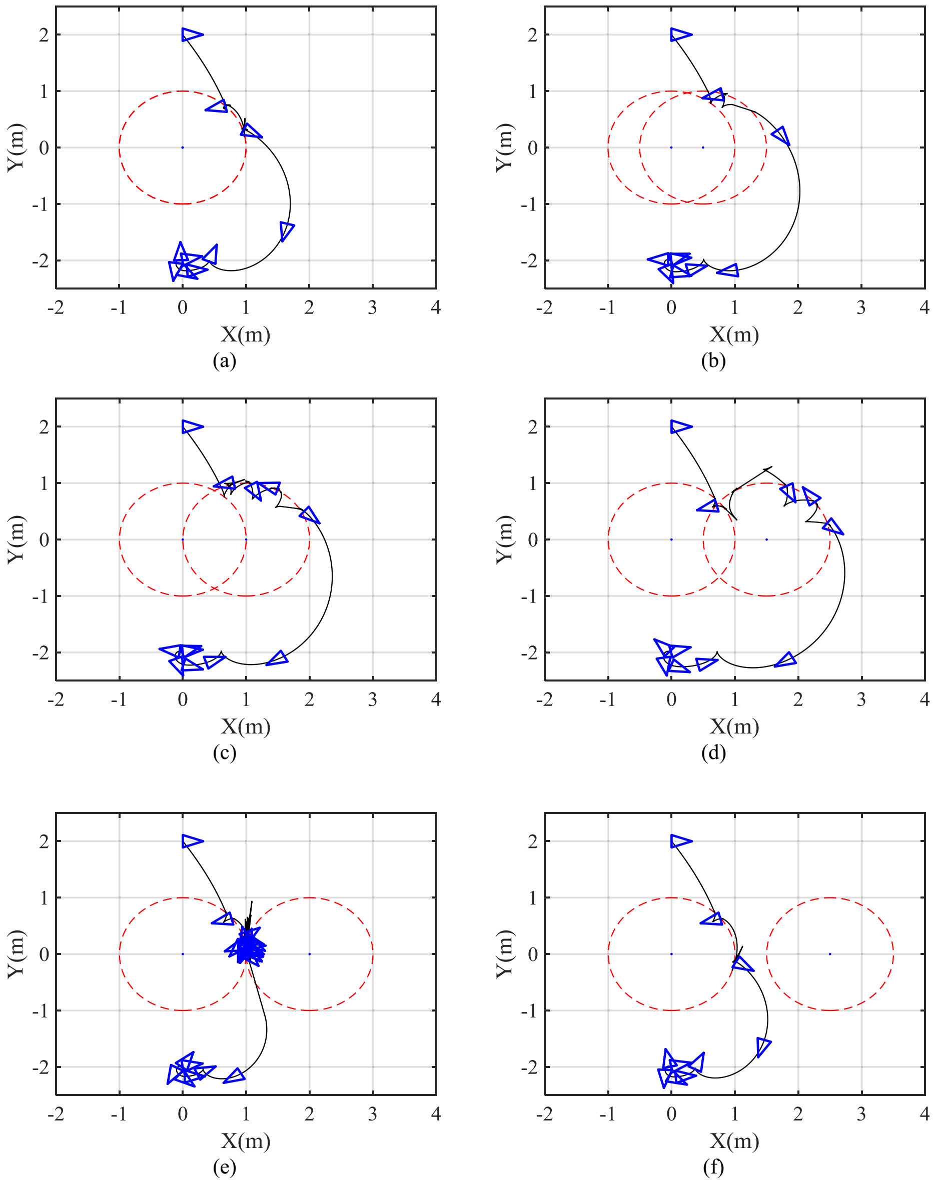


Fig. 28. Point stabilization in the presence of two obstacles. The robot is depicted by triangles.

position of the other obstacle is changed. The controller's gains are the same as Table 8. The obtained results show the effectiveness of presented algorithm (see Fig. 28). The influence zone is determined as a circle with radius of 1 (m). The value of K_{avoid} is set to 4. The robot always rotates clockwise, bypasses the obstacles and goes toward the desired target. As seen, the repulsive command keeps the robot away from the obstacles. The fixed direction of rotation of the robot does not allow it to get stuck in a loop. Notice that the suggested algorithm can also be used for a dynamic environment, which contains moving obstacles.

7. Conclusion

In this paper, two control strategies for point stabilization of the differentially driven wheeled mobile robot were presented. The obtained results proved the effectiveness of the suggested novel lower bounded functions and Barbalat's lemma. Duality of the control commands, the existence of two separate paths and good performance are the prominences of the control laws in comparison with the previous studies. Both of the proposed controllers do not have any singularity or limitation in Cartesian coordinate. The simple structure of control laws provides that the suggested controllers can be run by inexpensive processor since they have low computational burden. Modification of the developed control laws in the presence of obstacles were also introduced. Therefore, the proposed methods have ability to work in a more realistic situation. Future work includes a switching rule between the two control laws to obtain a more optimal path.

References

[1] J. Ni, W. Wang, J. Hu, C. Xiang, Relaxed static stability for four-wheel independently actuated ground vehicle, Mech. Syst. Sig. Process. 127 (2019) 35–49.

[2] Y. Li, L. Ding, Z. Zheng, Q. Yang, X. Zhao, G. Liu, A multi-mode real-time terrain parameter estimation method for wheeled motion control of mobile robots, Mech. Syst. Sig. Process. 104 (2018) 758–775.

[3] P. Kassaeiyan, B. Tarvirdizadeh, K. Alipour, Control of tractor-trailer wheeled robots considering self-collision effect and actuator saturation limitations, Mech. Syst. Sig. Process. 127 (2019) 388–411.

[4] K. Alipour, A.B. Robat, B. Tarvirdizadeh, Dynamics modeling and sliding mode control of tractor-trailer wheeled mobile robots subject to wheels slip, Mech. Mach. Theory 138 (2019) 16–37.

[5] R.W. Brockett, Asymptotic stability and feedback stabilization, Differ. Geom. Control Theory 27 (1) (1983) 181–191.

[6] C. Samson, Control of chained systems application to path following and time-varying point-stabilization of mobile robots, IEEE Trans. Autom. Control 40 (1) (1995) 64–77.

[7] C. Samson, Control of chained systems application to path following and time-varying point-stabilization of mobile robots, IEEE Trans. Autom. Control 42 (5) (1997) 698.

[8] P. Lucibello, G. Oriolo, Stabilization via iterative state steering with application to chained-form systems, in: Decision and Control, 1996., Proceedings of the 35th IEEE Conference on, IEEE, 1996, pp. 2614–2619.

[9] O.J. Sordalen, O. Egeland, Exponential stabilization of nonholonomic chained systems, IEEE Trans. Autom. Control 40 (1) (1995) 35–49.

[10] M. Aicardi, G. Casalino, A. Bicchi, A. Balestrino, Closed loop steering of unicycle like vehicles via Lyapunov techniques, IEEE Rob. Autom. Mag. 2 (1) (1995) 27–35.

[11] J. Guldner, V.I. Utkin, Stabilization of non-holonomic mobile robots using Lyapunov functions for navigation and sliding mode control, in: Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on, IEEE, 1994, pp. 2967–2972.

[12] R. Mukherjee, D. Chen, G. Song, Feedback control strategies for a nonholonomic mobile robot using a nonlinear oscillator, J. Rob. Syst. 16 (4) (1999) 237–248.

[13] K. Amar, S. Mohamed, Stabilized feedback control of unicycle mobile robots, Int. J. Adv. Rob. Syst. 10 (4) (2013) 187.

[14] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, A Stable Tracking Control Method for an Autonomous Mobile Robot, 1990.

[15] R. Fierro, F.L. Lewis, Control of a nonholomic mobile robot: backstepping kinematics into dynamics, J. Rob. Syst. 14 (3) (1997) 149–163.

[16] T. Dierks, S. Jagannathan, Control of nonholonomic mobile robot formations: backstepping kinematics into dynamics, in: Control Applications, 2007. CCA 2007. IEEE International Conference on, IEEE, 2007, pp. 94–99.

[17] S.X. Yang, A. Zhu, G. Yuan, M.Q.H. Meng, A bioinspired neurodynamics-based approach to tracking control of mobile robots, IEEE Trans. Ind. Electron. 59 (8) (2012) 3211–3220.

[18] C. Samson, Time-varying feedback stabilization of car-like wheeled mobile robots, Int. J. Rob. Res. 12 (1) (1993) 55–64.

[19] J.B. Pomet, Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift, Syst. Control Lett. 18 (2) (1992) 147–158.

[20] A. Astolfi, Discontinuous control of nonholonomic systems, Syst. Control Lett. 27 (1) (1996) 37–46.

[21] M. Thomas, B. Bandyopadhyay, L. Vachhani, Posture stabilization of unicycle mobile robot using finite time control techniques, IFAC-PapersOnLine 49 (1) (2016) 379–384.

[22] M. Guerra, D. Efimov, G. Zheng, W. Perruquetti, August). Finite-time supervisory stabilization for a class of nonholonomic mobile robots under input disturbances, 19th IFAC World Congress, 2014.

[23] M. Michalek, K. Kozlowski, Vector-Field-Orientation feedback control method for a differentially driven vehicle, IEEE Trans. Control Syst. Technol. 18 (1) (2010) 45–65.

[24] A. Zeiaee, R. Soltani-Zarrin, S. Jayasuriya, R. Langari, A uniform control for tracking and point stabilization of differential drive robots subject to hard input constraints pp. V001T04A005-V001T04A005, ASME 2015 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2015.

[25] Z. Cao, Y. Zhao, S. Wang, Trajectory tracking and point stabilization of noholonomic mobile robot, in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 1328–1333.

[26] A.K. Khalaji, S.A.A. Moosavian, Stabilization of a tractor-trailer wheeled robot, J. Mech. Sci. Technol. 30 (1) (2016) 421–428.

[27] J.J.E. Slotine, W. Li, Applied Nonlinear Control, Vol. 199, No. 1, Prentice Hall, Englewood Cliffs, NJ, 1991.

[28] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: Autonomous Robot Vehicles, Springer, New York, NY, 1986, pp. 396–404.

[29] C.J. Kim, D. Chwa, Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network, IEEE Trans. Fuzzy Syst. 23 (3) (2015) 677–687.