Research Article

State Dependent Riccati Equation (SDRE) controller design for moving obstacle avoidance in mobile robot

Motahareh Asgari[1] · Hoda Nokhbe Foghahayee[2]

Received: 1 June 2020 / Accepted: 7 October 2020 / Published online: 30 October 2020
© Springer Nature Switzerland AG 2020

Abstract

In this paper, a new navigating method for a non-holonomic wheeled moving robot in a dynamic environment with the moving obstacles is proposed. This method is indeed the combination of the nonlinear optimal controller based on State-Dependent Riccati Equation (SDRE) and an obstacle avoidance algorithm named artificial potential field (APF). The corresponding cost function of the SDRE is obtained of APF algorithm. APF algorithm forces the robot to approach the target as an attracting (low-potential) point and to get away from the obstacle as a repulsing (high-potential) point. This method calculates the best path from origin to destination which also implicitly guaranties the stability. The obtained path is the best according to the both amount of traveled distance and input energy. Moreover, this approach not only avoid both fixed and moving obstacle, but also create a smooth path in presence of them. Keeping the nonlinear structure of the system instead of eliminating them during the linearization process is the advantage of SDRE method. Here, the robot navigation is done in the presence of the three different movements of obstacle: (1) fixed speed, (2) fixed acceleration and (3) non-uniform circular. The represented simulation results indicate a suitable performance of the proposed algorithm.

Keywords Mobile robots · Optimal control · SDRE method · Moving obstacle avoidance · Path planning

1 Introduction

Todays, the robot control problem is an important challenging issue due to the increase in the applications of robots in industries. The robots have more superiority over human beings in working under dangerous and difficult conditions, higher iteration capability, and high accuracy and speed. The robot control contains both navigation and obstacle avoidance. In the robot control problem, the different aspects are considered such as: (a) the kinds of environment, targets and obstacles, (b) the algorithms of tracking and obstacles avoidance.

The robot environment is divided into two categories: (1) known environment (2) unknown environment. In the known environment, all required information about the environment is available. Therefore, the rout design is offline and globally optimal. However, in the unknown environment, the information is instantly given to the robot by the sensors, which causes the on-line locally optimal route design.

Obstacles are also divided into two categories: (1) fixed obstacles; and (2) moving obstacles. The number of obstacles and being animate (like human beings and animals) or inanimate (like objects or other co-worker robots) will be affective on the employing avoidance methods.

The route design and obstacle avoidance algorithms are selected based on the proper route (the shortest route with minimum time and cost) and the kind of environment

✉ Motahareh Asgari, Motahareh.asgari@yahoo.com; Hoda Nokhbe Foghahayee, nokhbeh@aut.ac.ir | [1]Department of Electrical Engineering, Islamic Azad University, Science and Research Branch, Tehran, Tehran, Iran. [2]Department of Electrical Engineering, Larestan University, Lar, Fars, Iran.

SN Applied Sciences (2020) 2:1928 | https://doi.org/10.1007/s42452-020-03649-3

and obstacles. There are several obstacle avoidance methods such as simple algorithm BUG [1] and its improvements [2, 3]; Vector Field Histogram(VFH) [4]; Follow the Gap Method (FGM) [5]; Artificial Potential Field (APF) [6] and some other methods which operate based on the notion of obstacle speed [7, 8].

In this paper, a non-holonomic wheeled moving robot in an unknown environment in the presence of moving obstacles is considered. An optimal path is then designed by employing the SDRE nonlinear optimal controller. The existence of nonlinearity of states and state dependent weight matrices in the controller structure make it useful as well as flexible. The main idea of this method is to factor nonlinear dynamics as multiplying the state vector. This is done by a matrix which is a function of the system states. In this method, at every moment, the Riccati Equation with the State Dependent Coefficients (SDCs) is solved [9].

This method was first introduced by Pearson [10] and was then developed by Wernli and Cook [11]. Cloutier et al. showed that the feedback law based on SDRE in cases of nonlinear optimal controls on the multi-variable infinite horizon was locally asymptotically stable [12, 13]. They also discussed about the using of weighting matrices on the control and on the state [14]. In [9], the formulating and applying of SDRE, non-uniqueness of parameters, the sufficient and necessary conditions for the existence of the solution, stability, optimality, capacities, limitations of this method and how to resolve such limitations are fully discussed. Some research has also been done on SDRE for non-affine systems [15]. One of the most important advantages of SDRE is its capability of combining with other algorithms such as obstacles avoiding algorithms. Therefore, it is used in different industries including auto-pilot [16], controlling spacecraft and satellites [17, 18], robotics [19–21] and etc.

There are various researches for navigating and avoiding obstacles for a moving robot [22, 23] or other vehicles [4, 24–28]. In the all these works, the moving obstacle avoidance problem was tackled as a path planning problem. In other words, the robot was considered as a position point without time variable and any dynamics. Therefore, neither internal and/or external stability of robot nor control input cost were not considered. In [29], the authors proposed an algorithm which deals with the obstacle avoidance using APF and selected traffic rules. The potential field method is improved by a decentralized market-based optimization (MBO) for multi-robot system. In [30, 31], the route design problem of the robot is deliberated on a control problem by considering the robot dynamics. In order to do that, an SDRE controller combined with an artificial potential function was proposed for fixed obstacle avoidance. Although the proposed algorithm works for the fixed obstacle, it makes instability in the present of moving obstacles. It is worth mentioning that there are a few papers which use the combination of SDRE and APF but with different applications [20, 21]. However, the position of APF term in [20, 21] is also similar to [30, 31]. It means that the APF term had been located in the cost function directly, while in our proposed method the inverse of APF term has been added to the cost function.

In our proposed method, a stabilizing optimal control is design by combination of the SDRE with a new cost function and the obstacle avoidance algorithm, APF. The two levels of control both moving obstacle avoidance and dynamic stabilization are done simultaneously due to the proposed controller.

In this paper, three kinds of obstacles have been considered including fixed speed, fixed acceleration and non-uniform circular moving obstacles. The paper is innovative in terms of dealing with moving obstacles and defining a new function for the cost function in the SDRE controller. The advantage of this method is in the situations where the obstacle is near to the target, whereas the other methods mentioning above cannot avoid from hitting the obstacle when it is near the target.

In the next section, SDRE method, APF algorithm and robot dynamic equations are summarized. The proposed algorithm which is the combination of SDRE controller and APF algorithm is applied to the robot dynamics in section three. In fourth and fifth sections the simulation results are represented and the paper is concluded, respectively.

2 Preliminaries

2.1 SDRE methodology

Consider the following affine nonlinear system.

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}) + B(\boldsymbol{x})\boldsymbol{u}(t), \quad \boldsymbol{x}_0 = \boldsymbol{x}(0) \tag{1}$$

where $\boldsymbol{x} \in R^n$ and $\boldsymbol{u} \in R^m$ are the state and input vector respectively. Assume that the function $f : R^n \to R^n$ *is* continuous nonlinear and $f(0) = 0$, $f(.) \in C^1(R^n)$ and $B : R^n \to R^{n\times m}, B(\boldsymbol{x}) \neq 0 \forall x$. In order to use SDRE method, it should write $f(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x}$ by employing a suitable factor of $f(\boldsymbol{x})$. Here $A : R^n \to R^{n\times n}$, and is non-unique for $n > 1$. By inserting $\boldsymbol{A}(\boldsymbol{x})$ in the Eq. (1), the nonlinear structure of the system becomes as

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{t}), \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{2}$$

$\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are called as matrices of SDC parameter-making.

The cost function which should be minimized is

$$J = \frac{1}{2}\int_{0}^{\infty}\left[\boldsymbol{x}^{\boldsymbol{T}}(t)\boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x}(t) + \boldsymbol{u}^{T}(t)\boldsymbol{R}(\boldsymbol{x})\boldsymbol{u}(t)\right]dt \tag{3}$$

where $\boldsymbol{R}_{\boldsymbol{m}\times\boldsymbol{m}} > 0,\boldsymbol{Q}_{\boldsymbol{n}\times\boldsymbol{n}} \geq 0$ are some diagonal and symmetric matrices. These matrices are called control and state matrices, respectively.

The structure of the control input which minimizes the cost function (3) is

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}(\boldsymbol{x})\boldsymbol{B}^{T}(\boldsymbol{x})P(\boldsymbol{x})\boldsymbol{x}(t) \tag{4}$$

where $P(\boldsymbol{x})$ is a symmetrical positive definite solution to the following algebraic Riccati equation [9].

$$P(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x}) + \boldsymbol{A}^{T}(\boldsymbol{x})P(\boldsymbol{x}) - P(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{x})\boldsymbol{B}^{T}(\boldsymbol{x})P(\boldsymbol{x}) + \boldsymbol{Q}(\boldsymbol{x}) = 0 \tag{5}$$

The necessary and sufficient condition for the existence of $P(\boldsymbol{x})$ is the controllability of the pair $\{\boldsymbol{A}(\boldsymbol{x}), \boldsymbol{B}(\boldsymbol{x})\}$ which needs to be taken into account in SDC parameter-making. System (2) is controllable if rank $\left(M_c(\boldsymbol{x})\right) = n,\forall\boldsymbol{x} \in R^n$ for the following matrix [9].

$$M_c(\boldsymbol{x}) = \left[\boldsymbol{B}(\boldsymbol{x})\middle|\boldsymbol{A}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\middle|\ldots\middle|\boldsymbol{A}^{n-1}\ (\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\right] \tag{6}$$

Furthermore, the condition of the controllability in these systems is a sufficient condition not necessary and sufficient one [32]. Thus, according to [9], "the presence or lack of controllability of this pair, {A(x),B(x)}, need not have any implication on the controllability of the original dynamics".

In this case since Riccati equation is dependent on the state variables, the solution $P(\boldsymbol{x})$ is dependent on state variables too.

2.2 APF algorithm

APF is an obstacle avoidance algorithm which works based on attractive field around the target and repulsive field around the obstacle. Thus, the target has a positive force to attract the robot and the obstacle has a negative force to repel it. Finally, direction is done by result of all forces toward the target and avoiding from obstacle. Therefore, the artificial potential field, $U(q)$, and the artificial force, $F(q)$ are obtained as follows

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{7}$$

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = F_{att}(q) + F_{rep}(q) \tag{8}$$

where $q = (\mathrm{x},\mathrm{y})^{\mathrm{T}}$ is the position of robot, $U_{att}(q)$, $U_{rep}(q)$ are the attractive and repulsive potential field and $F_{att}(q)$, $F_{rep}(q)$ are the attractive and repulsive forces respectively [33]. Now in order to attract the robot toward the goal, the attractive potential field and force is computed as:

$$U_{att} = \frac{1}{2}k\left(q - q_g\right)^2 = \frac{1}{2}k\rho_{goal}^2(q) \tag{9}$$

$$F_{att}(q) = -\nabla U_{att}(q) = -\frac{1}{2}k\nabla\rho_{goal}^2(q) = -k\left(q - q_{goal}\right) \tag{10}$$

where k is a positive coefficient, $q_g = \left(x_g, y_g\right)^T$ is the position of the target and $\rho_{goal}(q) = ||q - q_g||$ [33]. Similarly, in order to prevent the robot from hitting the obstacle, the repulsive potential field and force are calculated as:

$$U_{rep}(q) = \left\{\begin{array}{ll} 0, & \rho(q) \geq \rho_0 \\ \frac{1}{2}\mu\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \rho(q) \leq \rho_0 \end{array}\right\} \tag{11}$$

$$F_{rep}(q) = -\nabla U_{rep}(q) = \left\{\begin{array}{ll} 0, & \rho(q) \geq \rho_0 \\ \mu\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\left(\frac{1}{\rho^2(q)}\right)\nabla\rho(q), & \rho(q) \leq \rho_0 \end{array}\right\} \tag{12}$$

where $q_c = \left(x_c, y_c\right)$ is the position of obstacle, μ is the positive scaling factor, and ρ_0 is the largest distance between robot and obstacle, $\rho(q) = ||q - q_c||$, and $\nabla\rho(q) = \frac{q-q_c}{||q-q_c||}$ [33].

In this way, the robot which is affected by artificial potential field can find the safe path toward the target without collision with the obstacle. The safe path is defined based on the sum of all repulsive and attractive forces in the artificial field.

2.3 Robot dynamics

The kinematic model of the robot is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ v/L(\tan(\varphi)) \\ a \\ \omega \end{bmatrix} \tag{13}$$

where x and y are the positions of the robot in a two-dimensional $x - y$ plane, θ is the movement angle against the horizontal axis, v is the linear speed of the robot, φ is the angle between wheels and the movement direction (steering angle) and ω, a are the control variables [30]. Moreover, the constant L shows the distance between the front wheels of the moving robot and its back wheels [34] (also see Fig. 1).

In the first structure of the underlying system v and φ are the input variables as are given below

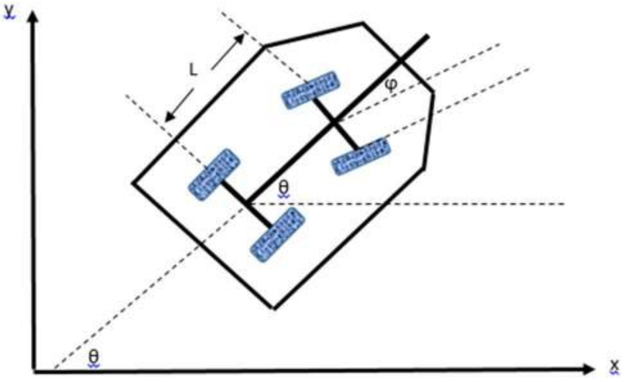

Fig. 1 The kinematic model of a wheeled moving robot [34]

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} v \\ \varphi \end{bmatrix} \tag{14}$$

In such a structure, sudden changes in φ and v may practically be impossible due to physical constraints of some robots. Therefore, in order to smooth the parameters φ and v, two new control inputs ω, a are defined. In this way, φ and v are also regarded as state variables. Thus, the "acceleration" variable which is made of force turns into a control variable, while the parameter of "speed" is taken as a new input to the system.

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ \theta \\ v \\ \varphi \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} a \\ \omega \end{bmatrix} \tag{15}$$

As a result, in the end, the robot movement equations will be in the form of (13) and a and ω are labeled as linear acceleration and angular speed, respectively.

3 Proposed SDRE-APF algorithm

In order to apply the SDRE controller on the robot dynamics, there is a need to determine the matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ of that. Notice that the SDC parameterization process should be done in a way that the pair $\{\boldsymbol{A}(\boldsymbol{x}), \boldsymbol{B}(\boldsymbol{x})\}$ becomes controllable [9]. For this purpose, the following procedure is done.

Since $\boldsymbol{A}(\boldsymbol{x})$ is obtained from mathematical factor, it will not be unique for $n > 1$ [9]. The underlying system $\dot{x} = f(x)$, can be rewritten as

$$\dot{x} = f(x) = \alpha f(x) + (1 - \alpha) f(x) = A(x, \alpha) x \tag{16}$$

where $A(x, \alpha) = \alpha A_1(x) + (1 - \alpha) A_2(x)$ and $\alpha = \left[\alpha_1 \ \ldots \ \alpha_n \right]^T$. This procedure provides an additional degree of freedom.

Dynamic Eq. (13) based on (16) becomes

$$\dot{x} = v \cos\theta = \alpha_1 \left[v \left(\frac{\cos\theta}{\theta} \right) \right] \theta + \left(1 - \alpha_1 \right) (\cos\theta) v$$

and then by adding and subtracting $\alpha_1 v$ and using some mathematical manipulation becomes

$$\begin{aligned} \dot{x} = v \cos\theta &= \alpha_1 \left(v \frac{\cos\theta}{\theta} \right) \theta - \alpha_1 v + \left[\left(1 - \alpha_1 \right) (v \cos\theta) + \alpha_1 v \right] \\ &= \alpha_1 v \left[\frac{\cos\theta}{\theta} - \frac{1}{\theta} \right] \theta + \left[\left(1 - \alpha_1 \right) \cos\theta - \left(1 - \alpha_1 \right) + 1 \right] v \\ &= \alpha_1 v \left[\frac{\cos\theta - 1}{\theta} \right] \theta + \left[\left(1 - \alpha_1 \right) (\cos\theta - 1) + 1 \right] v \end{aligned}$$

$$\dot{y} = v \sin\theta = \alpha_2 \left[v \left(\frac{\sin\theta}{\theta} \right) \right] \theta + \left(1 - \alpha_2 \right) (\sin\theta) v \tag{17}$$

$$\dot{\theta} = \frac{v}{l} \tan\varphi = \alpha_3 \left[\frac{v}{l} \left(\frac{\tan\varphi}{\varphi} \right) \right] \varphi + \left(1 - \alpha_3 \right) \left(\frac{\tan\varphi}{l} \right) v$$

$$\dot{v} = a$$

$$\dot{\varphi} = \omega$$

Therefore, matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 & \alpha_{13} & \alpha_{14} & 0 \\ 0 & 0 & \alpha_{23} & \alpha_{24} & 0 \\ 0 & 0 & 0 & \alpha_{34} & \alpha_{35} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{18}$$

where

$$\begin{aligned} &\alpha_{13} = \frac{\alpha_1 v(\cos\theta - 1)}{\theta}, \quad \alpha_{14} = \left(1 - \alpha_1 \right) (\cos\theta - 1) + 1 \\ &\alpha_{23} = \alpha_2 v \left(\frac{\sin\theta}{\theta} \right), \quad \alpha_{24} = \left(1 - \alpha_2 \right) \sin\theta \\ &\alpha_{34} = \left(1 - \alpha_3 \right) \frac{\tan\varphi}{l}, \quad \alpha_{35} = \alpha_3 \left(\frac{v}{l} \right) \left(\frac{\tan\varphi}{\varphi} \right) \end{aligned} \tag{19}$$

The controllability matrix $M_c(x)$ of the system matrices $(A(x), B(x))$ is

$$M_c(x) = \begin{bmatrix} 0 & 0 & \alpha_{14} & 0 & \alpha_{13}\alpha_{34} & \alpha_{13}\alpha_{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{24} & 0 & \alpha_{23}\alpha_{34} & \alpha_{23}\alpha_{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{34} & \alpha_{35} & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

The matrix $M_c(x)$ is full rank for every arbitrary $\alpha_1 \neq \alpha_2$ and $\alpha_3 \neq 0$. According to these conditions the underlying system is fully controllable.

Since the robot should reach a target $\left(x_d, y_d\right)$, the error signals are defined as

$$e_x = x - x_d, e_y = y - y_d, e_\theta = \theta, e_v = v, e_{\emptyset} = \varphi \tag{21}$$

Therefore the error system becomes

$$\begin{aligned}
\dot{e}_x &= e_v \cos e_\theta = \alpha_1 \left[e_v \left(\frac{\cos e_\theta}{e_\theta} \right) \right] e_\theta + \left(1 - \alpha_1\right) \left(\cos e_\theta\right) e_v \\
\dot{e}_y &= e_v \sin e_\theta = \alpha_2 \left[e_v \left(\frac{\text{sine}_\theta}{e_\theta} \right) \right] e_\theta + \left(1 - \alpha_2\right) \left(\sin e_\theta\right) e_v \\
\dot{e}_\theta &= \frac{e_v}{l} \tan e_\varphi = \alpha_3 \left[\frac{e_v}{l} \left(\frac{\tan e_\varphi}{e_\varphi} \right) \right] e_\varphi + \left(1 - \alpha_3\right) \left(\frac{\tan e_\varphi}{l} \right) e_v \\
\dot{e}_v &= a \\
\dot{e}_\varphi &= \omega
\end{aligned} \tag{22}$$

Fortunately the matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are not changed. Furthermore, the cost function (3) is rewritten as:

$$J = 1/2 \int_0^\infty \left[\boldsymbol{e}^{\boldsymbol{T}}(t) \boldsymbol{Q}(\boldsymbol{e}) \boldsymbol{e}(t) + \boldsymbol{u}^T(t) \boldsymbol{R}(\boldsymbol{e}) \boldsymbol{u}(t) \right] dt \tag{23}$$

where matrices $\boldsymbol{R}(\boldsymbol{e})$ and $\boldsymbol{Q}(\boldsymbol{e})\left(\boldsymbol{Q}(\boldsymbol{e}) = diag\left(Q_x, Q_y, Q_\theta, Q_v, Q_\varphi\right)\right)$ are defined based on the *APF* algorithm. In this algorithm, potential function is defined based on the positions of the target and the obstacles. The target is like a low-potential valley with an attractive force, and the obstacle is like a high-potential mountain peak with a force of repulsion [35]. The quantities of these forces of attraction and repulsion are proportionate to the distance between the target and the obstacle. Therefore, *APF* leads the robot to approach the target or get away from the obstacles respectively (see Fig. 2).

According to the APF algorithm, when the robot is moving toward the target, the potential function should be decreased and when it is approaching the obstacle, it should be increased. Consequently, a suitable candidate for the potential function is

$$PPF = w \left(\frac{\left(e_x\right)^2 + \left(e_y\right)^2}{r_{ob} + s\left(\left(x - x_{ob}\right)^2 + \left(y - y_{ob}\right)^2 \right)} \right) \tag{24}$$

where (x, y) and $\left(x_{ob}, y_{ob}\right)$ indicate the current position of the robot and the moving obstacle respectively. Here, s is the reliability coefficient, r_{ob} is appropriated distance from obstacle and w represents the weighting coefficient to realize the distance from the robot to the obstacle whenever it passes. Knowing that the objective of an optimal control such as SDRE controller is to decrease a cost function, the inverse of defined obstacle avoidance term (24) is added to the cost function (23) as follows

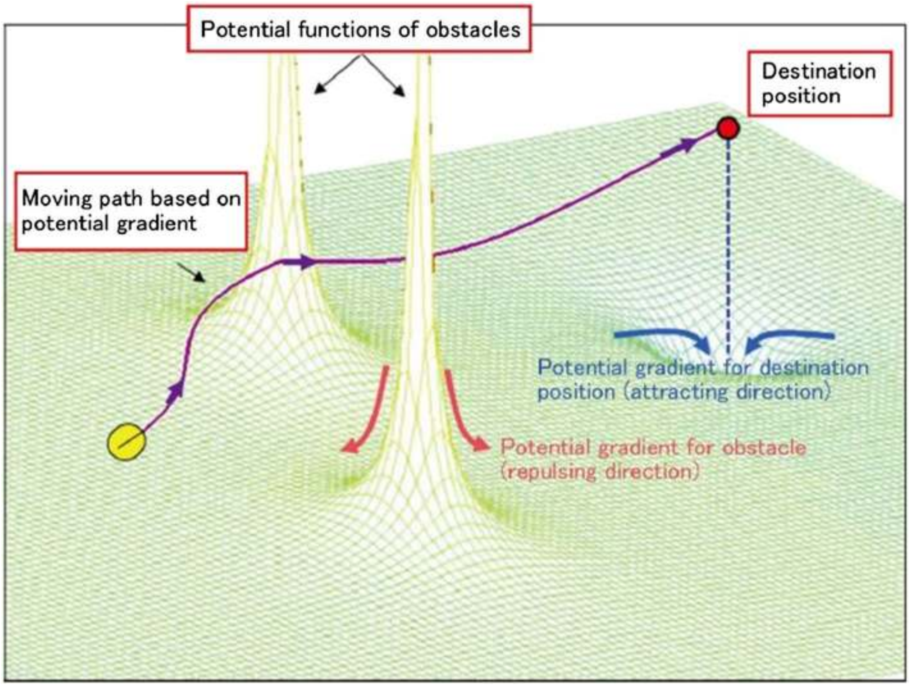
Potential functions of obstacles
Destination position
Moving path based on potential gradient
Potential gradient for destination position (attracting direction)
Potential gradient for obstacle (repulsing direction)

Fig. 2 An instance of route production based on the Potential Field Methodology [35]

$$J = 1/2 \int_0^\infty \left(\left[\boldsymbol{e}^{\boldsymbol{T}}(t)\boldsymbol{Q}(\boldsymbol{e})\boldsymbol{e}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}(\boldsymbol{e})\boldsymbol{u}(t) \right] + PPF^{-1} \right) dt \tag{25}$$

Therefore, the matrix Q becomes as:

$$\boldsymbol{Q} = diag\left(Q_x + \frac{1}{e_x^2 \times PPF}, Q_y + \frac{1}{e_y^2 \times PPF}, Q_v, Q_\theta, Q_\varphi \right) \tag{26}$$

Notice that the inverse of function PPF increase if the robot approaches the target or gets away from the obstacle and causes the system to continue its behavior in that manner. Notice that the increment in the inverse of PPF is equal to increase the value of matrix $\boldsymbol{Q}$ than matrix $\boldsymbol{R}$. This in turn allows the input signal to increase. Here are four critical situations for robot control based on robot position. The robot could be:

1. Near the target, far from the obstacle.
2. Near the both target and obstacle.
3. Far from both target and obstacle.
4. Far from the target and near the obstacle.

It can be shown that the PPF function in our proposed method works if the robot approaches the target. In other words, when the robot is far from the target, PPF term has a small effect on the control signal than Q_x and Q_y like case 3.

Consequently, according to the new definitions, the SDRE controller produces a control signal that minimizes the cost function (25). According to the cost function (25) and matrix Q in (26) the robot's approaching the target results in a decrease of the costs while approaching the obstacle leads to an increase of the costs. In other words, the robot starts moving from the start point towards the target; when approaches the obstacle, the '*PPF*' potential function increases and then causes decrease of matrix Q. This allows the robot to increases its distance from the target and so allows itself to avoid the obstacle by circling round it. In such way, the navigation is done without hitting the obstacle.

Remark 1 In [30], a combination of SDRE controller and APF has already been shown. There, the potential function, matrix Q and cost function are in the forms of (27)–(29) respectively.

$$obs = w\left(\frac{(x - x_d)^2 + (y - y_d)^2}{r_{ob} + \sqrt{\left((x - x_{ob})^2 + (y - y_{ob})^2 \right)}} \right) \tag{27}$$

$$\boldsymbol{Q} = diag\left(\frac{1}{e_x^2}(obs), \frac{1}{e_y^2}(obs), 1, 1, 1 \right) \tag{28}$$

$$J = 1/2 \int_0^\infty \left[\boldsymbol{e}^{\boldsymbol{T}}(t)\boldsymbol{Q}(\boldsymbol{e})\boldsymbol{e}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}(\boldsymbol{e})\boldsymbol{u}(t) \right] dt \tag{29}$$

According to (27) when the robot approaches the obstacle, the potential function '*obs*' increases which increase the matrix Q given in (28). Note that the increasing of Q makes stronger forces to decrease the error. Consequently, it makes the robot approaches the target, while it may not avoid the obstacle which is in its neighborhood. Moreover, in this research, coefficient "w" is initialized as 1000. The high value of w leads to a high-gain control which obviously is not applicable because of the huge amount of control cost. According to the simulations which are done by the authors, cost function (29) with (28) is not only capable of route design in the presence of moving obstacles but also makes instability in some cases. Based on Eqs. (27) and (28) when the obstacle moves near the target, the huge amount of Q (because of "obs" term) makes a huge peak on the control input which gets robot away from the target. This behavior often causes instability too. It is clear that this method not only cannot work in the presence of the moving obstacles but also cannot have a good performance when the fixed obstacle is near the target.

Remark 2 The defined cost function in [31] is too similar to [30]. However, the potential function, "OBS", is a little different.

$$R_{at} = \sqrt{(x - x_d)^2 + (y - y_d)^2}, \quad R_{rep} = \sqrt{(x - x_{ob})^2 + (y - y_{ob})^2}, \quad obs_a = m\left(\frac{R_{at}^z}{R_{rep}^z + 0.1} \right),$$
$$obs_b = \left(\frac{R_{at}^{(z-1)}}{R_{rep}^z + 0.1} \right), \quad OBS = obs_a + obs_b, \quad m = 5, z = 2 \tag{30}$$

$$\boldsymbol{Q} = diag\left(\frac{1}{e_x^2}(obs), \frac{1}{e_y^2}(obs), 1, 1, 1\right) \tag{31}$$

$$J = 1/2\int_0^{\infty}\left[\boldsymbol{e}^{\boldsymbol{T}}(t)\boldsymbol{Q}(\boldsymbol{e})\boldsymbol{e}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}(\boldsymbol{e})\boldsymbol{u}(t)\right]dt \tag{32}$$

The difference created in the potential function is not effective in solving the problem described in remark 1. Although the authors of [31] have claimed that their proposed method can avoid the fixed obstacle with smaller "w" than that of [30], the simulations do not confirm that. Actually, this cost function cannot work without high-gain control effectively especially when the obstacle is near the target.

When case 2 is occurred, Q function becomes big. The big Q allows the input signal to increase in order to decrease the error. However, the obstacle is near to the target, and the robot should move slowly with accuracy near the obstacle. In case 3, when the robot is far from both the target and obstacle, the input signal could increase in order to tend to the target, While here the small Q causes slow movement toward the target.

Remark 3 In comparison with the existing studies [30, 31], the proposed methodology has the following main advantages:

1. It is applicable to avoid from moving obstacles, whereas the existing studies were only proposed to avoid from fixed obstacles.
2. It works in the presence of fixed or moving obstacles which are near the target. By contrast, the existing methods cannot avoid this types of obstacles and they lead to instability or collision with obstacles.
3. The produced input control signal in the proposed method is small as well as smooth to apply to practical cases.

For the better compare see the Table 1 and "Appendix 1".

4 Simulations

In this section, the simulation method is presented within the context of MATLAB software. In order to explore the performance of this method, three different kinds of movements for the obstacles are considered:

1. Fixed speed.
2. Fixed acceleration.
3. Non-uniform circular.

Due to the mobility of the obstacle, two different views by a three dimensional rotation of the graph are provided for each case so that the two dimensions of the robot's avoidance of the obstacle can be shown in order to make comparisons.

It is clear that a major advantage of the SDRE controller is not to ignore certain important states due to the no need for the linearization of a nonlinear system. So as to see this advantage better, there are the comparison between our proposed method and the LQR controller for all three movements, one of which is indicated in the simulations. The results for all three movements were the same. LQR controller is designed after linearization process around the working point $\left(\theta = 0.7\,\mathrm{rad}, V = 1.42\,\frac{\mathrm{m}}{\mathrm{s}}, \varphi = 0.33\,\mathrm{rad}\right)$. To do so, see Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 24, 25 and 26.

As it is obvious in the figures, in the system with the control of the SDRE controller, there has been no accident between the robot and the obstacle. In this case, whenever the robot approaches the obstacle, it immediately gets away from it and avoids it through increasing the cost. By using the LQR controller, the robot's navigation is also done without any collisions. However, there is a great difference between SDRE controller and LQR controller. The LQR is capable of being applied in a limited number of working points. Therefore, changes of the working point, which the system has been linearized, make the LQR controller inefficient.

Table 1 Display the different results by using three method in presence of fixed obstacle (for case 2)

Used method	Type of obstacle	Hit	Approach time (s)	Travelled distance	Smooth response	Admissible input	Maximum of input variable
Proposed method by [30]	Fixed	Yes	2.5	11.0775	No	No	67.6391
Proposed method by [31]	Fixed	No	12	9.7525	No	No	5.8929
Proposed method by this paper	Fixed	No	5	10.6445	Yes	Yes	14.7482

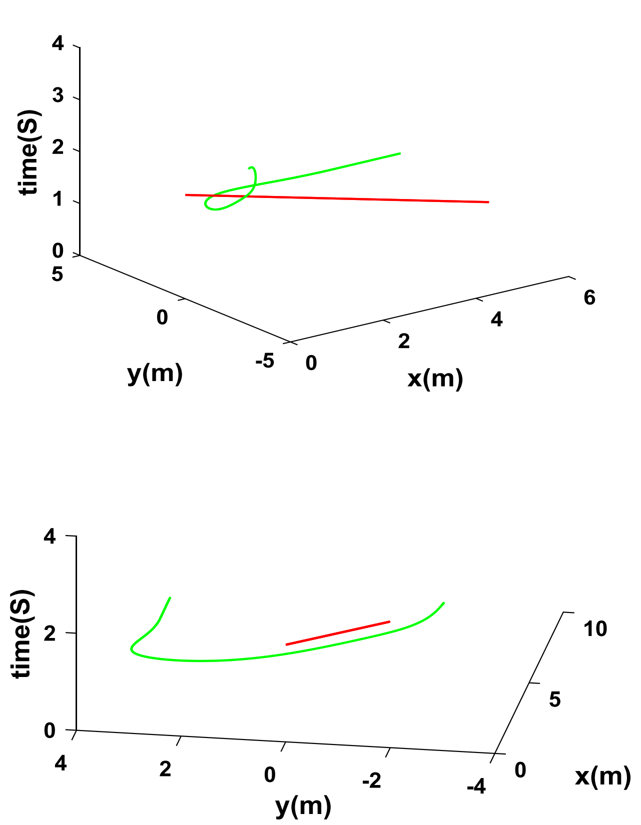

Fig. 3 Robot's route (green line), moving obstacle (red line) -SDRE controller

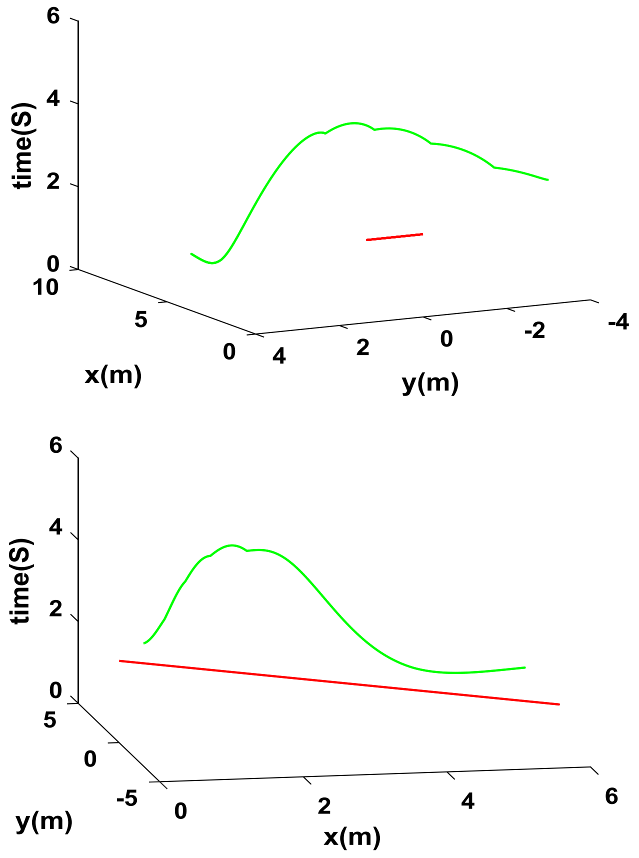

Fig. 4 Robot's route (green line), moving obstacle (red line)-LQR controller

The SDRE controller's reaction, however, is not affected by such changes, and this is the most important advantage of this controller. Furthermore, the state and input variables through the SDRE controller are smoother and without severe fluctuations in comparison with LQR controller.

It is worth mentioning that under all three different movements for obstacle, for any changes in the robot's and the obstacle's initial conditions, the performance of our proposed methodology without any changes in the SDRE controller's conditions was assessed and it was concluded that the performance of the method in all cases had been desirable.

4.1 Fixed speed

In this state, the robot starts moving from the point $(x_0, y_0) = (-3, 3)$, while the target is to get to the point $(x_d, y_d) = (3, 1)$. The obstacle also starts moving with a fixed speed of – 0.2 m/s along '*x*' and '*y*' from the point

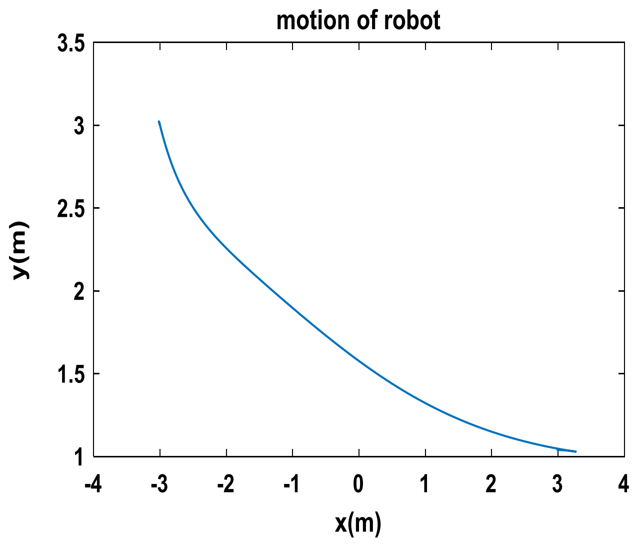

Fig. 5 The route travelled by the robot, SDRE controller

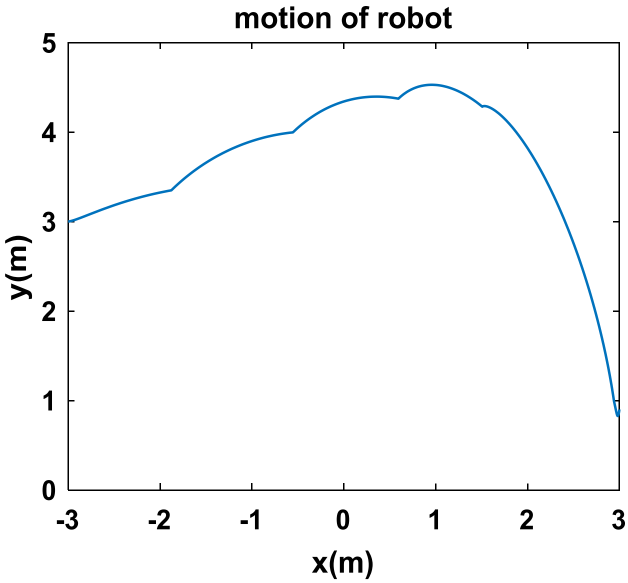

Fig. 6 The route travelled by the robot, LQR controller

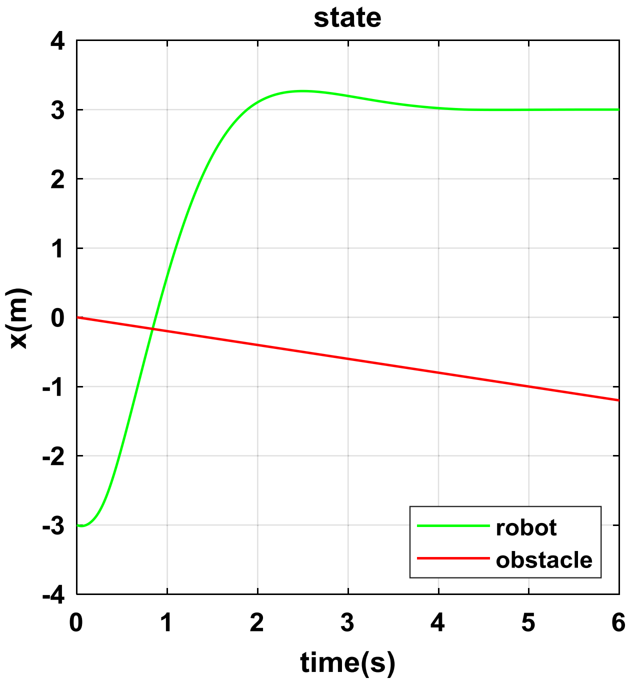

Fig. 7 Behavior of variable ***x*** SDRE controller

$\left(x_{ob}, y_{ob}\right) = (0, 2)$ (Figs. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14), $Q_x = 50\,(1/\mathrm{m})^2, Q_y = 50\,(1/\mathrm{m})^2, Q_\theta = 1\,(1/\mathrm{rad})^2, Q_v = 1$ $(\mathrm{s/m})^2, Q_\varphi = 30(1/\mathrm{rad})^2, R_a = 5\,\left(\mathrm{s}^2/\mathrm{m}\right)^2, R_\omega = 5\,(\mathrm{s/rad})^2$.

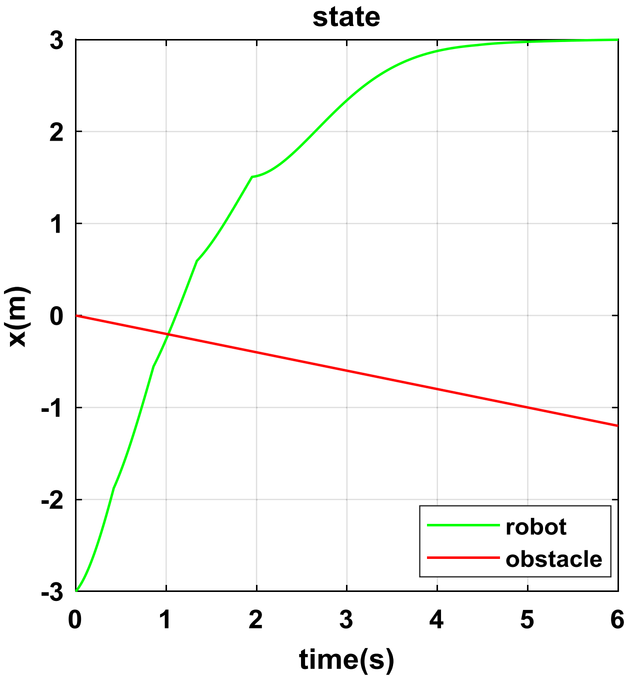

Fig. 8 Behavior of variable ***x*** LQR controller

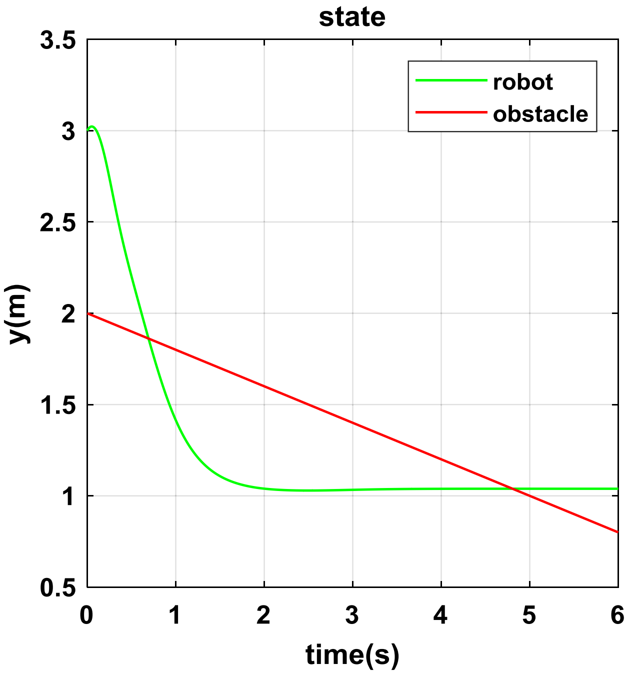

Fig. 9 Behavior of variable ***y*** SDRE controller

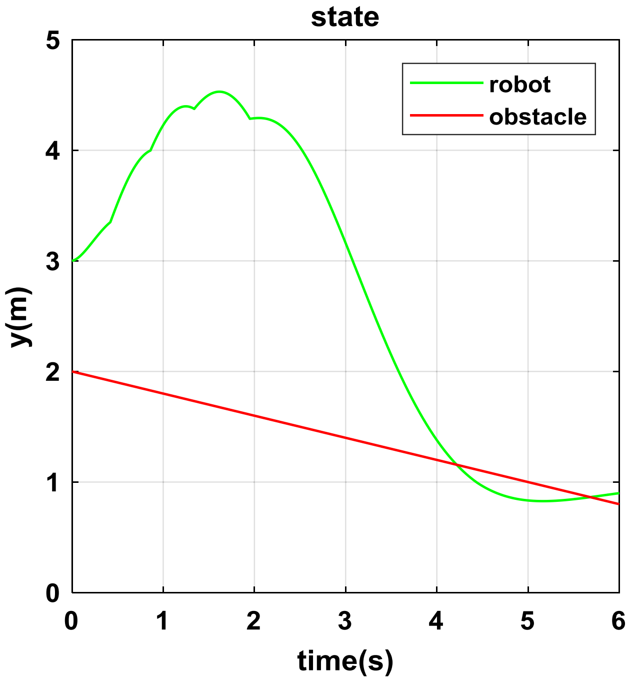

Fig. 10 Behavior of variable ***y*** LQR controller

4.2 Fixed acceleration

In this state, the robot is located at the start point of $(x_0, y_0) = (2, 2)$ and the target is specified at the point $(x_d, y_d) = (-4, 1)$. The obstacle starts moving from the point $(x_{ob}, y_{ob}) = (-1, 1)$ with an initial speed of 0.1 m/s along 'x' and 0.1 m/s along 'y' with a fixed acceleration of + 0.3 m/s^2 along 'x' and – 0.2 m/s^2 along 'y' (Figs. 15, 16, 17, 18, 19, 20), $Q_x = 50(1/m)^2, Q_y = 50(1/\text{m})^2, Q_\theta = 1(1/\text{rad})^2, Q_v = 1$ $(\text{s/m})^2, Q_\varphi = 1(1/\text{rad})^2, R_a = 20\left(\text{s}^2/\text{m}\right)^2, R_\omega = 5(\text{s/rad})^2$.

4.3 Non-uniform circular

In this state, the robot is located at the start point of $(x_0, y_0) = (2, 2)$ and the target is specified at the point $(x_d, y_d) = (-4, 1)$. The obstacle, on the other hand, starts moving from the point $(x_{ob}, y_{ob}) = (-3, 2)$ with an angular frequency of 2 rad/s (Figs. 21, 22, 23, 24, 25, 26). $Q_x = 50(1/m)^2, Q_y = 50(1/\text{m})^2, Q_\theta = 1(1/\text{rad})^2, Q_v = 1$ $(\text{s/m})^2, Q_\varphi = 20(1/\text{rad})^2, R_a = 10\left(\text{s}^2/\text{m}\right)^2, R_\omega = 5(\text{s/rad})^2$.

As it is given in remark 3 and simulations, the proposed method not only can avoid from the fixed obstacle but also has a good ability to avoid from the moving obstacle due to the new cost function. Moreover, the smooth response is observed and input variables are gained which can be applied to real systems.The most important advantage is the ability to avoid from the obstacle near the target in comparison with the two other methods, and the ability to respond to all the working points in comparison with LQR (see Tables 2, 3, 4 and "Appendix 2").

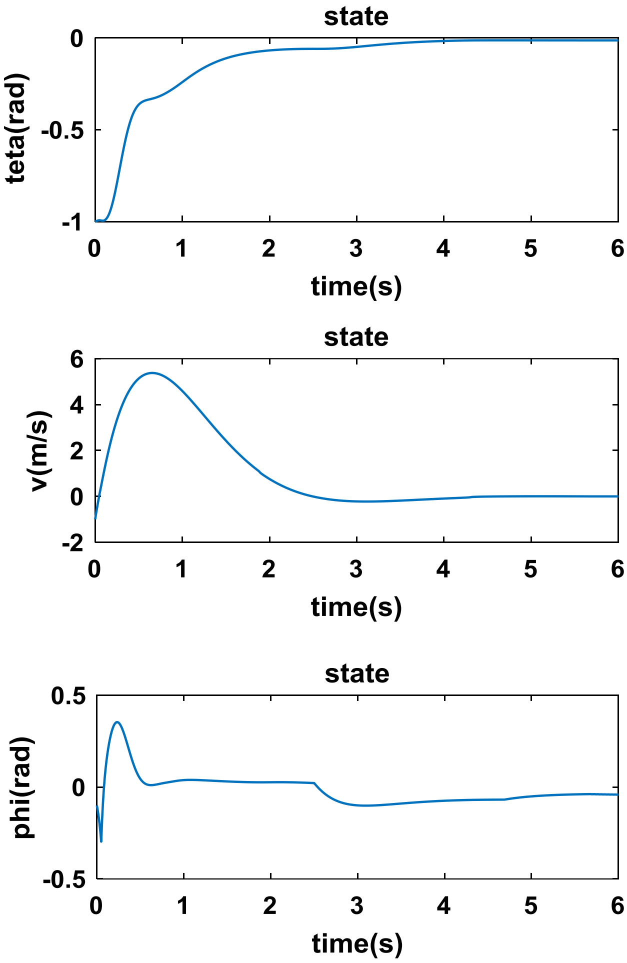

Fig. 11 Behavior of variables θ, v and φ, SDRE controller

Response means the behavior of all the variables of system.

5 Conclusion

In this study, a new controller was proposed for navigating a non-holonomic wheeled moving robot within a dynamic environment in the presence of a moving obstacle. The proposed controller was constructed of two parts; the nonlinear optimal controller and the obstacle avoidance algorithm. Indeed, the control rule extracted from the SDRE (nonlinear optimal controller) in combination with the APF algorithm (obstacle avoidance algorithm). The APF algorithm operated based on defining the potential

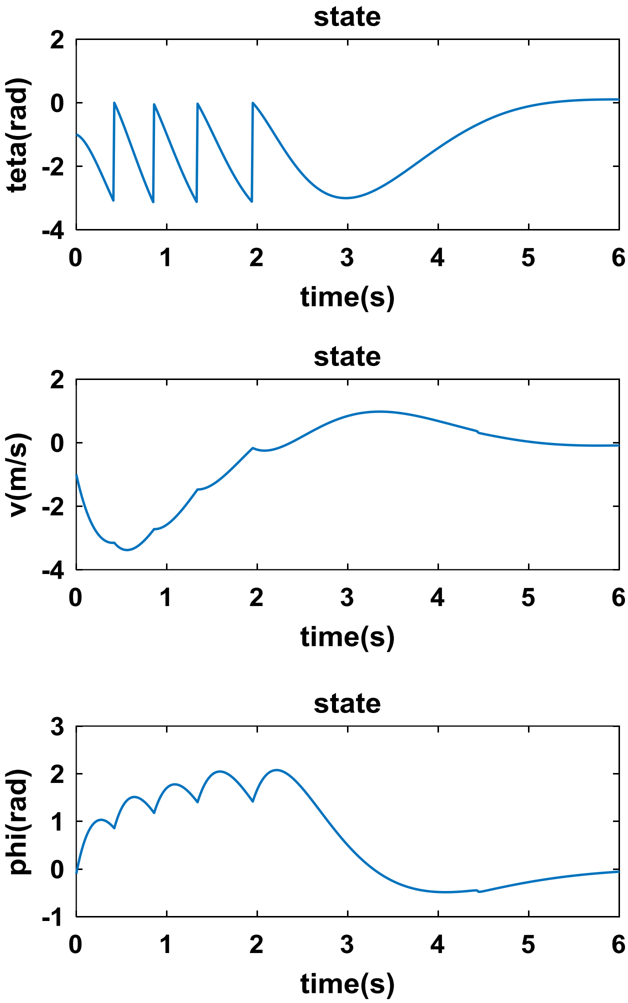

Fig. 12 Behavior of variables θ, v and φ, LQR controller

function over the target and obstacle position. This potential function was employed in the cost function of SDRE. Any changes in the positions of the target or the obstacle change the potential function and the cost function respectively. Therefore, when the robot approached the obstacle, it first avoided the obstacle by increasing the distance from the target and cost function and then managed to find its route by decreasing the cost function in the presence of the obstacle. In this paper, route design was done in the presence of the obstacles with three different movements (i.e. moving with 'a fixed speed', 'a fixed acceleration' and 'a non-uniform circular' movement). It should also be added that the method applied in this study has more advantages than the LQR controller since the SDRE methodology is not restricted by working points. Hence, it results in a better navigation and an obviously smoother response.

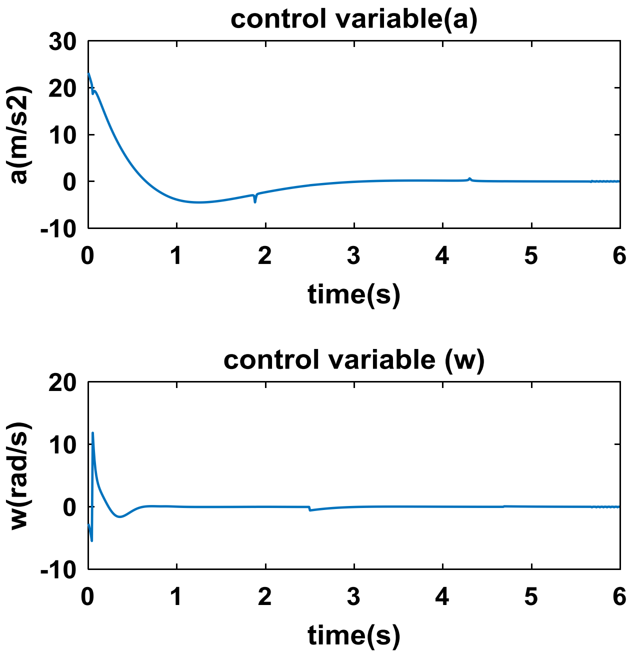

Fig. 13 Behavior of variables a and ω, SDRE controller

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1 Simulation results for Table 1

Case 2: Near the both target and obstacle

Results of proposed method by [30]:

In this case robot starts moving from the point $\left(x_0, y_0\right) = (-2, -2)$ and the target is to get the point $\left(x_d, y_d\right) = (5, 5)$. The obstacle, also, is located in the point $\left(x_{ob}, y_{ob}\right) = (4.5, 5)$. All the design parameters are the same for all methods (Figs. 27, 28, 29, 30, 31).

Results of proposed method by [31]:

See Figs. 32, 33, 34, 35, 36.

Results of proposed method by this paper:

In this case robot should approach the target which is located near the obstacle.

According to the Figs. 27, 28, 29, 30, 31 and 32, the method proposed by [31] is not able to reach the target in the time defined in the controllers as the methods proposed by this paper and [30] (the robot get the point about (2.2, 2.4) in that time). So there is a compulsory for us to increase the time, and it can be seen that after almost 12 s, when the robot is approaching the target located near the obstacle, the input variables are not smooth.

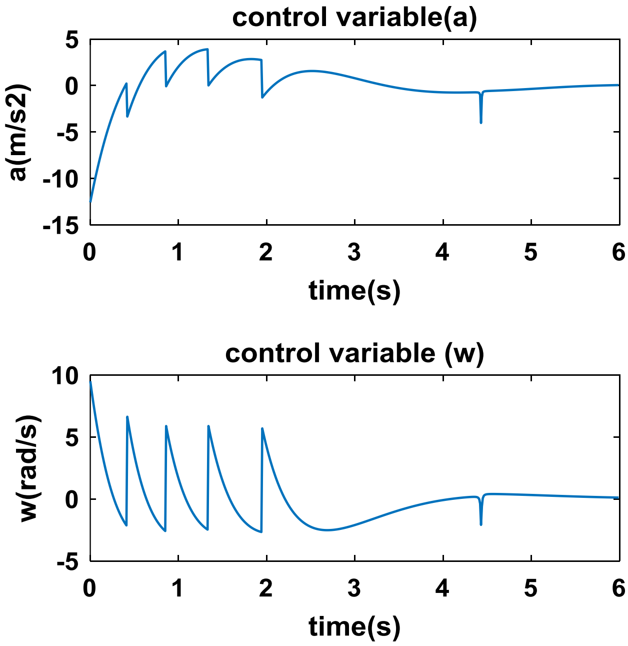

Fig. 14 Behavior of variables *a* and *ω*, LQR controller

In other words, the input variables do not have suitable behavior from t = 12 s when the robot is near to both the target and obstacle (see Fig. 36). Via the other method proposed by [30], the robot has a collision with obstacle, and its input variables are not smooth. As shown in Fig. 31, not only the control inputs have very large values but they also behave inappropriately. This inadmissible behavior of the input variables happen from t = 3 s when the robot is near to both the target and obstacle. Through the proposed method by this paper, the robot not only can avoid hitting the obstacle but also get to the target in optimal time and distance (Fig. 37). Furthermore, the control input signal is admissible and smooth, and this distinction is on the basis of new cost function (Figs. 37, 38, 39, 40, 41).

Case 4: Far from the target and near the obstacle

Results of proposed method by [30, 31] and this method:

In this case robot starts moving from the point $\left(x_0, y_0\right) = (-2, -2)$ and the target is to get the point $\left(x_d, y_d\right) = (5, 5)$. The obstacle, also, is located in the point $\left(x_{ob}, y_{ob}\right) = (-1.5, -1.5)$. All the design parameters are the same for all methods.

Results of proposed method by [30]:

See Figs. 42, 43, 44, 45 and 46.

Results of proposed method by [31]:

See Figs. 47, 48, 49, 50 and 51.

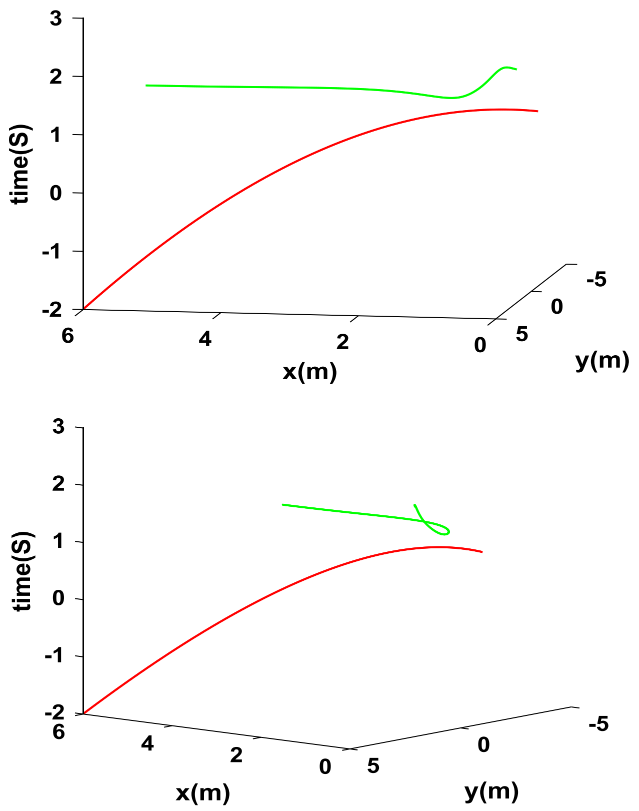

Fig. 15 Robot's route (green line), moving obstacle (red line)

Results of proposed method by this paper:

In this case robot should move toward the target which is located far from the obstacle (Figs. 52, 53, 54, 55, 56).

According to the Figs. 42, 43, 44, 45 and 46, through the proposed method by [30], the robot reach the point (6, 5) and the input variables are too big. Via the method proposed by [31], the robot pass the target and reach the point about (8, 5) (see Fig. 47). Also, the input variables especially w is not smooth (see Fig. 51). Via the proposed method by this paper, the robot reach the target without collision and has a smooth and admissible input variables (see Figs. 52, 53, 54, 55, 56).

Appendix 2 Simulation results for LQR in other movements (Tables 3 and 4)

Fixed acceleration

In this state, the robot is located at the start point of $\left(x_0, y_0\right) = (2, 2)$ and the target is specified at the point $\left(x_d, y_d\right) = (-4, 1)$. The obstacle starts moving from the point $\left(x_{ob}, y_{ob}\right) = (-1, 1)$ with an initial speed of 0.1 m/s along '*x*' and 0.1 m/s along '*y*' with a fixed

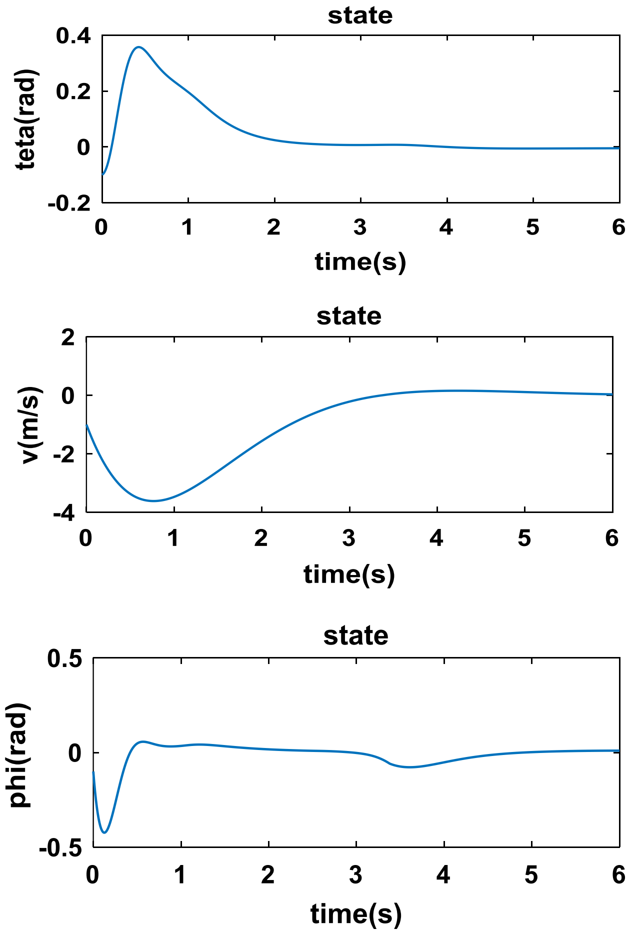

Fig. 16 Behavior of variables θ, v and φ

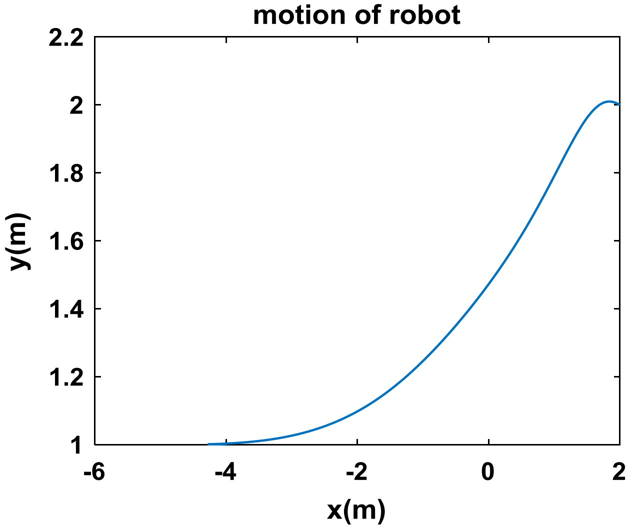

Fig. 17 The route travelled by the robot

acceleration of + 0.3 m/s^2 along '*x*' and − 0.2 m/s^2 along '*y*'

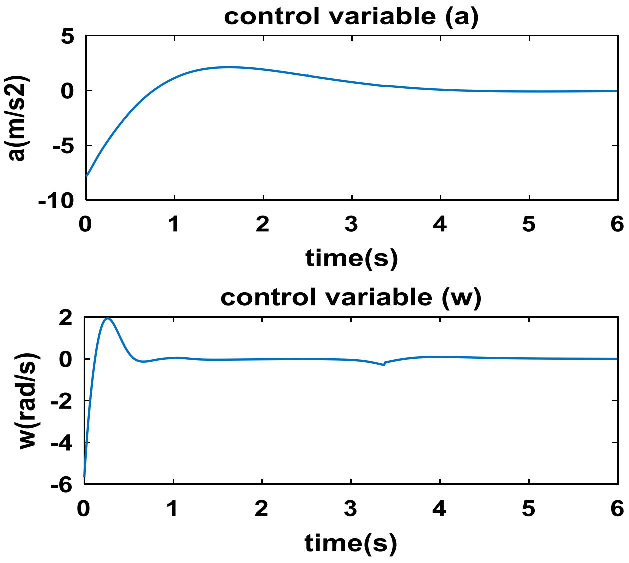

Fig. 18 Behavior of variables a and ω

$(Q_x = 50(1/m)^2, Q_y = 50(1/\mathrm{m})^2, Q_\theta = 1(1/\mathrm{rad})^2, Q_v = 1(\mathrm{s}/\mathrm{m})^2$, $Q_\varphi = 1(1/\mathrm{rad})^2, R_a = 20\left(\mathrm{s}^2/\mathrm{m}\right)^2, R_\omega = 5(\mathrm{s}/\mathrm{rad})^2$ (Figs. 57, 58, 59, 60, 61, 62).

Non-uniform circular

In this state, the robot is located at the start point of $\left(x_0, y_0\right) = (2, 2)$ and the target is specified at the point $\left(x_d, y_d\right) = (-4, 1)$. The obstacle, on the other hand, starts moving from the point $\left(x_{ob}, y_{ob}\right) = (-3, 2)$ with an angular frequency of 2 rad/s. $Q_x = 50(1/\mathrm{m})^2, Q_y = 50(1/\mathrm{m})^2, Q_\theta = 1(1/\mathrm{rad})^2, Q_v = 1(\mathrm{s}/\mathrm{m})^2$, $Q_\varphi = 20(1/\mathrm{rad})^2, R_a = 10\left(\mathrm{s}^2/\mathrm{m}\right)^2, R_\omega = 5(\mathrm{s}/\mathrm{rad})^2$ (Figs. 63, 64, 65, 66, 67, 68).

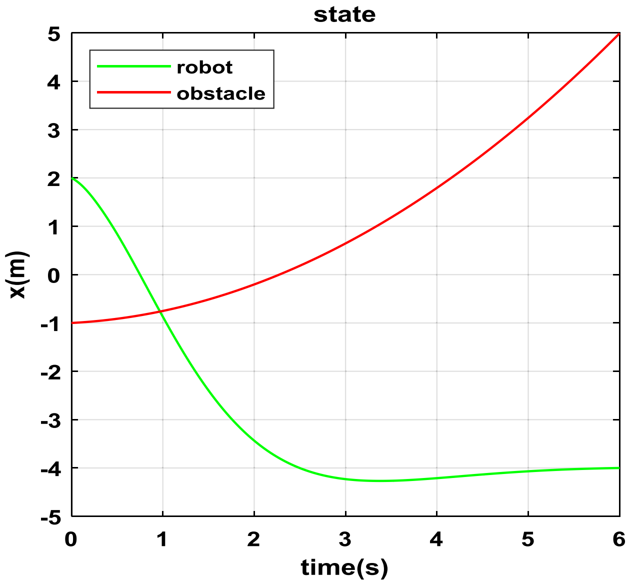

Fig. 19 Behavior of variable ***x***

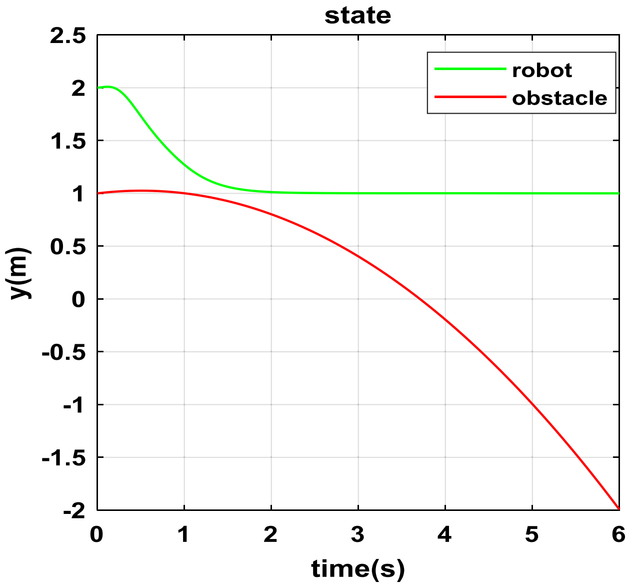

Fig. 20 Behavior of variable ***y***

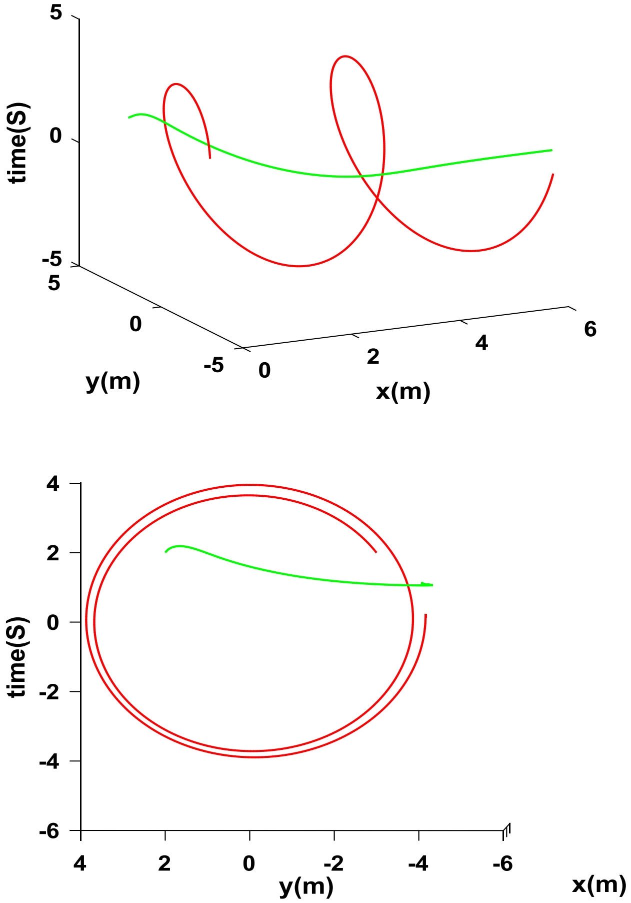

Fig. 21 Robot's route (green line), moving obstacle (red line)

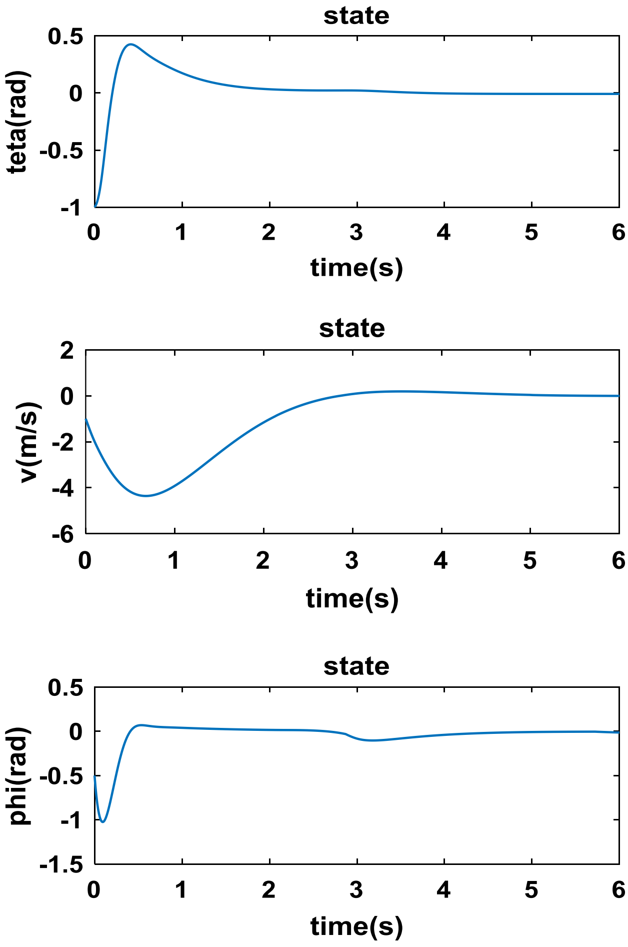

Fig. 22 Behavior of variables θ, v and φ

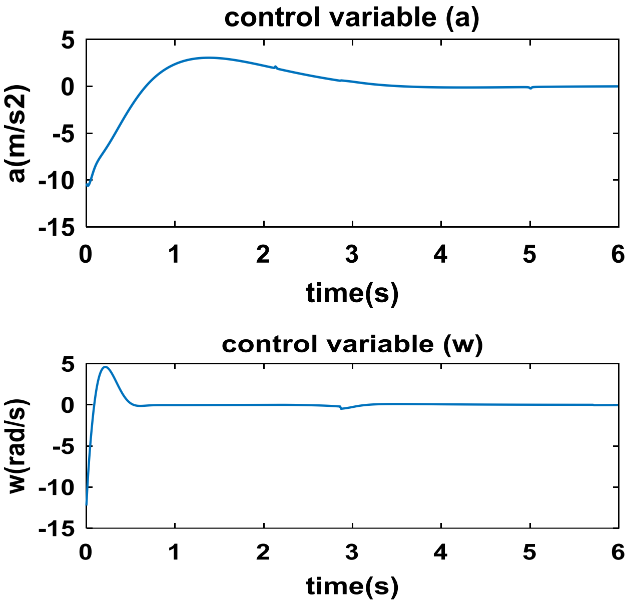

Fig. 24 Behavior of variables a and ω

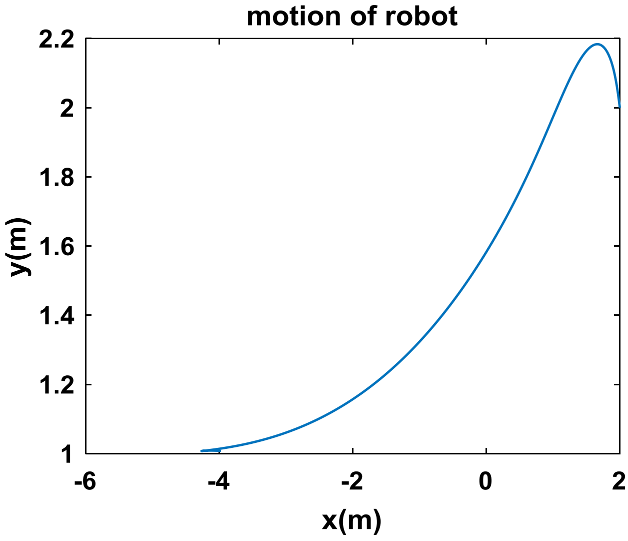

Fig. 23 The route travelled by the robot

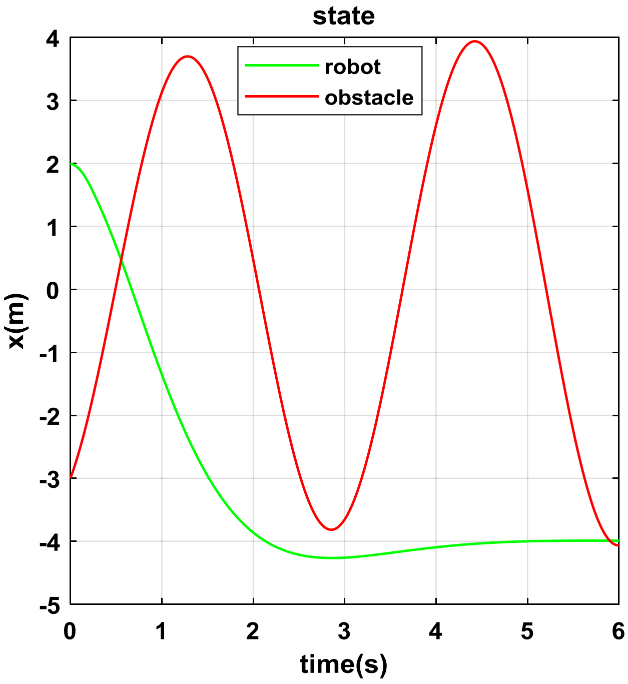

Fig. 25 Behavior of variable ***x***

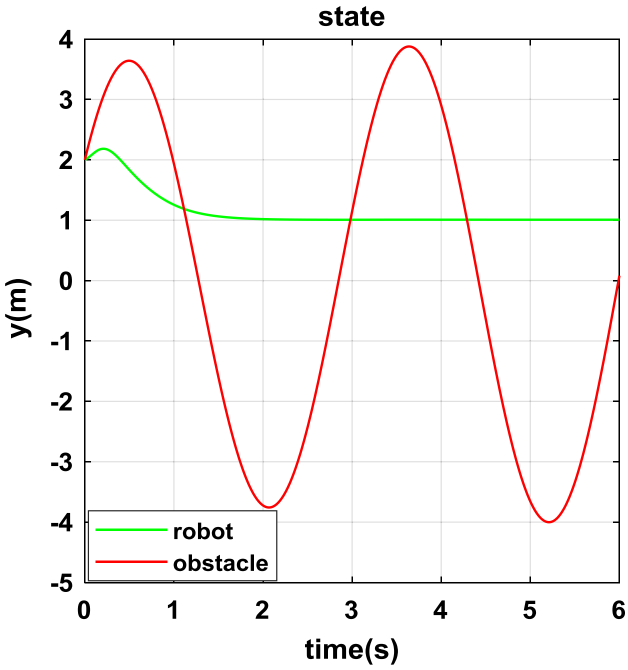

Fig. 26 Behavior of variable ***y***

Table 2 Display the results from proposed method and LQR in presence of moving obstacle

Used method	Type of obstacle	Hit	Approach time (s)	Travelled distance (m)	Smooth response	Admissible input	Maximum of input variable
LQR	Moving (fixed speed)	No	6	9.0182	No	No	9.5182
Proposed method by this paper	Moving (fixed speed)	No	4	7.0324	Yes	Yes	23.1876

Table 3 Display the results from proposed method and LQR in presence of moving obstacle

Used method	Type of obstacle	Hit	Approach time (s)	Travelled distance (m)	Smooth response	Admissible input	Maximum of input variable
LQR	Moving (fixed acceleration)	No	5	6.7076	No	No	18.3214
Proposed method by this paper	Moving (fixed acceleration)	No	5	6.6585	Yes	Yes	2.1236

Table 4 Display the results from proposed method and LQR in presence of moving obstacle

Used method	Type of obstacle	Hit	Approach time (s)	Travelled distance (m)	Smooth response	Admissible input	Maximum of input variable
LQR	Moving (non-uniform circular)	No	5	6.7630	No	No	16.8362
Proposed method by this paper	Moving (non-uniform circular)	No	4	6.7745	Yes	Yes	4.6076

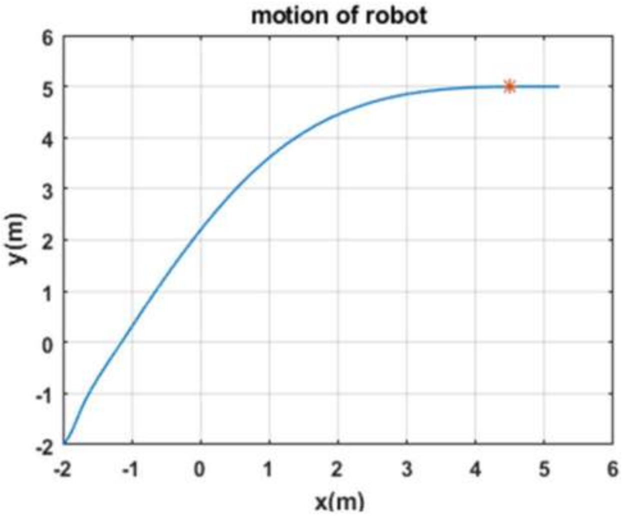

Fig. 27 The rout travelled by robot

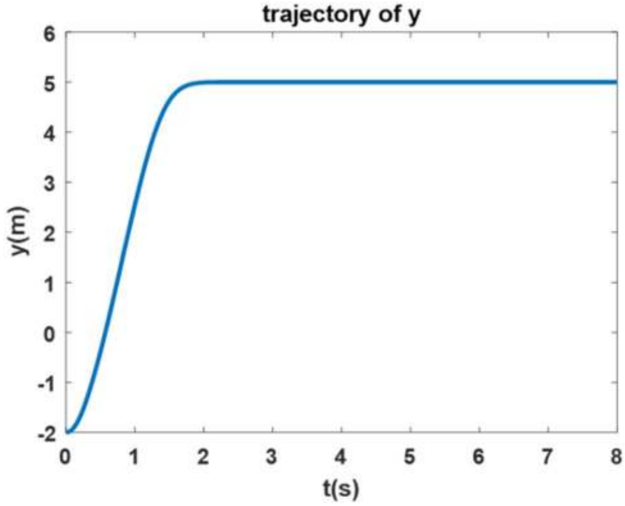

Fig. 29 Behavior of variable **y**

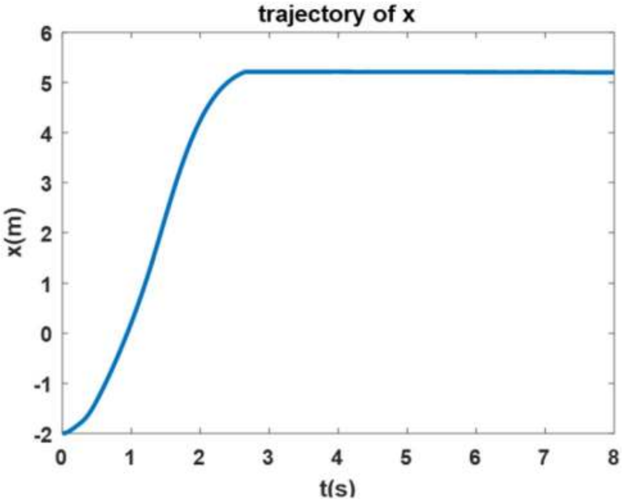

Fig. 28 Behavior of variable **x**

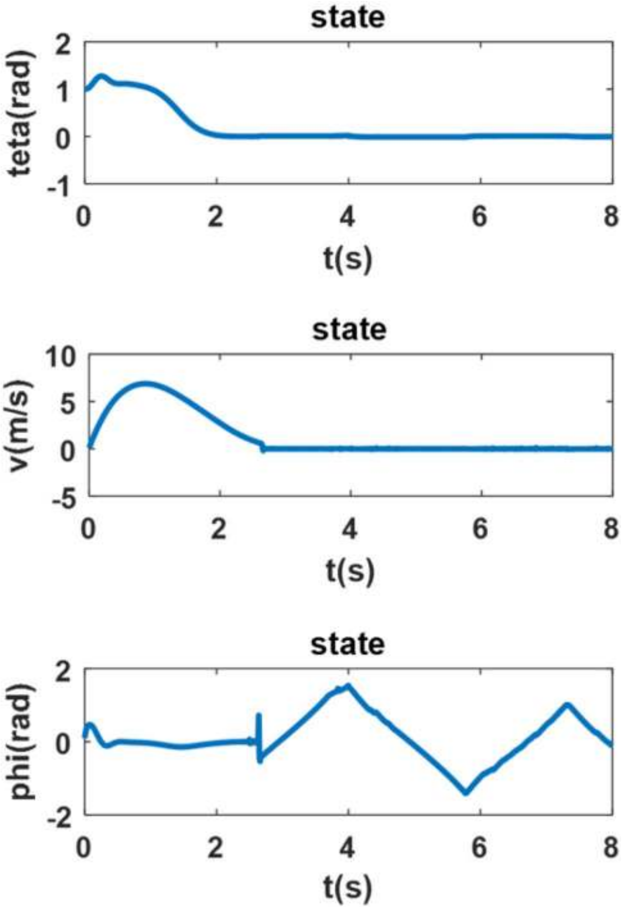

Fig. 30 Behavior of variables θ, v,φ

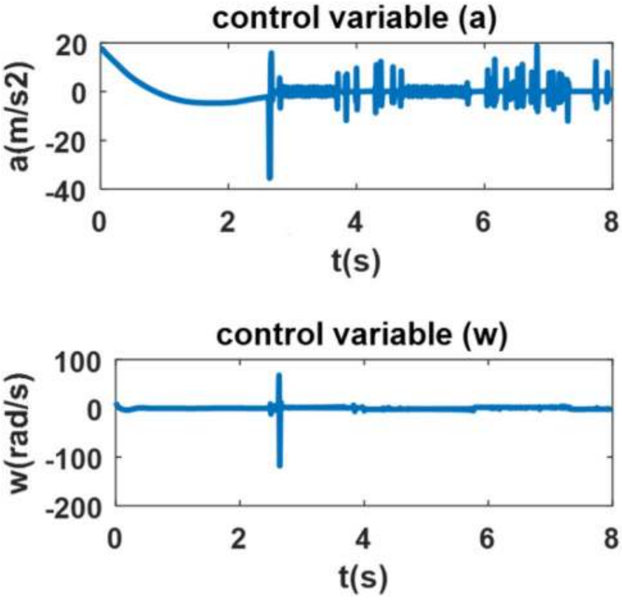

Fig. 31 Behavior of variables *a* and *ω*

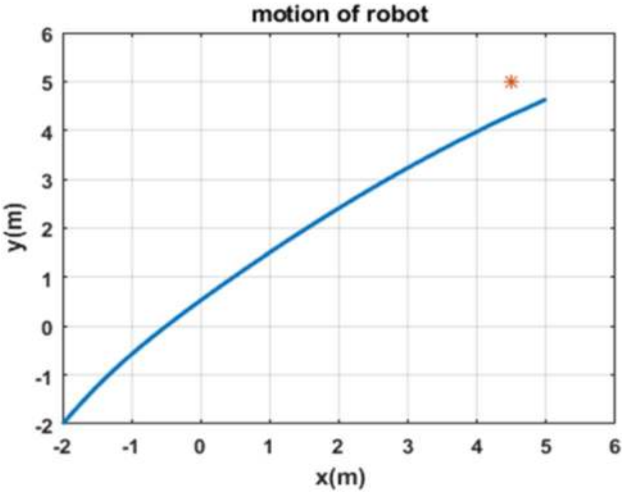

Fig. 32 The rout travelled by robot

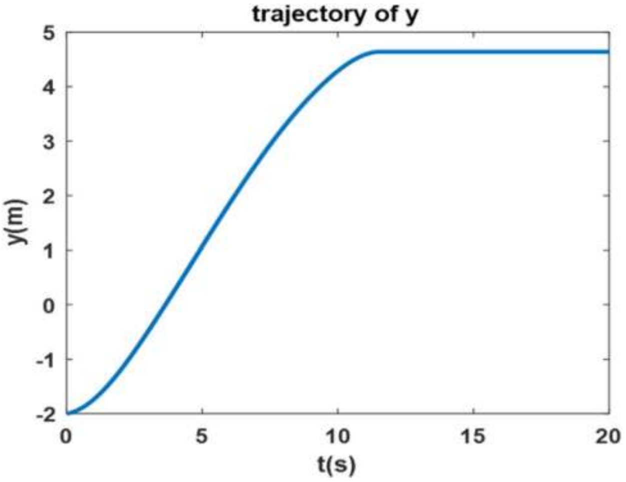

Fig. 33 Behavior of variable **y**

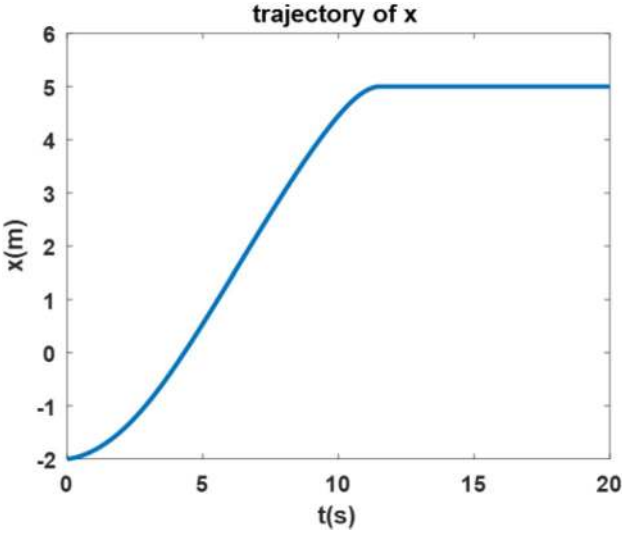

Fig. 34 Behavior of variable **x**

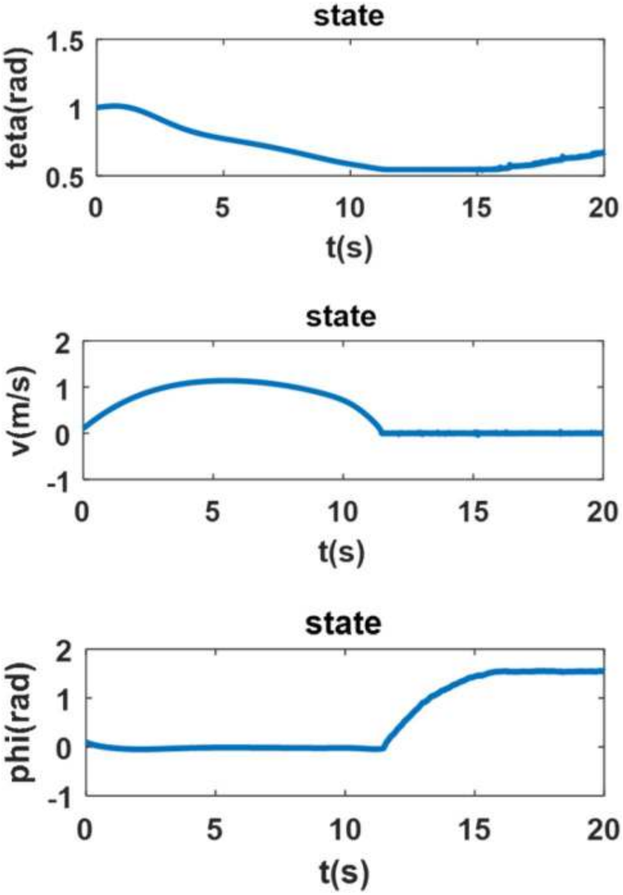

Fig. 35 Behavior of variables θ, v,φ

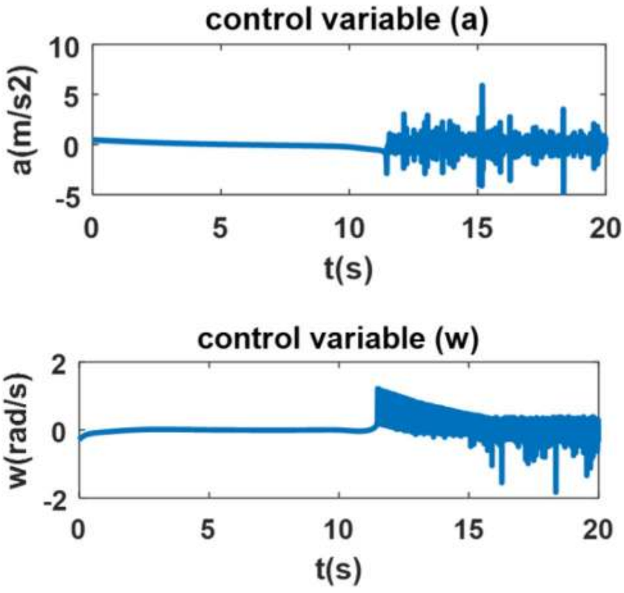

Fig. 36 Behavior of variables a and ω

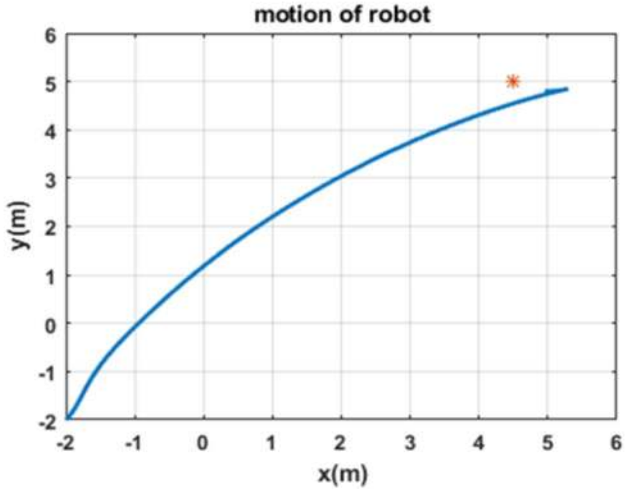

Fig. 37 The rout travelled by robot

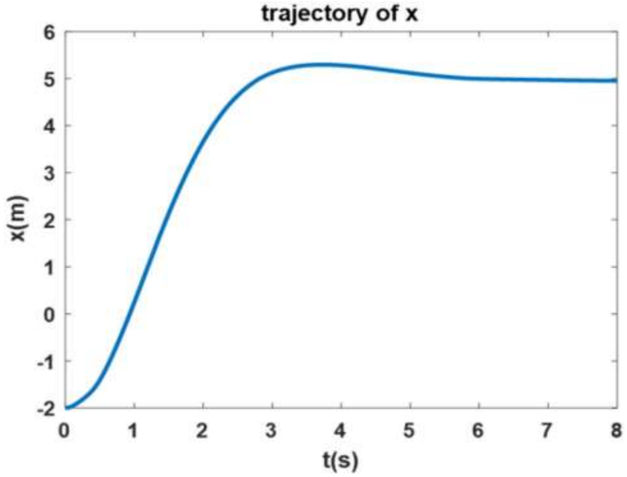

Fig. 38 Behavior of variable **x**

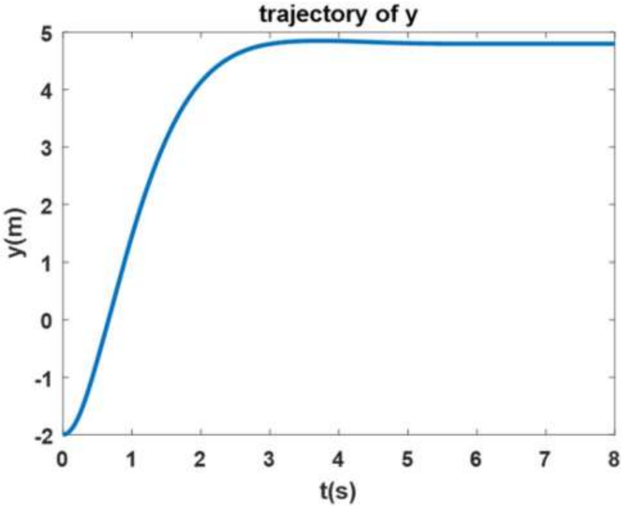

Fig. 39 Behavior of variable **y**

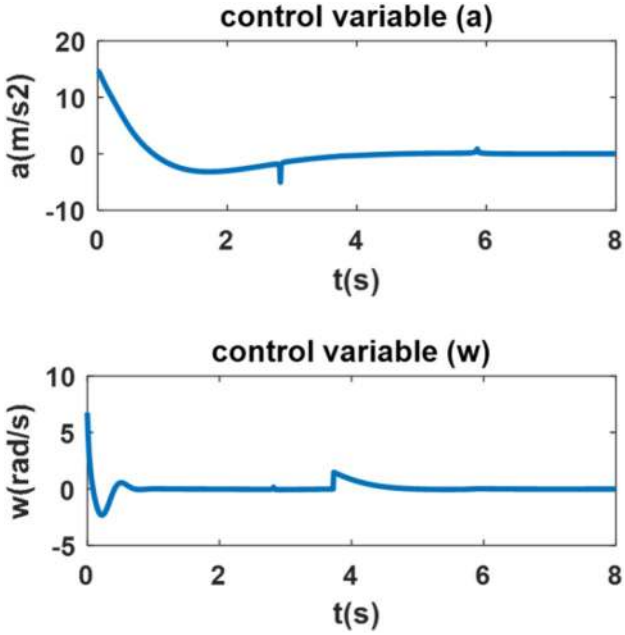

Fig. 41 Behavior of variables *a* and *ω*

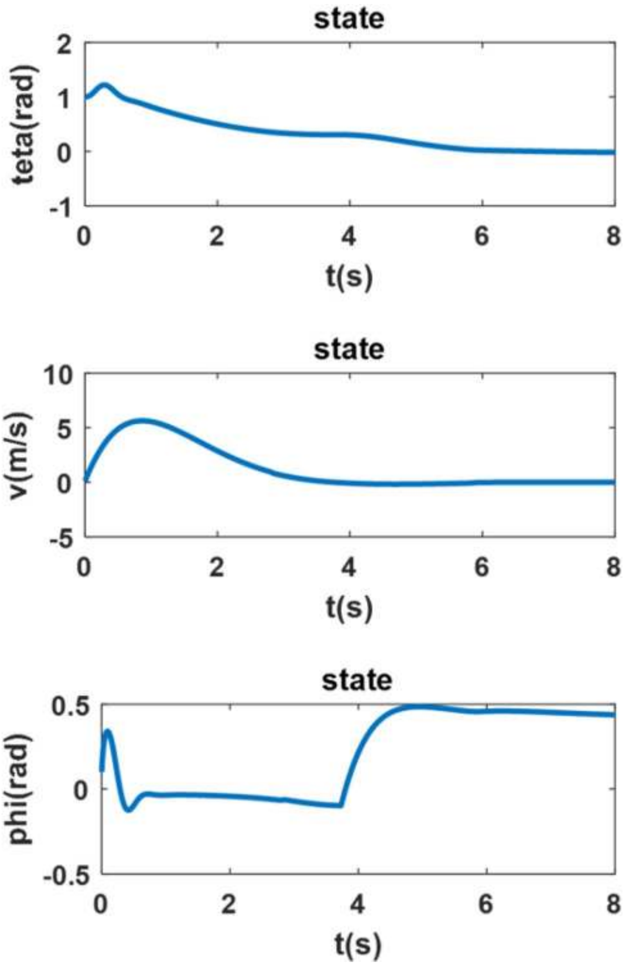

Fig. 40 Behavior of variables θ, v,φ

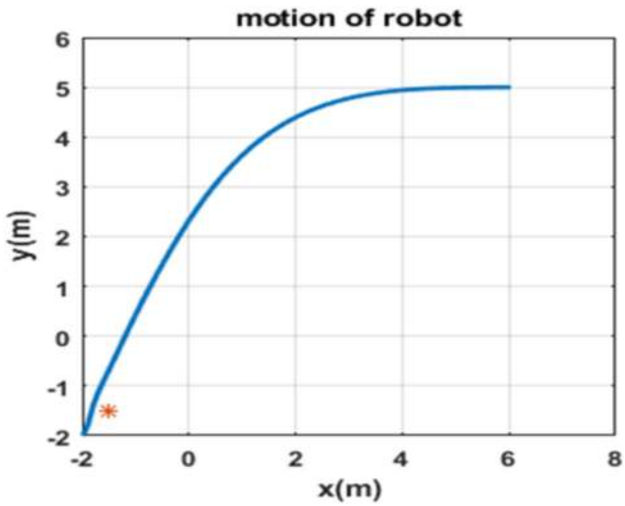

Fig. 42 The rout travelled by robot

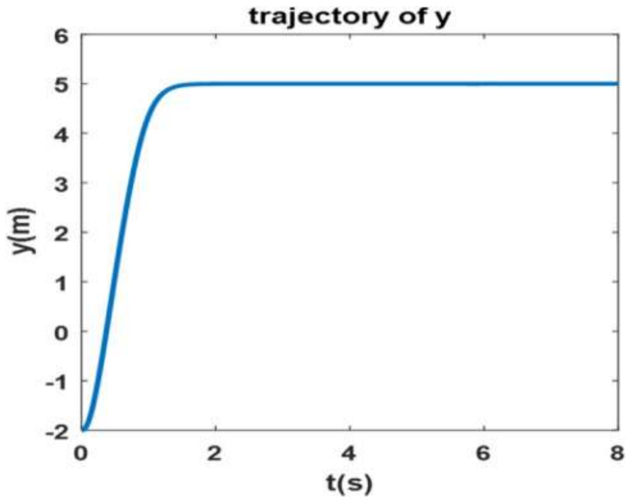

Fig. 43 Behavior of variable **y**

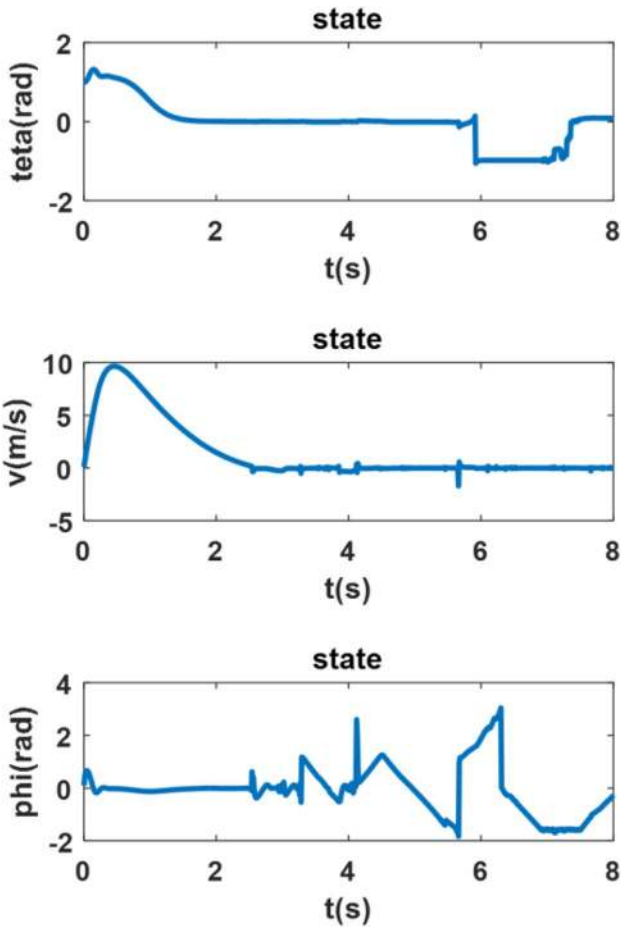

Fig. 44 Behavior of variables θ, v,φ

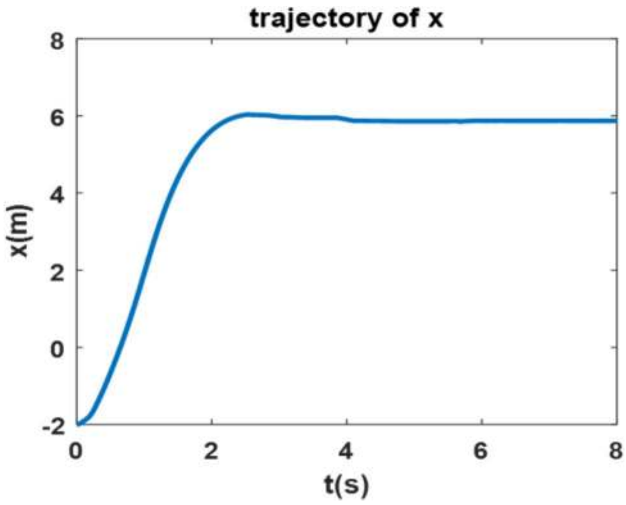

Fig. 45 Behavior of variable **x**

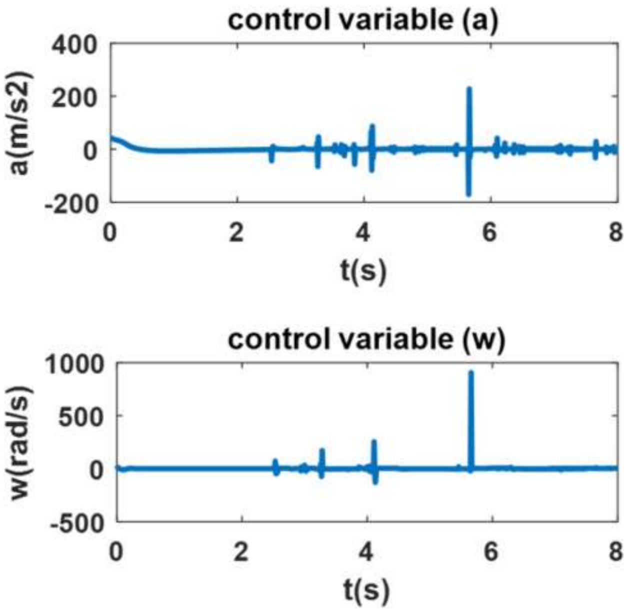

Fig. 46 Behavior of variables a and ω

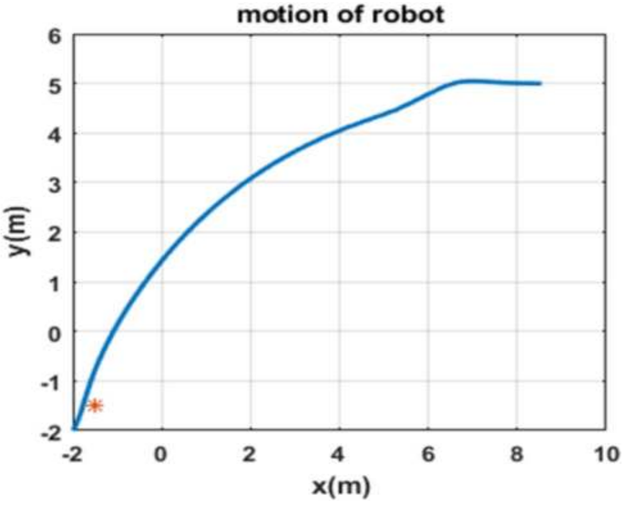

Fig. 47 The rout travelled by robot

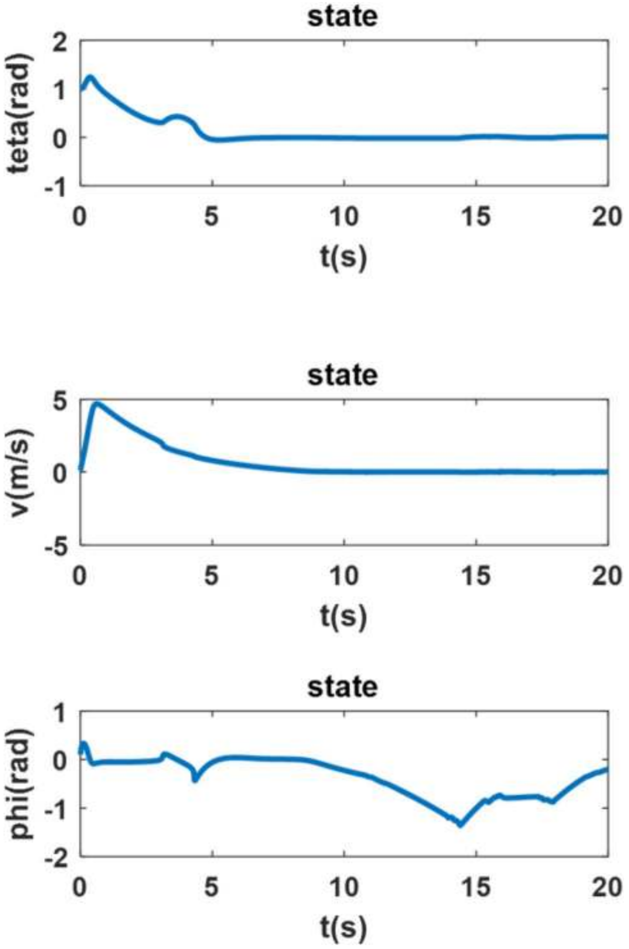

Fig. 49 Behavior of variables θ, v,φ

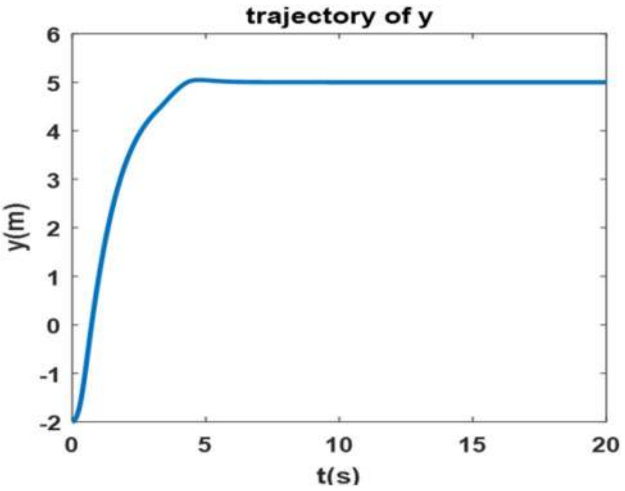

Fig. 48 Behavior of variable **y**

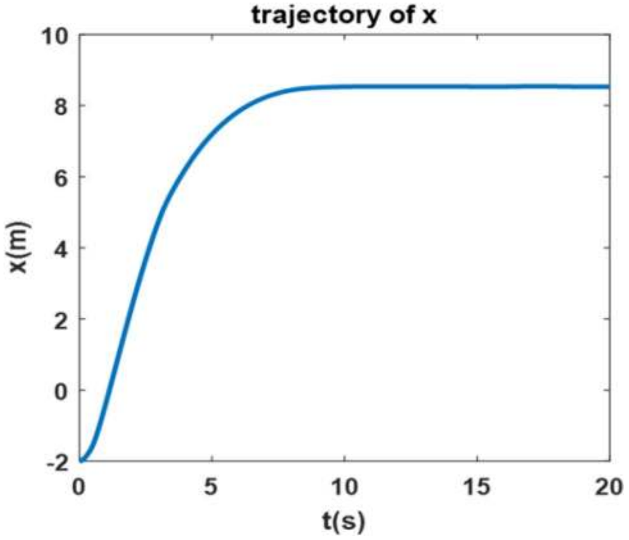

Fig. 50 Behavior of variable **x**

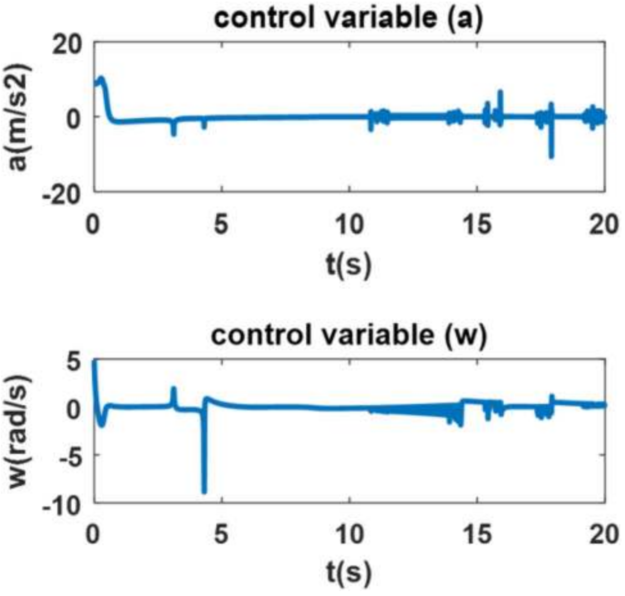

Fig. 51 Behavior of variables *a* and *ω*

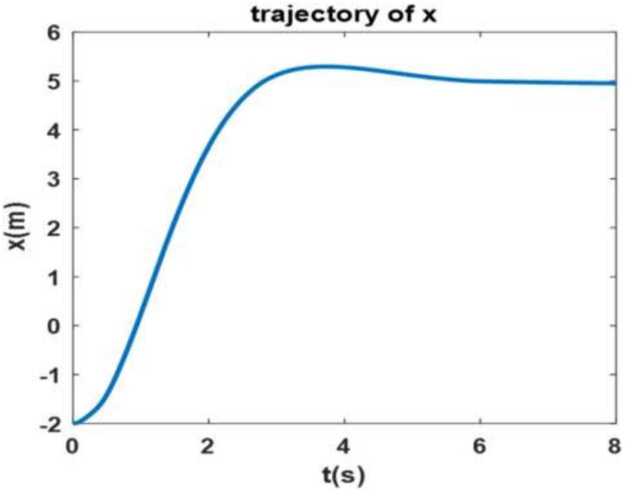

Fig. 53 Behavior of variable **x**

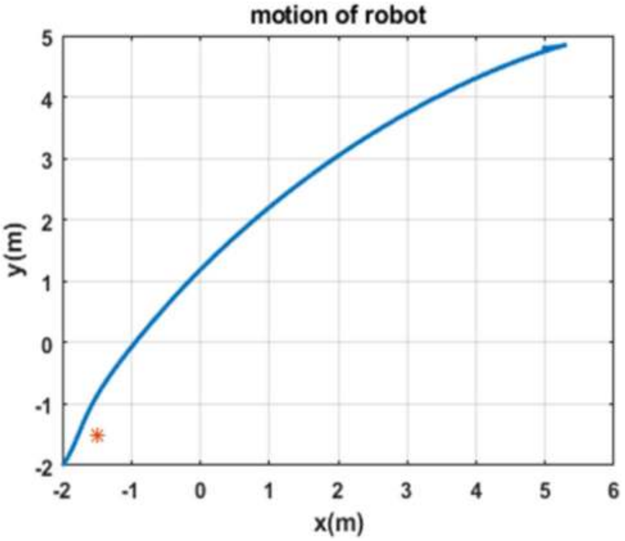

Fig. 52 The rout travelled by robot

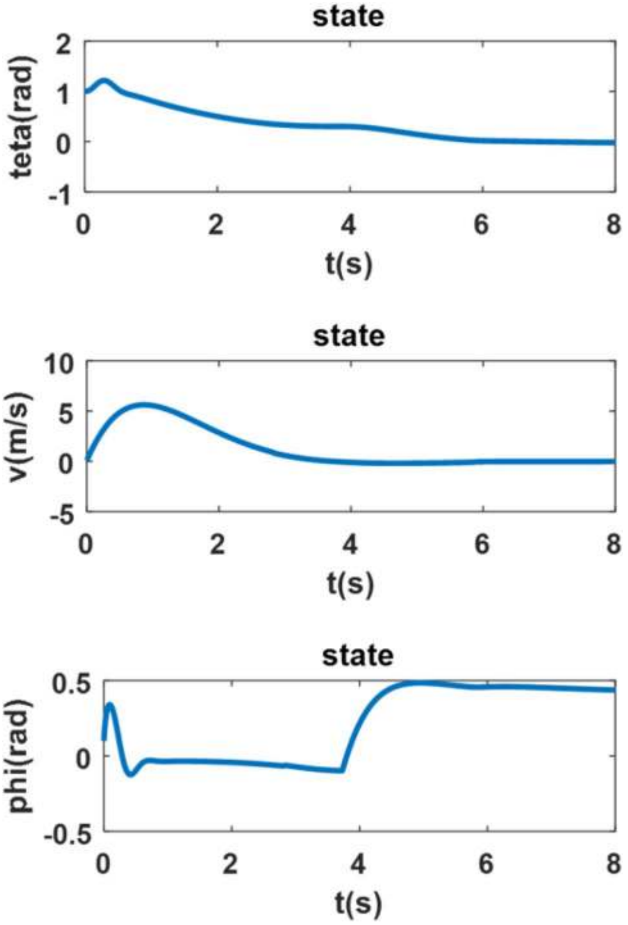

Fig. 54 Behavior of variables θ, v,φ

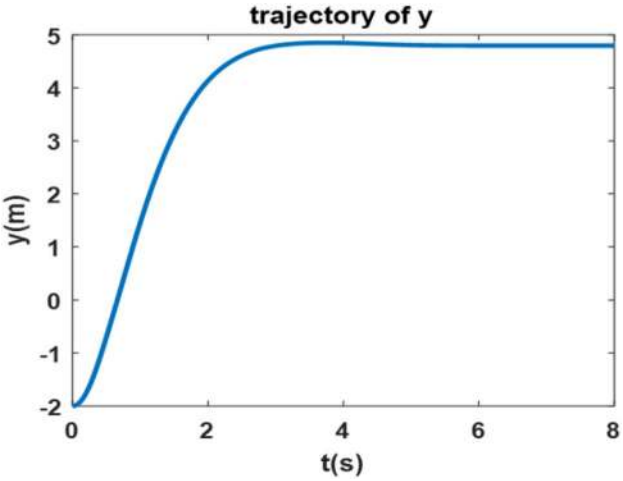

Fig. 55 Behavior of variable **y**

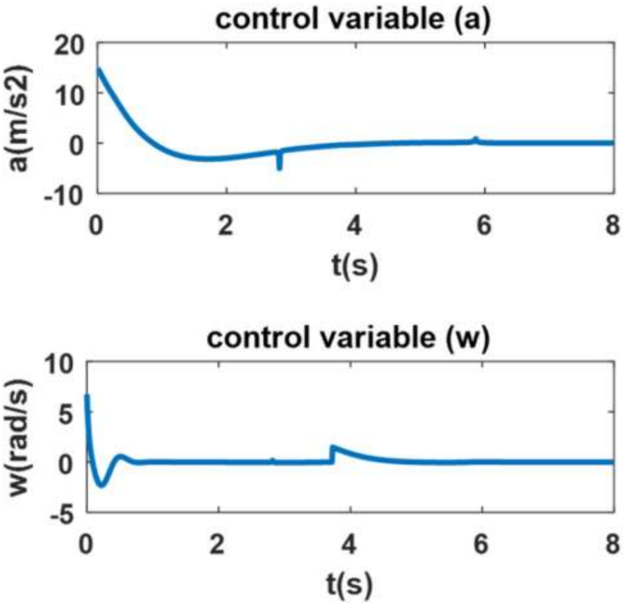

Fig. 56 Behavior of variables *a* and *ω*

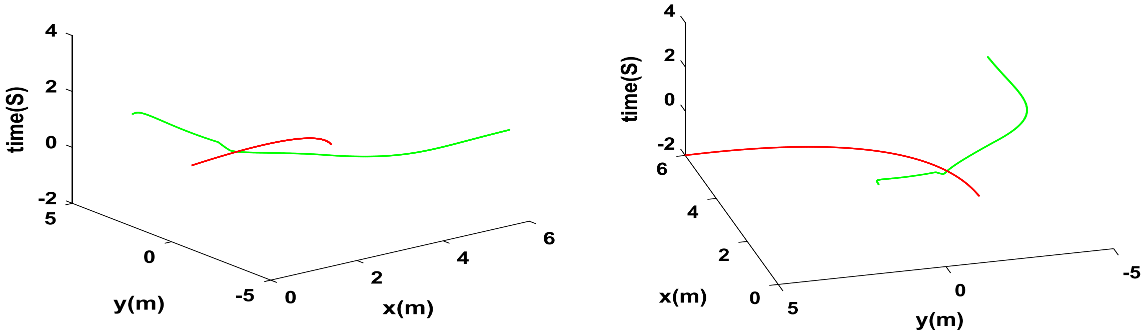

Fig. 57 Robot's routing (green line), moving obstacle (red line)—LQR controller

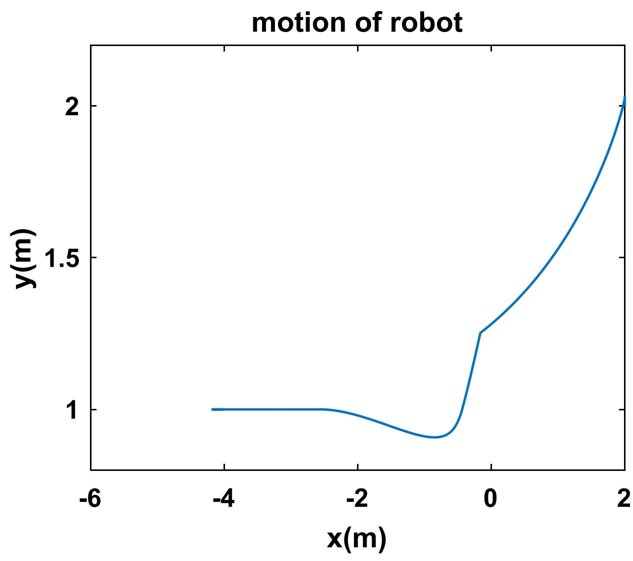

Fig. 58 The route travelled by the robot, LQR controller

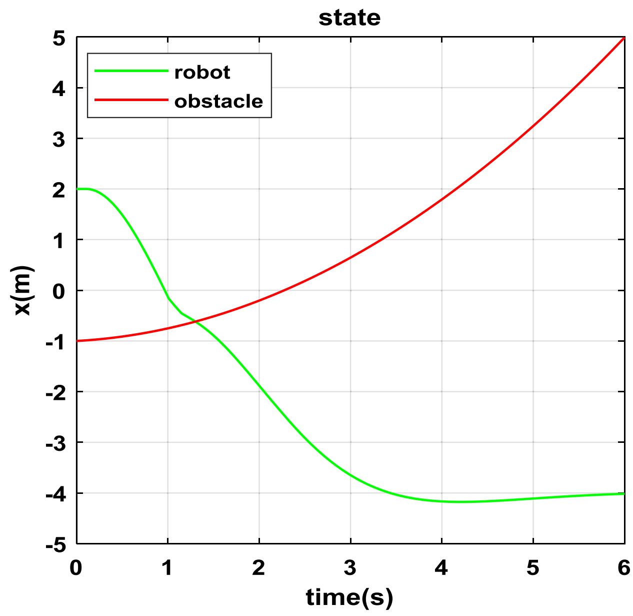

Fig. 59 Behavior of variable ***x*** LQR controller

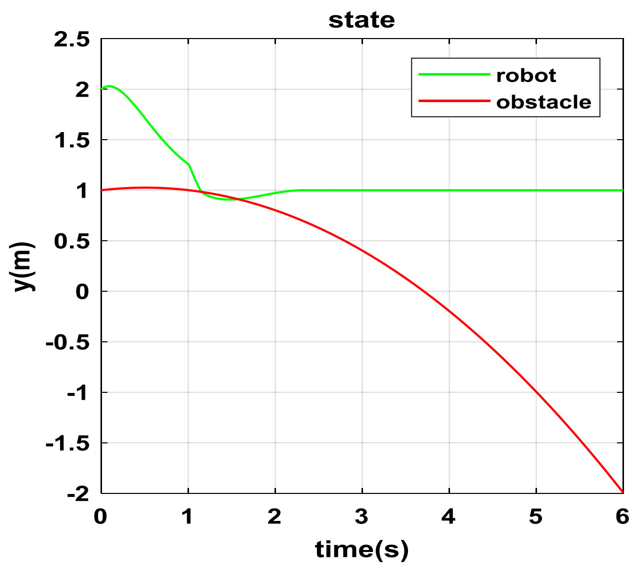

Fig. 60 Behavior of variable ***y*** LQR controller

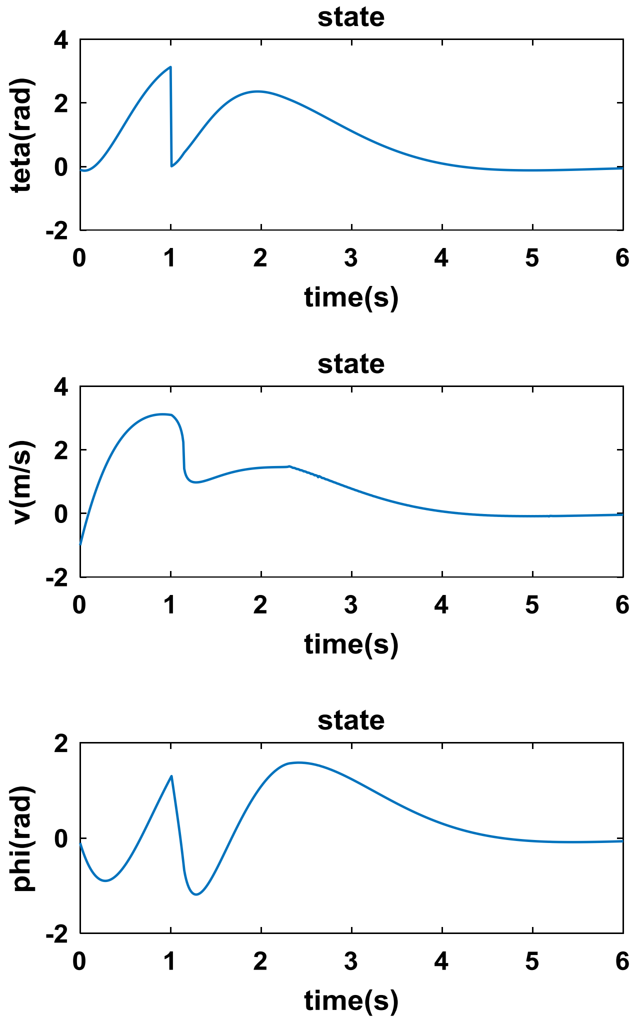

Fig. 61 Behavior of variables θ, v and φ LQR controller

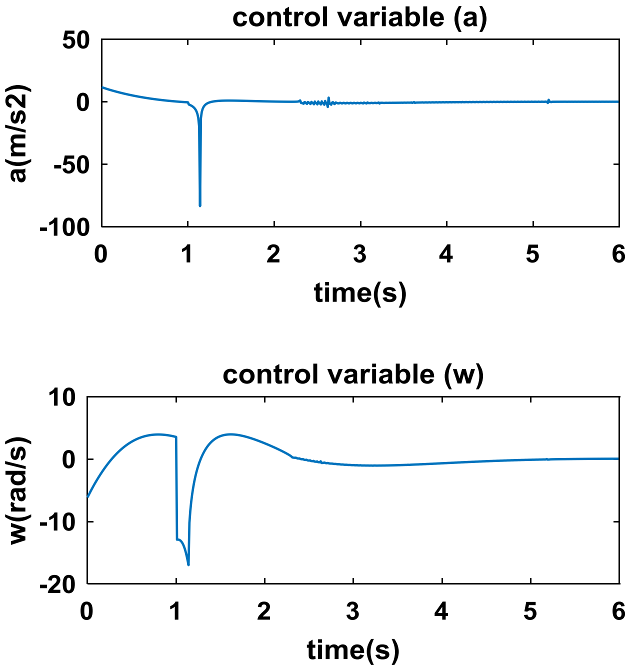

Fig. 62 Behavior of variables a and ω, LQR controller

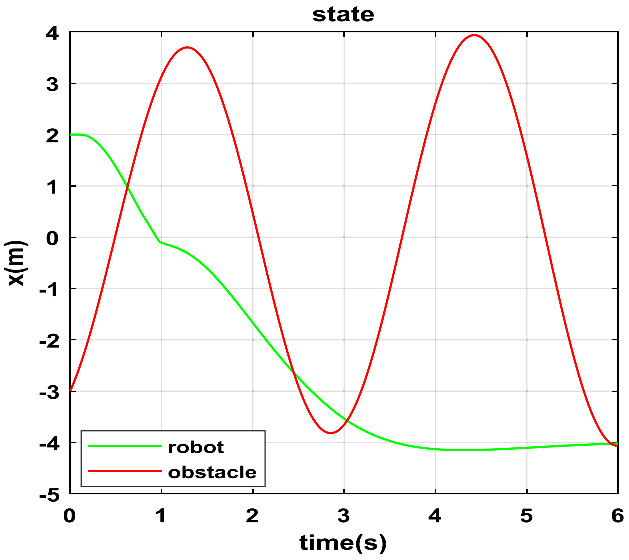

Fig. 64 Behavior of variable $\boldsymbol{x}$ LQR controller

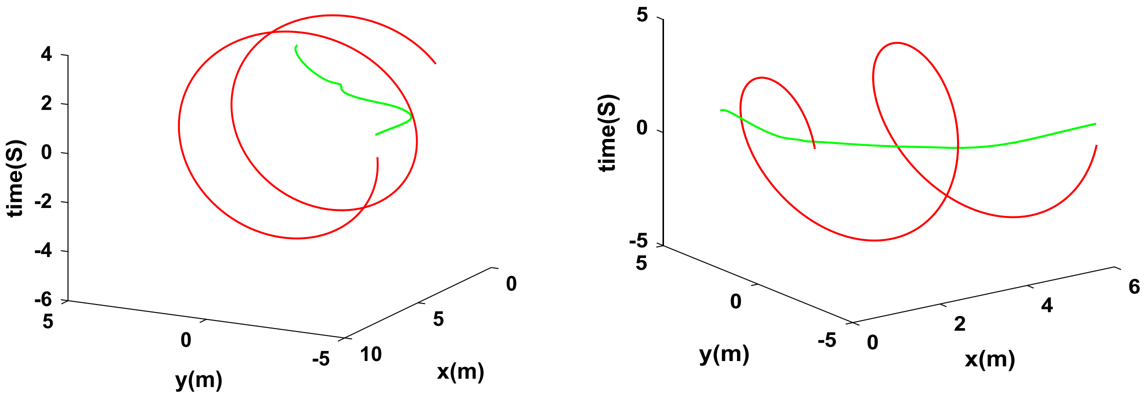

Fig. 63 Robot's routing (green line), moving obstacle (red line)—LQR controller

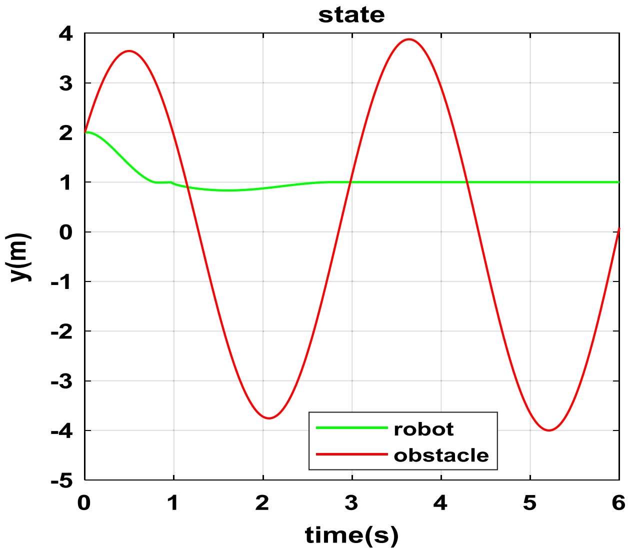

Fig. 65 Behavior of variable ***y*** LQR controller

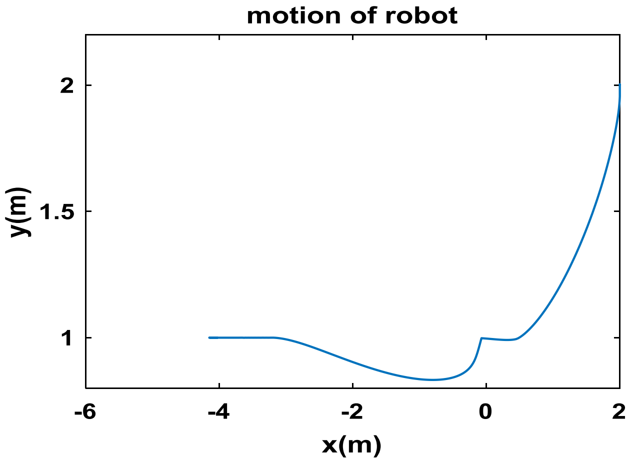

Fig. 66 The route travelled by the robot, LQR controller

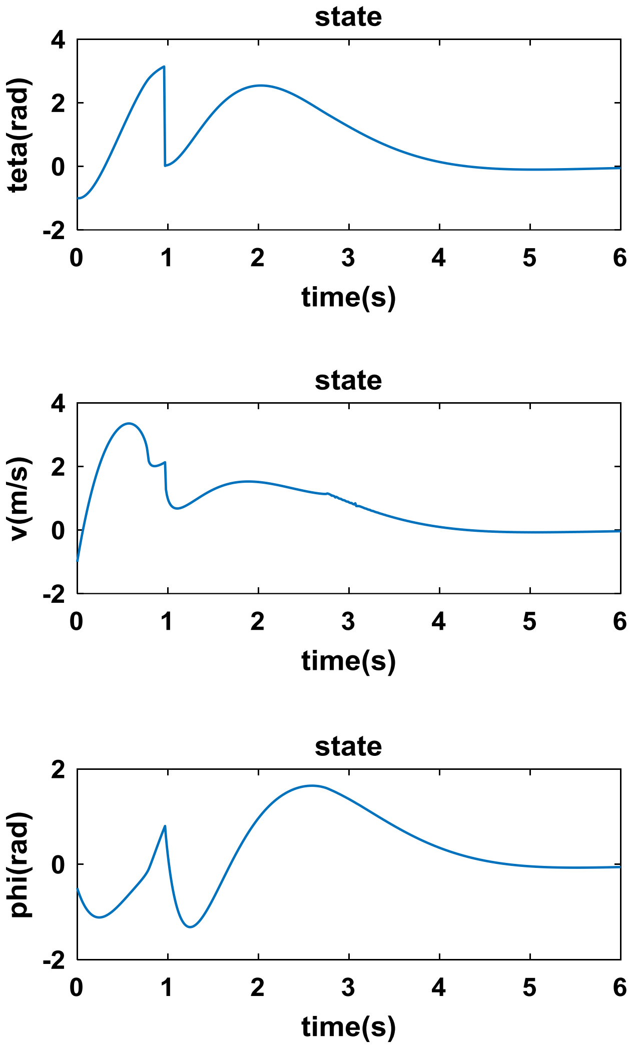

Fig. 67 Behavior of variables θ, v and φ, LQR controller

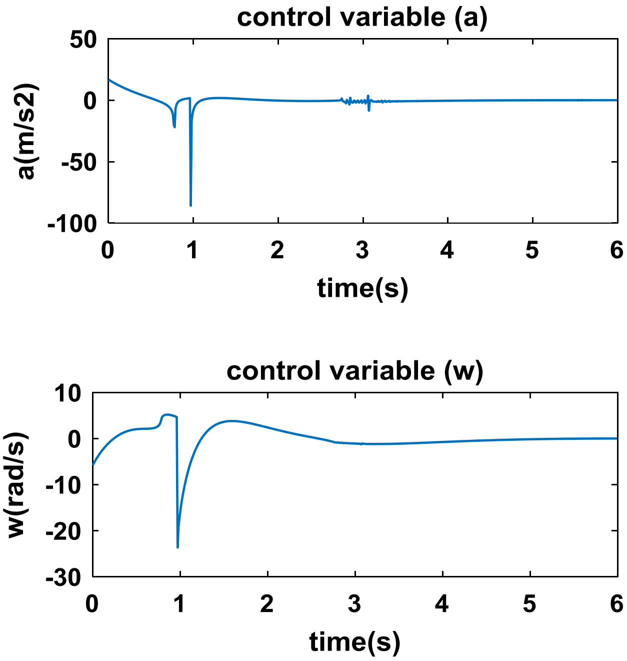

Fig. 68 Behavior of variables a and ω, LQR controller

References

1. Lumelsky VJ, Skewis T (1990) Incorporating range sensing in the robot navigation function. IEEE Trans Syst Man Cybern 20(5):1058–1069
2. Kamon I, Rivlin E, Rimon E (1996) A new range-sensor based globally convergent navigation algorithm for mobile robots. In: Proceedings of 1996 IEEE international conference on robotics and automation, 1996. IEEE, pp 429–435
3. Lumelsky VJ, Stepanov AA (1990) Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Autonomous robot vehicles:363–390
4. Borenstein J, Koren Y (1991) The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Trans Robot Autom 7(3):278–288
5. Sezer V, Gokasan M (2012) A novel obstacle avoidance algorithm: "Follow the Gap Method." Robot Autonom Syst 60(9):1123–1134
6. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. Int J Robot Res 5(1):90–98
7. Lee PS, Wang LL (1994) Collision avoidance by fuzzy logic control for automated guided vehicle navigation. J Field Robot 11(8):743–760
8. Reignier P (1994) Fuzzy logic techniques for mobile robot obstacle avoidance. Robot Autonom Syst 12(3–4):143–153
9. Cimen T (2008) State-dependent Riccati equation (SDRE) control: a survey. IFAC Proc Vol 41(2):3761–3775
10. Pearson J (1962) Approximation methods in optimal control I. Sub-optimal control. Int J Electron 13(5):453–469
11. Wernli A, Cook G (1975) Suboptimal control for the nonlinear quadratic regulator problem. Automatica 11(1):75–84
12. Cloutier JR, D'Souza CN, Mracek CP (1996) Nonlinear regulation and nonlinear H∞ control via the state-dependent Riccati equation technique: part 1, theory. In: Proceedings of the first international conference on nonlinear problems in aviation and aerospace, 1996. Embry-Riddle Aeronautical University Press Daytona Beach, FL, pp 117–130
13. Mracek CP, Cloutier JR (1998) Control designs for the nonlinear benchmark problem via the state-dependent Riccati equation method. Int J Robust Nonlinear Control 8(4–5):401–433
14. Cloutier JR, Stansbery DT (2002) The capabilities and art of state-dependent Riccati equation-based design. In: Proceedings of the American control conference, 2002. IEEE, pp 86–91
15. Roudkenary KA, Khaloozadeh H, Sedigh AK (2016) SDRE control of non-affine systems. In: 2016 4th international conference on control, instrumentation, and automation (ICCIA). IEEE, pp 239–244
16. Cloutier JR, Stansbery DT (2001) Nonlinear, hybrid bank-to-turn/skid-to-turn missile autopilot design. AIAA guidance, navigation, and control conference and exhibit, Montreal, Canada, pp 6–9
17. Parrish DK, Ridgely DB (1997) Attitude control of a satellite using the SDRE method. In: Proceedings of the American control conference, 1997. IEEE, pp 942–946
18. Parrish DK, Ridgely DB (1997) Control of an artificial human pancreas using the SDRE method. In: Proceedings of the American control conference, 1997.. IEEE, pp 1059–1060
19. Erdem EB, Alleyne AG (2001) Experimental real-time SDRE control of an underactuated robot. In: Proceedings of the 40th IEEE conference on decision and control, 2001. IEEE, pp 2986–2991
20. Korayem M, Zehfroosh A, Tourajizadeh H, Manteghi S (2014) Optimal motion planning of non-linear dynamic systems in the presence of obstacles and moving boundaries using SDRE: application on cable-suspended robot. Nonlinear Dyn 76(2):1423–1441
21. Korayem MH, Nekoo SR (2016) The SDRE control of mobile base cooperative manipulators: collision free path planning and moving obstacle avoidance. Robot Autonom Syst 86:86–105
22. Acosta L, Marichal G, Moreno L, Méndez JA, Rodrigo J (2000) Obstacle avoidance using the human operator experience for a mobile robot. J Intell Robot Syst 27(4):305–319
23. Belkhous S, Azzouz A, Saad M, Nerguizian C, Nerguizian V (2005) A novel approach for mobile robot navigation with dynamic obstacles avoidance. J Intell Robot Syst 44(3):187–201
24. Cetin O, Yilmaz G (2016) Real-time autonomous UAV formation flight with collision and obstacle avoidance in unknown environment. J Intell Robot Syst 84(1–4):415–433
25. Magree D, Mooney JG, Johnson EN (2014) Monocular visual mapping for obstacle avoidance on UAVs. J Intell Robot Syst 74(1–2):17–26
26. Cherubini A, Grechanichenko B, Spindler F, Chaumette F (2013) Avoiding moving obstacles during visual navigation. In: 2013 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3069–3074
27. Yamada T, Sa YL, Ohya A (2013) Moving obstacle avoidance for mobile robot moving on designated path. In: 2013 10th international conference on ubiquitous robots and ambient intelligence (URAI). IEEE, pp 509–514
28. Ishihara H, Hashimoto E (2013) Moving obstacle avoidance for the mobile robot using the probabilistic inference. In: 2013 IEEE international conference on mechatronics and automation (ICMA). IEEE, pp 1771–1776
29. Palm R, Bouguerra A (2011) Navigation of mobile robots by potential field methods and market-based optimization. In: 5th European conference on mobile robots, ECMR 2011, September 7–9, 2011, Örebro, Sweden, pp 207–212
30. Mohammadi SS, Khaloozadeh H (2016) Optimal motion planning of unmanned ground vehicle using SDRE controller in the presence of obstacles. In: 2016 4th international conference on control, instrumentation, and automation (ICCIA). IEEE, pp 167–171

31. Rostami SMH, Sangaiah AK, Wang J, Kim H-j (2018) Real-time obstacle avoidance of mobile robots using state-dependent Riccati equation approach. EURASIP J Image Video Process 1:79
32. Hammett KD, Hall CD, Ridgely DB (1998) Controllability issues in nonlinear state-dependent Riccati equation control. J Guid Control, Dyn 21(5):767–773
33. Li G, Yamashita A, Asama H, Tamura Y (2013) An efficient improved artificial potential field based regression search method for robot path planning. In: 2012 IEEE international conference on mechatronics and automation. IEEE, pp 1227–1232
34. Hsu D, Kindel R, Latombe J-C, Rock S (2002) Randomized kinodynamic motion planning with moving obstacles. Int J Robot Res 21(3):233–255
35. Yashiro Y, Eguchi K, Iwasaki S, Yamauchi Y, Nakata M (2014) Development of obstacle avoidance control for robotic products using potential method. Mitsubishi Heavy Ind Tech Rev 51(1):34

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.