

RESEARCH

Open Access



Real-time obstacle avoidance of mobile robots using state-dependent Riccati equation approach

Seyyed Mohammad Hosseini Rostami¹, Arun Kumar Sangaiah², Jin Wang³ and Hye-jin Kim^{4*}

Abstract

In this paper, state-dependent Riccati equation (SDRE) method-based optimal control technique is applied to a robot. In recent years, issues associated with the robotics have become one of the developing fields of research. Accordingly, intelligent robots have been embraced greatly; however, control and navigation of those robots are not easy tasks as collision avoidance of stationary obstacles to doing a safe routing has to be taken care of. A moving robot in a certain time has to reach the specified goals. The robot in each time step needs to identify criteria such as velocity, safety, environment, and distance in respect to defined goals and then calculate the proper control strategy. Moreover, getting information associated with the environment to avoid obstacles, do the optimal routing, and identify the environment is necessary. The robot must intelligently perceive and act using adequate algorithms to manage required control and navigation issues. In this paper, smart navigation of a mobile robot in an environment with certain stationary obstacles (known to the robot) and optimal routing through Riccati equation depending on SDRE is considered. This approach enables the robot to do the optimal path planning in static environments. In the end, the answer SDRE controller with the answer linear quadratic controller will be compared. The results show that the proposed SDRE strategy leads to an efficient control law by which the robot avoids obstacles and moves from an arbitrary initial point $\times 0$ to a target point. The robust performance of SDRE method for a robot to avoid obstacles and reach the target is demonstrated via simulations and experiments. Simulations are done using MATLAB software.

Keywords: Optimal control, Nonlinear control, Obstacle avoidance, SDRE

1 Introduction

A mobile intelligent robot is a useful tool which can lead to the target and at the same time avoid an obstacle when faced with it. Obstacle avoidance means that the robot avoids colliding with obstacles such as fixed objects or moving objects. So when a robot encounters with an obstacle, it must decide to avoid it and at the same time, consider the most efficient path to the target with a good decision; this decision is performed in this research using the state-dependent Riccati equation (SDRE) control method. The optimal control of nonlinear systems cannot be done similar to the methods of linear systems. One of the significant viewpoints for

optimal control of nonlinear systems is using the SDRE. This method in addition to stability creates a proper functioning and robust for a wide range of nonlinear systems. Nonlinear optimal controller SDRE is a developed linear optimal control LQR, in which equations Riccati is state-dependent. Strategy SDRE has been introduced in the past decade. This strategy is a very effective algorithm for feedback nonlinear analysis that its states are nonlinear; however, a flexibility idea of through the weight matrix is dependent on the state. This involves factoring (ie, parameterization) of nonlinear dynamics to state vector and produces a value function for the matrix which is dependent on its state.

There are conventional methods of obstacle avoidance such as the path planning method [1], the navigation function method [2], and the optimal regulator [3]. Hence, an SDRE regulator [4, 5] is used for this

* Correspondence: hye-jinkim@hotmail.com

⁴Business Administration Research Institute, Sungshin Women's University, Seoul, Republic of Korea

Full list of author information is available at the end of the article

paper. In recent years, many researchers have investigated the obstacle avoidance problem from different perspectives. SDRE technique developed as a design method, which provides a systematic and effective design of nonlinear controllers, filters, and observers. Because of its versatile features, SDRE is broadly used for different cases [6–14]. SDRE can also be used in the field of medicine, for example, in [15, 16] the presentation of the optimal chemotherapy protocol for cancer treatment, considering the metastasis, has used the full optimal SDRE feedback. In [17], the SDRE algorithm is studied on motion design of the cable-suspended robot with uncertainties and moving obstacles. A method for controlling the tracing of a robot has been developed by the SDRE. Vibration control of flexible-link manipulator in [18] is used in SDRE controller and Kalman filtering. The problem of estimating flexural states during the use of the SDRE for flexible control is the focus of this paper. Wanga et al. [19] present a novel $H_2 - H_\infty$ -SDRE control approach with the purpose of providing a more effective control design framework for continuous-time nonlinear systems to achieve a mixed nonlinear quadratic regulator and H_∞ -control performance criteria.

It should be noted that the algorithms discussed for avoiding an obstacle are different from each other, and each algorithm has proposed a separate and new method for avoiding obstacles. So far, novel methods to avoid obstacles in addition to the mentioned methods are also presented [20–23].

This paper is organized as follows: a description of the SDRE formulation is given in Section 2. Then, motion equations of the robot and its constraints are presented in Section 3. In Section 4, SDRE formulation of the robot is derived. And simulation results have been reported in Section 5. Finally, in Section 6, some conclusions are drawn from this research.

2 Methods

2.1 Problem formulation

Consider a nonlinear system that is autonomous, full-state observable and input affine that is presented as follows:

$$\dot{x}(t) = f(x) + B(x)u(t), x(0) = x_0 \tag{1}$$

Where $x \in R^n$ is the state vector, $u \in R^m$ is the input vector, and $t \in [0, \infty)$, with functions $f: R^n \rightarrow R^n$ and $B: R^n \rightarrow R^{n \times m}$, and $B(x) \neq 0 \forall x$. Without any loss of generality, the origin is assumed to be an equilibrium point, such that $f(0) = 0$. In this context, the minimization of the infinite-time performance criterion is as follows:

$$J(x_0, u(\cdot)) = \frac{1}{2} \int \{x^T(t)Q(x)x(t) + u^T(t)R(x)u(t)\}dt \tag{2}$$

Consider the above equation which is non-quadratic in x but quadratic in u . The state and input weighting matrices are assumed state-dependent such that $Q: R^n \rightarrow R^{n \times n}$ and $R: R^n \rightarrow R^{n \times m}$. These design parameters satisfy $Q(x) \geq 0$ and $R(x) > 0$ for all x [24].

2.2 SDRE nonlinear regulator problem

The SDRE methodology uses extended linearization as the key design concept in formulating the nonlinear optimal control problem. The linear control synthesis method for this case is the LQR synthesis method. SDRE feedback control is an “extended linearization control method” that provides a similar approach to the nonlinear regulation problem for the input-affine system (1) with cost functional (2). The nonlinear system of (1) can be rewritten in the linear structure using SDC form:

$$\dot{x}(t) = A(x)x(t) + B(x)u(t) \tag{3}$$

SDC form is not unique and should be chosen in such a way that $\{Q^{1/2}(x), A(x)\}$ and $\{A(x), B(x)\}$ are pointwise observable and pointwise stabilizable, respectively.

Where $W(x)$ is obtained from the solution of algebraic SDRE:

$$W(x)A(x) + A^T(x)W(x) + Q(x) - W(x)B(x)R^{-1}B^T(x)W(x) = 0 \tag{4}$$

to get $W(x) \geq 0$.

Where $W(x)$ is the unique, symmetric, positive-definite solution of the algebraic state-dependent Riccati equation. As a result of this, the nonlinear control is obtained as:

$$u(t) = -R^{-1}(x)B^T(x)W(x)x(t) \tag{5}$$

The response obtained is locally asymptotically stable and is optimal [25].

2.3 SDRE nonlinear tracking problem

Consider an infinite-time nonlinear tracking problem. Minimize the following nonlinear cost function that is nonquadratic in x but is quadratic in u [26]:

$$J = \frac{1}{2} \int_0^\infty \{E^T(t)Q(x)E(t) + u^T(t)R(x)u(t)\}dt \tag{6}$$

in which

$$E(t) = X(t) - X_{desired}(t) \tag{7}$$

$Q, R, x,$ and u are like the previous model and with the same conditions. By solving the Eq. (6), the input is obtained as the following equation:

$$u(t) = -R^{-1}(x)B^T(x)W(x)(X - X_d) \tag{8}$$

3 The four-wheeled car

Here is considered a four-wheeled robot with front-wheel drive as Fig. 1. In this robot model, R_{tu} instantaneous turning radius, L length between the rear axle and the front of the robot, and (x, y) the robot's position is measured at the center point of the rear wheels of the robot. The state vector consists of two-position variables (x, y) , one orientation variable (θ) , the speed of the robot (V) , and front-wheel drive angle (α) . Only the front wheels swivel in two directions, and the rear wheels swivel without slipping. The main reason for choosing this system for this project is the non-holonomic boundary movement of this system [27].

Here, (V) and (θ) are considered as input vectors, but using these control inputs, this system is faced with solutions that are not smooth. It is very obvious that in practice mode, the speed of the vehicle's robot and the wheel angle cannot change quickly and instantaneously. Therefore, to smooth the (V, θ) controls and make it possible to apply inputs to a real robot, a control level is added. In other words, (V, θ) is recorded as part of the state vector and (β, ψ) is added as a new control vector. These changes create smoother paths for (V, θ) that can be used in a real robot. Finally, with regard to all these

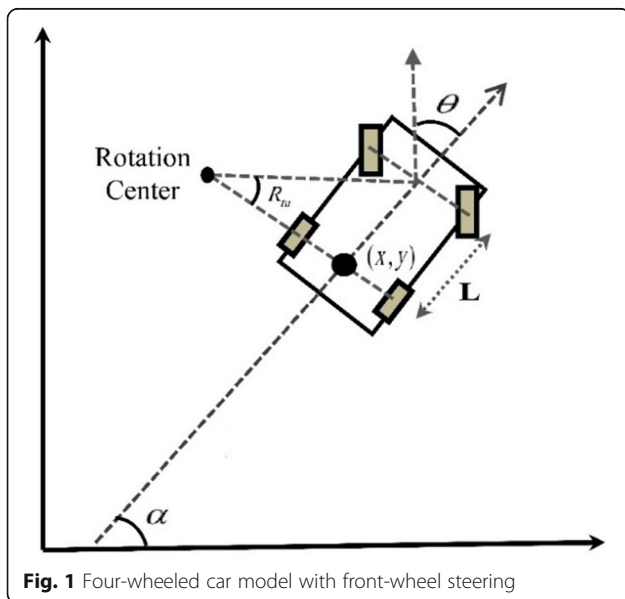


Fig. 1 Four-wheeled car model with front-wheel steering

points, the equation of motion for the robot kinematics derived as follows [27].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \\ \dot{V} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V \cos\alpha \\ V \sin\alpha \\ \frac{V}{L} \tan\theta \\ \beta \\ \psi \end{bmatrix} \tag{9}$$

3.1 Vehicle's constraints

The constraints of the state and control inputs are as follows:

$$x \in X = \left\{ \begin{array}{l} x : 0 \leq x(t) \leq 10 \\ y : 0 \leq y(t) \leq 10 \\ \alpha : -3\pi \leq \alpha \leq 3\pi \\ V : -30 \leq V \leq +30 \\ \theta : -1 \leq \theta \leq +1 \end{array} \right\} \tag{10}$$

$$u \in U = \left\{ \begin{array}{l} \beta : -10 \leq \alpha(t) \leq 10 \\ \psi : -0.33 \leq \psi(t) \leq 0.33 \end{array} \right\}$$

So far, dynamic equations of the robot and corresponding constraints on states and control input have been presented. In the following section, the proper formulation for utilizing SDRE approach for robot motion control will be presented.

4 Implementation of SDRE controller for the robot

4.1 SDC parameterization

The first step in designing the SDRE controller is to formulate the SDC. The most important point in SDC parameterization is that factorization should provide appropriate control over the entire area. SDC presentation of the system is written as follows [28].

$$A(x) = \begin{bmatrix} 0 & 0 & \beta_{13} & \beta_{14} & 0 \\ 0 & 0 & \beta_{23} & \beta_{24} & 0 \\ 0 & 0 & 0 & \beta_{34} & \beta_{35} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} B(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{11}$$

$$\beta_{13} = \frac{\beta_1 V (\cos\alpha - 1)}{\alpha}, \beta_{14} = (1 - \beta_1) (\cos\alpha - 1) + 1$$

$$\beta_{23} = \beta_2 V \sin\alpha, \beta_{24} = (1 - \beta_2) \sin\alpha$$

$$\beta_{34} = (1 - \beta_3) \frac{\tan\theta}{L}, \beta_{44} = \beta_3 \left(\frac{V}{L}\right) \left(\frac{\tan\theta}{\theta}\right) \tag{12}$$

4.2 Obstacle avoidance

In this paper, the concept of the APF method and the navigation function [29] is used to avoid collisions with obstacles during the robot's movement. The navigation function that is similar to the APF, with the help of the sensor, will detect obstacles and avoid them and converge the robot toward the target. This function has the local minimum at the destination and a local maximum at the obstacles.

In the modified model, OBS can be decomposed into obs_a and obs_b . Where obs_a is the component force under the direction along the line between the robot and the obstacle, and obs_b is the component force under the direction along the line between the robot and the target. m is a weighting factor that one can set the distance of the robot from an obstacle during movement. By setting $mand z$, the desired result is achieved which is obstacle avoidance and reach the target. The proposed navigation function added to the cost function is as follows:

$$\begin{cases} R_{at} = \sqrt{(x-x_d)^2 + (y-y_d)^2} \\ R_{rep} = \sqrt{(x-x_{ob})^2 + (y-y_{ob})^2} - R_{ob} \\ \begin{cases} obs_a = m \left(\frac{R_{at}^z}{R_{rep}^2 + 0.1} \right) \\ obs_b = \left(\frac{R_{at}^{(z-1)}}{R_{rep}^2 + 0.1} \right) \end{cases} \\ OBS = obs_a + obs_b \end{cases} \quad (13)$$

The form of obstacles is a sphere with center (x_{ob}, y_{ob}) and radius R_{ob} , and it is supposed that they do not overlap each other. The purpose of this study is to design a

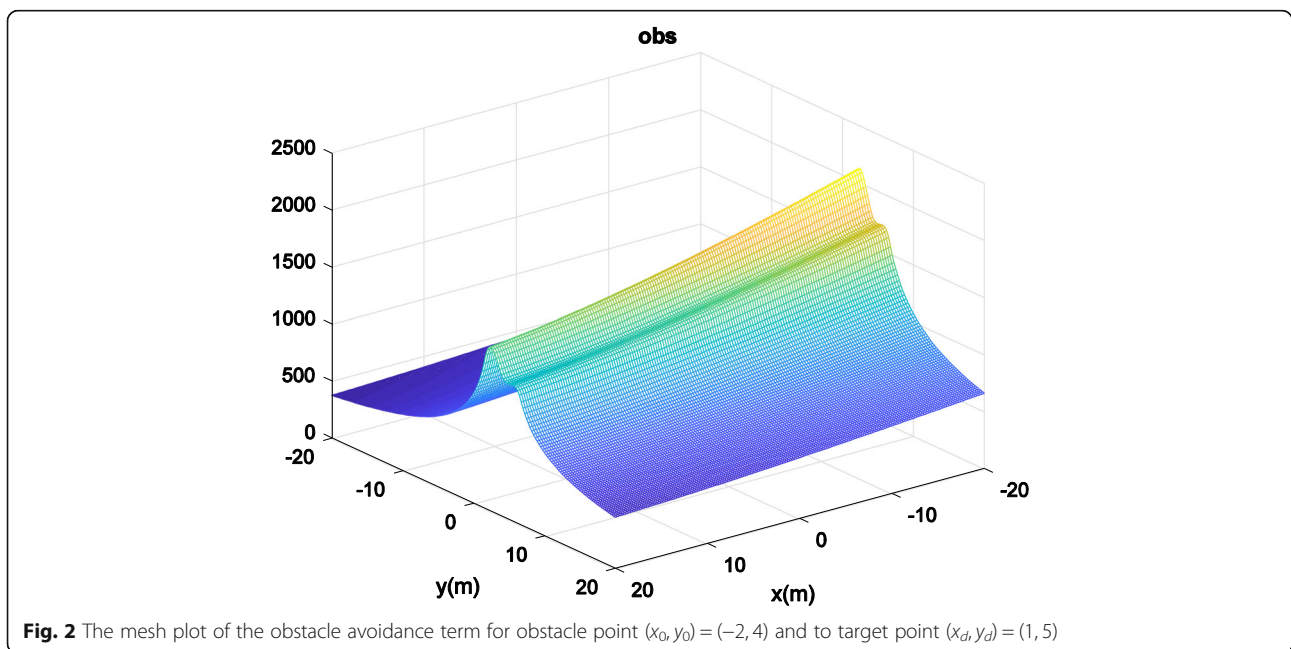
control law by which the controlled object avoids obstacles and moves from an arbitrary initial point x_0 to the destination point x_d . In the next step, this obstacle avoidance equation is added to the cost function. The purpose here is to minimize the performance of the criterion function that is considered as follows:

$$J = \frac{1}{2} \int_0^\infty \{E^T(t)Q(x)E(t) + u^T(t)R(x)u(t) + OBS\} dt \quad (14)$$

Then, using the factoring, the matrix $Q(x)$ is obtained as follows:

$$Q(x) = \text{diag} \left(\left[\frac{1}{E_X^2} (OBS), \frac{1}{E_Y^2} (OBS), 1, 1, 1 \right] \right) \begin{cases} E_X = x-x_d \\ E_Y = y-y_d \end{cases} \quad (15)$$

E_X and E_Y are errors of position. $Q(x)$ is not a single matrix and many matrices are obtained for it. By using this formulation, the robot can now start from an initial point and reach a destination while avoiding obstacles. The useful property of the obstacle avoidance term in the cost function led to the increase in the cost function while the robot is nearing obstacle and decreasing when leaving it. Figure 2 shows mesh plot of obstacle avoidance term in which the function has the global minimum at the destination and a local maximum at the obstacle. By rewriting (15), the second obstacle in the motion planning algorithm can be constructed as follows.



$$Q(x) = \text{diag}\left(\left[\frac{1}{E_x^2}(\text{OBS}_1 + \text{OBS}_2), \frac{1}{E_y^2}(\text{OBS}_1 + \text{OBS}_2), 1, 1, 1\right]\right) \tag{16}$$

5 Results and discussion

In this section, the effectiveness of the proposed method is verified by the simulation model of the robot. The general parameters include all states are as follows:

$$\begin{aligned} \beta_1 &= 0.715; \beta_2 = 0.8; \beta_3 = 0.5 \\ R &= \text{diag}([1, 1]), L = 0.5 \end{aligned} \tag{17}$$

Figure 2 shows the mesh plot of the obstacle avoidance term in which the function has the global minimum at the destination and a local maximum at the obstacle. It shows that the obstacles like the summit and the target act like the cavity and absorbs the robot into the inside itself.

In Fig. 3, obstacles have been considered as a circle of radius 0.4 and show that with SDRE method, the robot avoids obstacles and reaches the target with minimal cost. When the robot is faced by an obstacle; Q parameter is designed so that increases the cost function and in other words, finds the highest possible value to avoid colliding with the obstacle. Here, to avoid collision with the obstacle and reach the desired target, the value of m and z in the navigation obsis respectively put 5 and 2.

Figure 4 shows two variables x and y in reaching the target point $x_d = (1, 5)$, and these figures show that the

robot stops as soon as reaching to the target. Figure 5 shows that other states also regulated to zero after 3.1 s. Figure 6 shows control inputs that are applied to reach the desired target, after reaching the target became zero. The purpose of this control method is to find the control inputs that are applied to the system with Eq. (11); while stabilizing the system and satisfying the constraints defined for it, the defined cost function (16) is minimized, and the system state variables converge to zero with the least control effort. The figures obtained from the simulation show that SDRE method has achieved the desired goals. By comparing this method with the modified APF method, it is observed that this method selects a more optimal route according to the definition of the cost function for avoiding the obstacle and reaching the goal. The other difference is that in SDRE method, three parameters are added: vehicle velocity (V), the angle of the front wheels (α), and length of the car between the front and rear axles (L), to design appropriate control function in order to avoidance obstacles and reach the desired target.

Without taking navigation function and supposing $Q(x) = \text{diag}[100, 100, 1, 1, 1]$, Fig. 7 shows that the robot cannot reach the target with obstacle avoidance. Figure 8 of the LQR method in which a nonlinear system is linearized around the vector $X_v = [1, 1, 1, 1, 1]^T$ and initial conditions $X_p = [-3, 3, 0.0001, 0.0001, 0.0001]^T$. Figure 9 shows that although the control inputs became zero, as Fig. 8 depicts, the robot cannot avoid obstacles in reaching to the target. Like Fig. 8, in LQR method, with linearization of around none of the operating points and

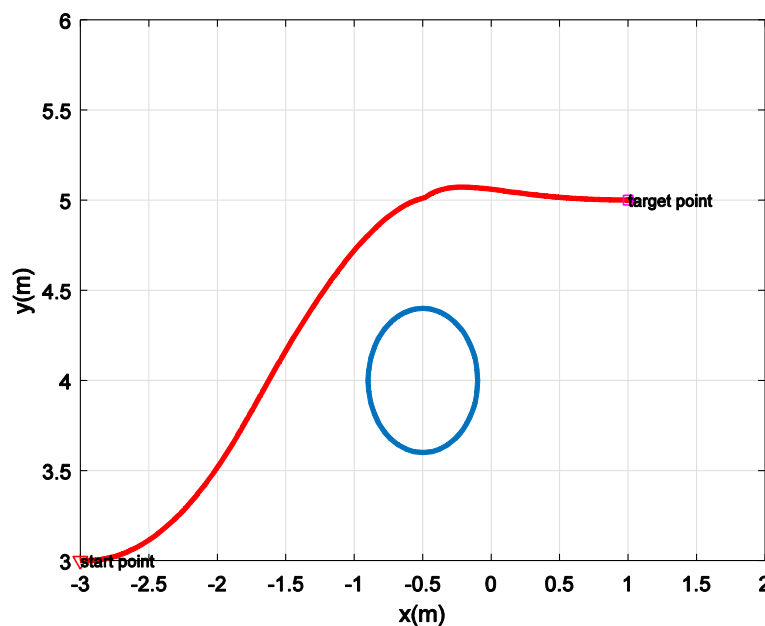


Fig. 3 Path of the robot with the obstacle avoidance term in presence obstacle point $(x_0, y_0) = (-2, 4)$ and target point $(x_d, y_d) = (1, 5)$ with SDRE method

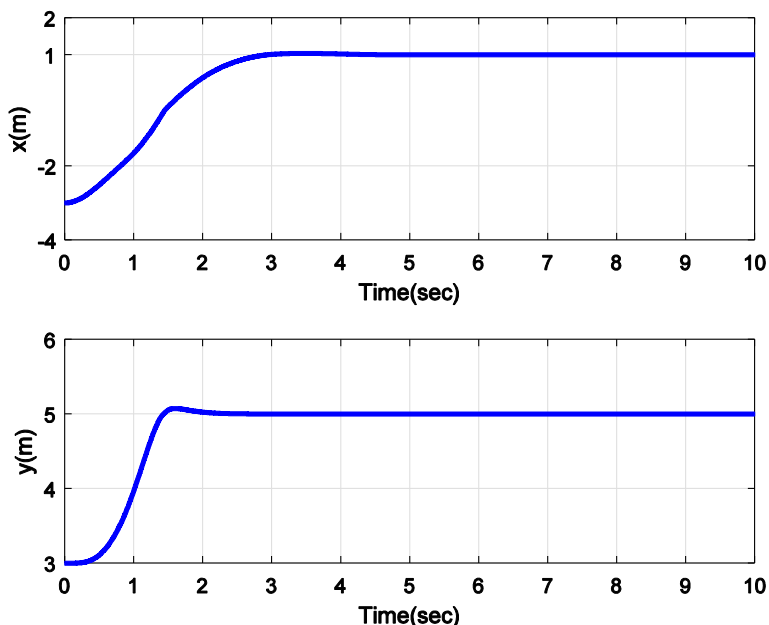


Fig. 4 x and y trajectories in presence obstacle point $(x_o, y_o) = (-2, 4)$ and target point $(x_d, y_d) = (1, 5)$ while avoiding an obstacle with SDRE method

initial conditions, the robot cannot avoid obstacles in reaching to the target, so is concluded, by SDRE method where the matrix A depends on its states, the answer is better.

In the second section, the effectiveness of the proposed method is verified by the simulation model of the robot. The general parameters include all states are as follows:

$$\begin{aligned} \beta_1 &= 0.1; \beta_2 = 0.78; \beta_3 = 0.88 \\ R &= \text{diag}([1, 1]), L = 0.5 \end{aligned} \tag{18}$$

Figure 10 shows the mesh plot of the obstacle's avoidance term in which the function has the global minimum at the destination and a local maximum at the obstacle. It shows that the obstacles like the summit and the target act like the cavity and absorbs

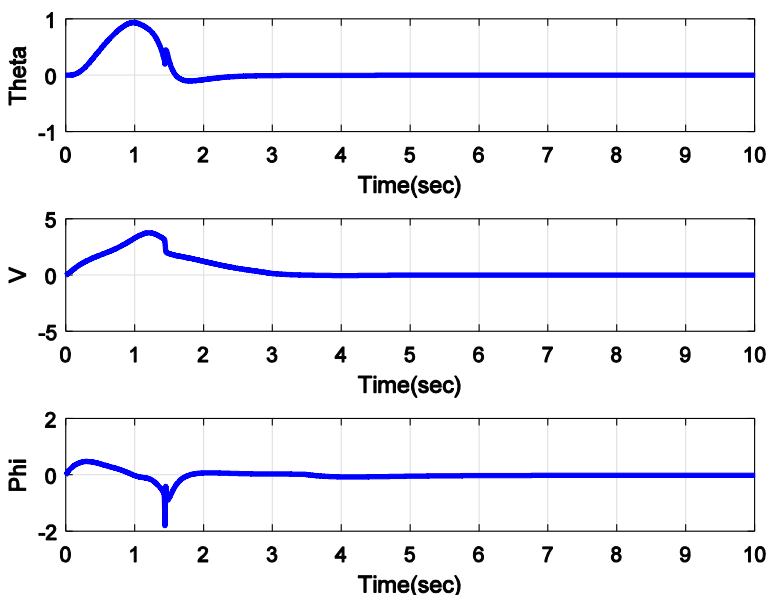


Fig. 5 θ , V and ϕ trajectories in presence obstacle point $(x_o, y_o) = (-2, 4)$ and target point $(x_d, y_d) = (1, 5)$ while avoiding an obstacle with SDRE method

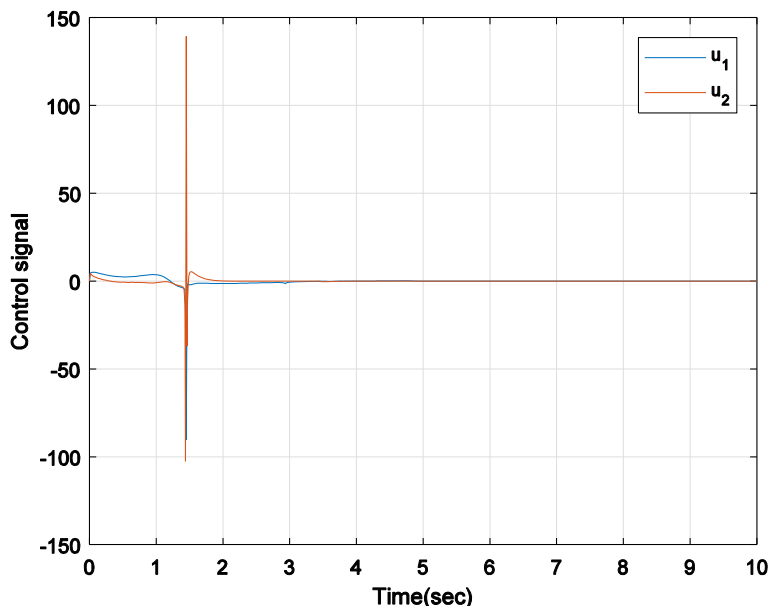


Fig. 6 Control inputs applied to the robot to reach the target point $(x_d, y_d) = (1, 5)$ and in presence obstacle avoidance point $(x_0, y_0) = (-2, 4)$ with SDRE method

the robot into the inside itself. Here, to avoid collision with the obstacle and reach the desired target, the value of m and z in the navigation *obs* is respectively put 0.5 and 2. In the face of obstacles more than an obstacle. The principles of work for other obstacles are like one obstacle, but in this case, a

navigation function is put. For example, in Fig. 11, two obstacles have been considered as a circle of radius 0.25 and shows that with SDRE method; the robot avoids obstacles and reaches the target with minimal cost. Figure 12 shows the curve of two variable's x and y in reaching the target point $x_d = (2, 5.5)$,

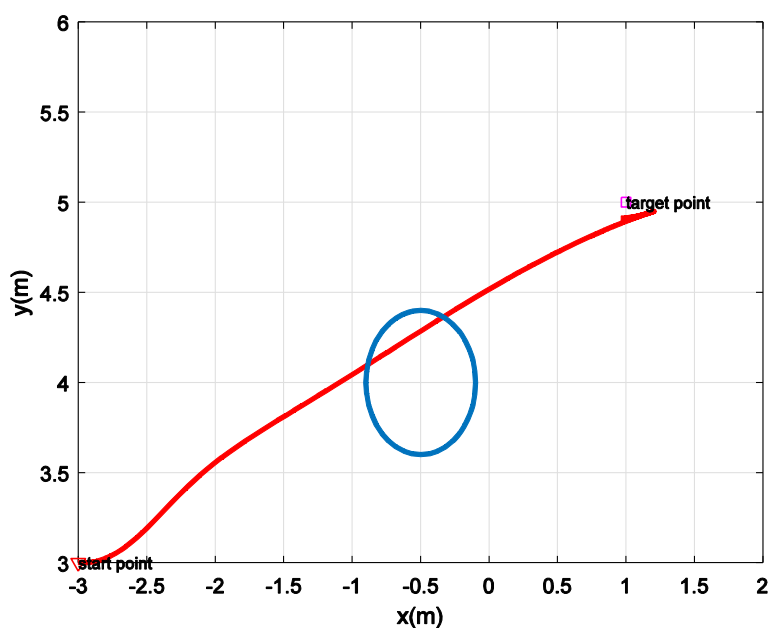


Fig. 7 Path of the robot without obstacle avoidance term and in presence obstacle point $(x_0, y_0) = (-2, 4)$ and target point $(x_d, y_d) = (1, 5)$

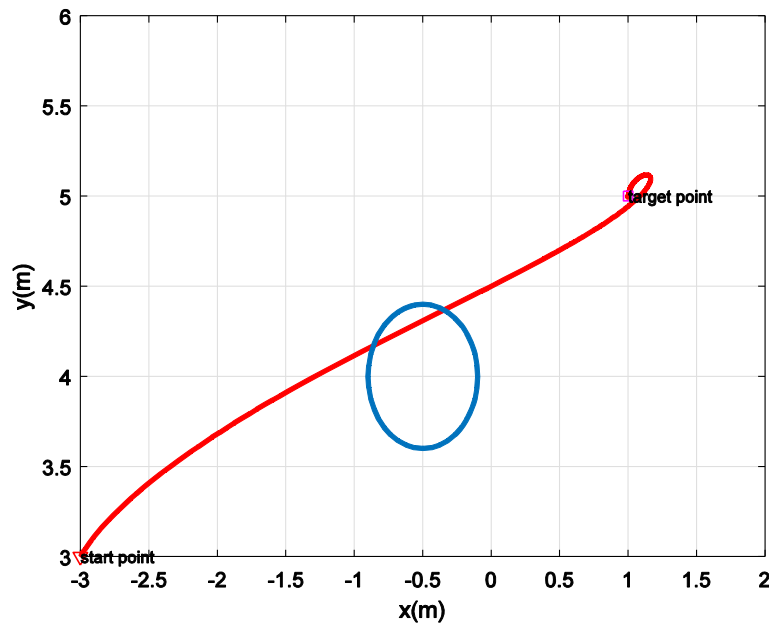


Fig. 8 Path of the robot to the target point $(x_d, y_d) = (1, 5)$ in presence obstacle point $(x_o, y_o) = (-2, 4)$ with LQR method

and these figures show that the robot stops as soon as reaching to the target. Figure 13 shows that other states also regulated to zero after 4.2 s. Figure 14 shows control inputs that are applied to reach the desired target, after reaching the target the control attempt becomes zero.

Without taking navigation function and supposing $Q(x) = \text{diag}[100, 100, 1, 1, 1]$, Fig 15 shows that the robot cannot

reach the target with obstacle avoidance. Figure 16 of the LQR method in which a nonlinear system is linearized around the vector $X_v = [1, 1, 1, 1, 1]^T$ and initial conditions $X_p = [-3, 3, 0.0001, 0.0001, 0.0001]^T$. Figure 17 shows that although the control inputs became zero, as Fig. 16 depicts, the robot cannot avoid obstacles in reaching to the target. Like Fig. 16, in LQR method, with linearization of around none of the operating points and initial conditions, the

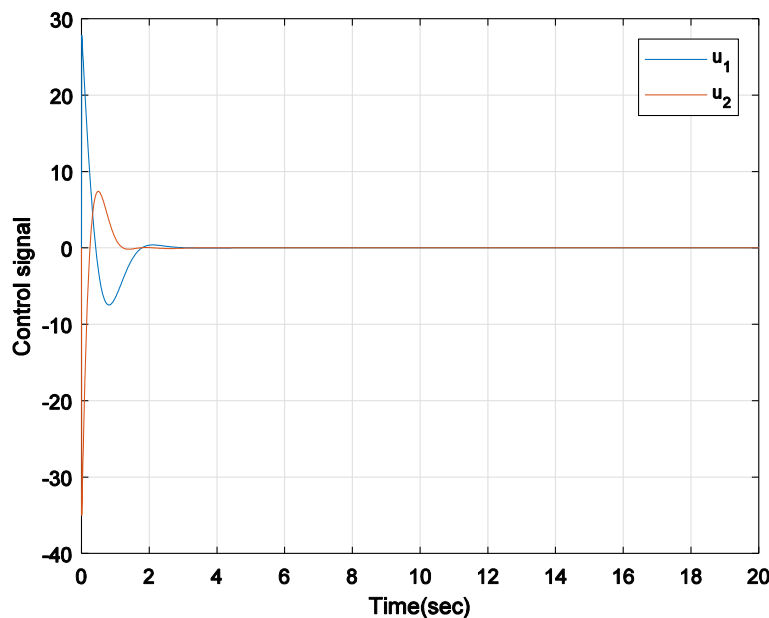
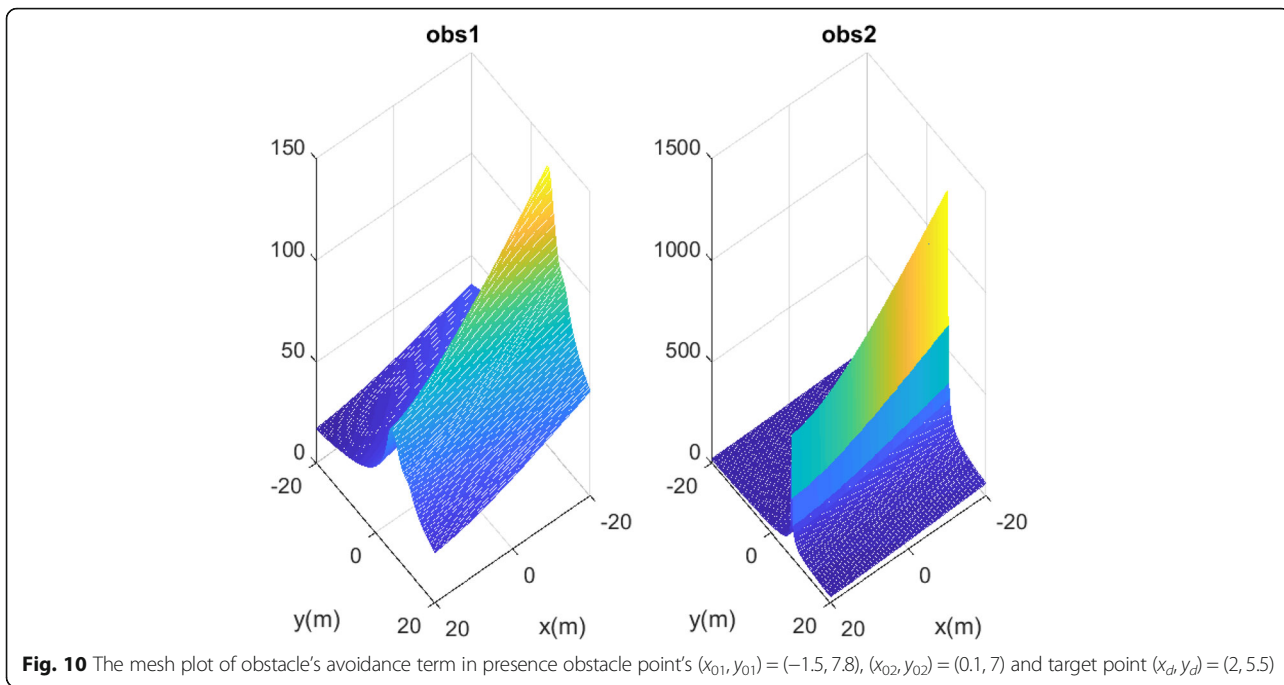


Fig. 9 Control inputs applied to the robot to reach the target point $(x_d, y_d) = (1, 5)$ and in presence obstacle point $(x_o, y_o) = (-2, 4)$ with LQR method

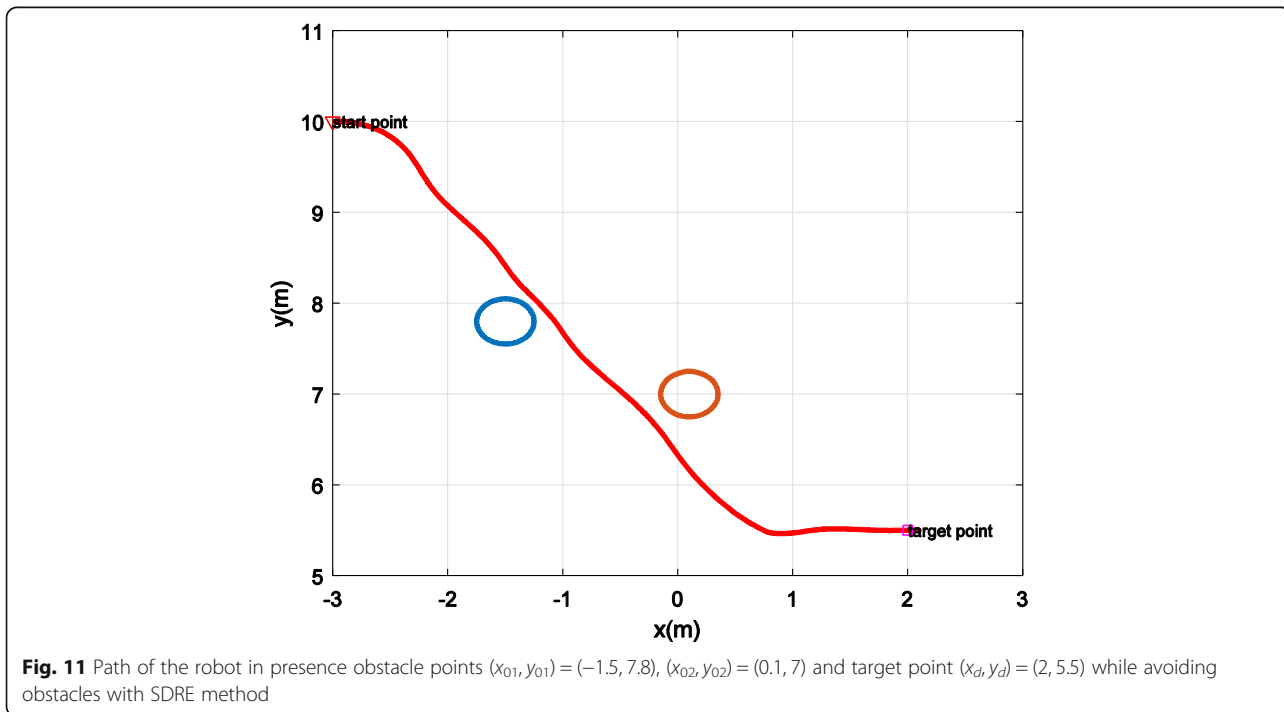


robot cannot avoid obstacles in reaching to the target, so is concluded, by SDRE method where the matrix A depends on its states, the answer is better.

6 Conclusions

This paper focuses on the SDRE nonlinear regulator for solving the nonlinear optimal control problems.

The existence of solutions as well as optimality and stability properties associated with SDRE controllers are the main of this paper. The paper is organized as follows. In Section 2, the formulation of the nonlinear optimal control problem, the concept of extended linearization and the SDRE controller for nonlinear optimal regulation are presented, then the



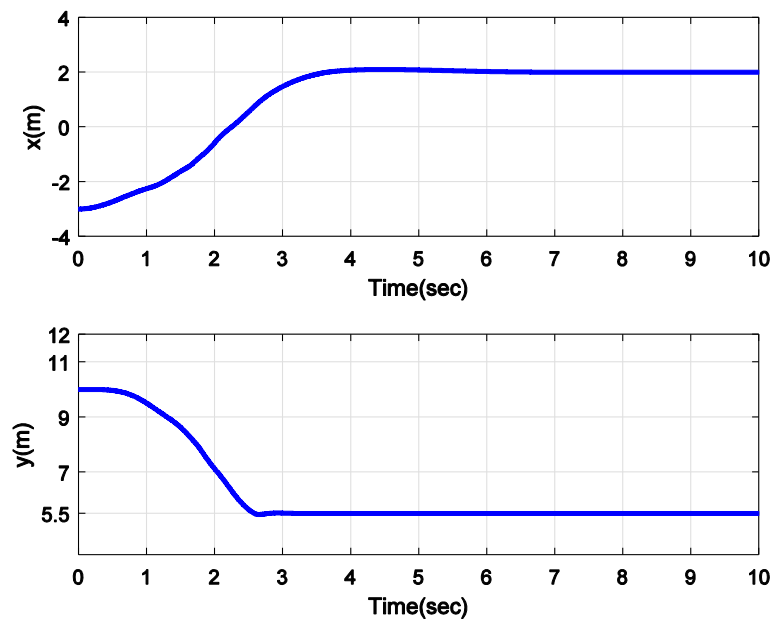


Fig. 12 x and y trajectories in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$, $(x_{02}, y_{02}) = (0.1, 7)$ and target point $(x_d, y_d) = (2, 5.5)$ while avoiding obstacles with SDRE method

additional degrees of freedom provided by the nonuniqueness of the SDC parameterization is reviewed. The necessary and sufficient conditions on the existence of solutions to the nonlinear optimal control problem, in particular, by SDRE feedback control, are reviewed. A theoretical study on the stability and

optimality properties of SDRE feedback controls is pursued. SDRE method is compared with LQR method. It is concluded for that SDRE is a semilinear method, give the best answer than linear methods for nonlinear systems, because the basis of this method is that takes nonlinearity of the system and creates the nonlinear system that its

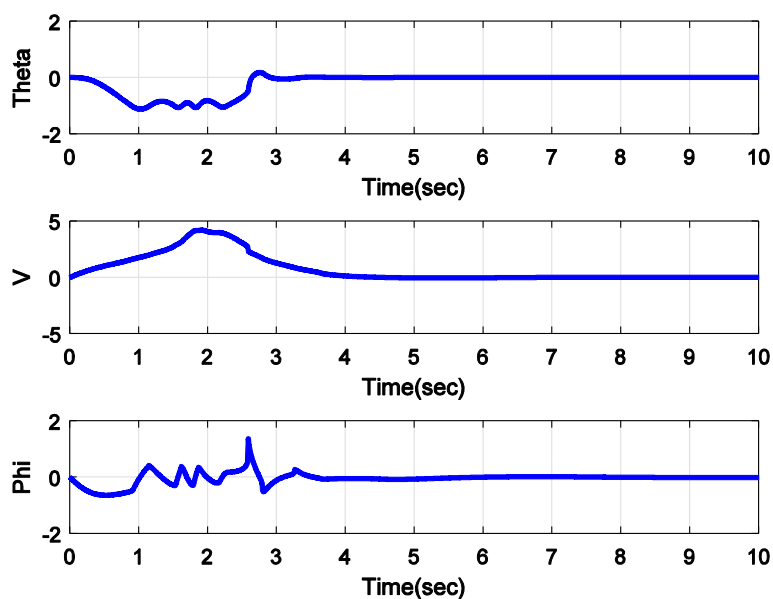


Fig. 13 θ , V and ϕ trajectories in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$, $(x_{02}, y_{02}) = (0.1, 7)$ and target point $(x_d, y_d) = (2, 5.5)$ while avoiding obstacles with SDRE method

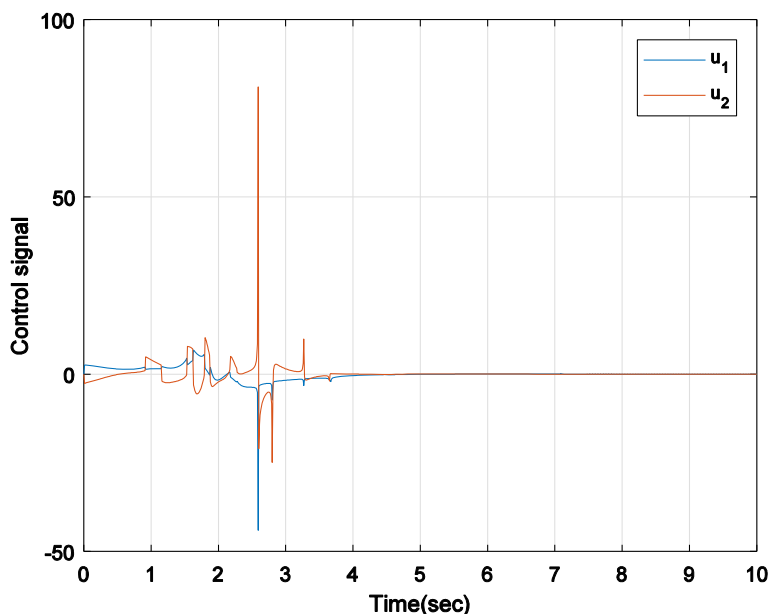


Fig. 14 Control inputs applied to the robot to reach the target point $(x_d, y_d) = (2, 5.5)$ in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$ and $(x_{02}, y_{02}) = (0.1, 7)$ with SDRE method

state-dependent coefficient matrix structure has semi-linear and minimum nonlinear performance index with semi-quadratic structure. The result is that contrary to controllers like LQR that first, they linearize nonlinear controllers, then the laws are designed for stability that

can cause to remove some of the important elements of nonlinear systems that have a key role in system stability. In the method of SDRE, the important elements are not removed from the nonlinear system and as a result, according to the results of the various articles, it is

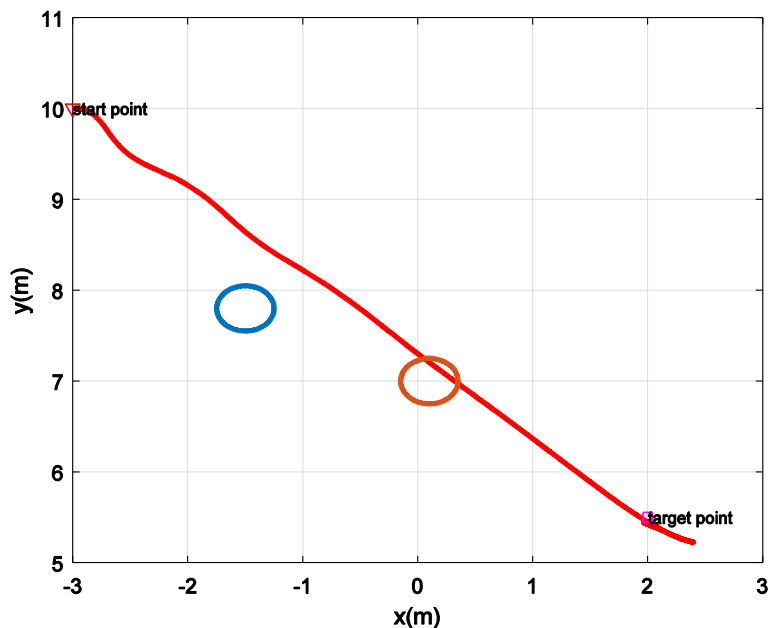


Fig. 15 Path of the robot without obstacle avoidance term in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$, $(x_{02}, y_{02}) = (0.1, 7)$ and target point $(x_d, y_d) = (2, 5.5)$

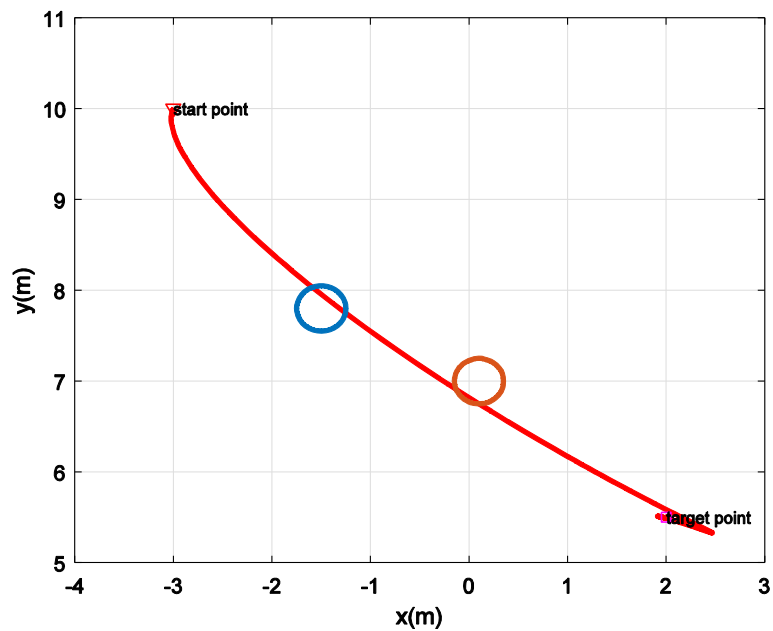


Fig. 16 Path of the robot to the target point $(x_d, y_d) = (2, 5.5)$ in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$ and $(x_{02}, y_{02}) = (0.1, 7)$ with LQR method

concluded that the system is robust against disturbances and uncertainties and achieves better performance than LQR. Our target in this research is finding an equation control using the method SDRE for routing a robot and avoid collisions with obstacles and path optimization for

the robot to reach the target. In the end, the proper suggestion for each robot's path planning is to create an optimal combination algorithm according to the specific structure of each robot so that each algorithm can be covered the constraints of the other algorithm.

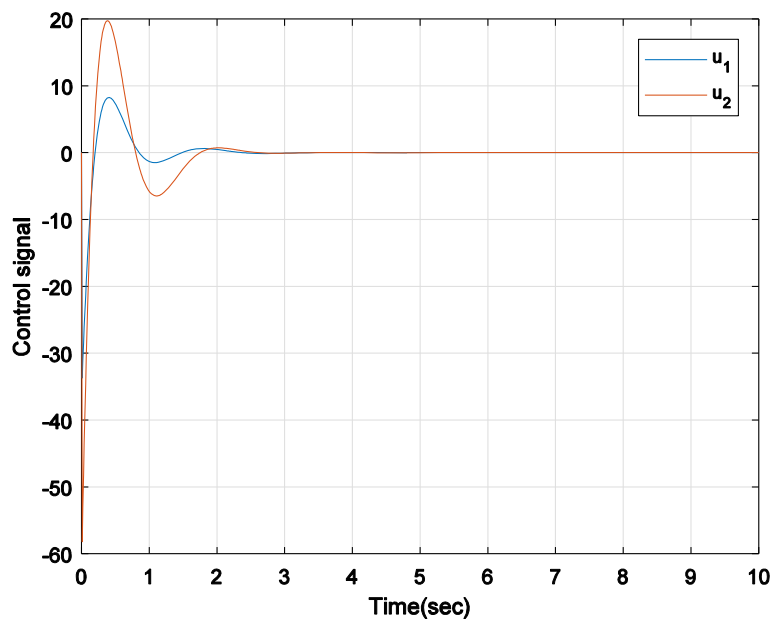


Fig. 17 Control inputs applied to the robot to reach the target point $(x_d, y_d) = (2, 5.5)$ and in presence obstacle points $(x_{01}, y_{01}) = (-1.5, 7.8)$ and $(x_{02}, y_{02}) = (0.1, 7)$ with LQR method

Abbreviations

LQR: Linear quadratic regulator; SDC: State-dependent coefficient; SDRE: State-dependent Riccati equation

Notation

R: Weighting matrix inputs; Q: Weighting matrix of states; W: Matrix final cost; J: Cost function; u: Control signal

Acknowledgements

Authors of this would like to express the sincere thanks to the National Natural Science Foundation of China for their funding support to carry out this project.

Funding

This work is supported by the National Natural Science Foundation of China (61772454, 6171171570).

Availability of data and materials

The data is available on request with prior concern to the first author of this paper.

Authors' contributions

All the authors conceived the idea, developed the method, and conducted the experiment. SMHR contributed to the formulation of methodology and experiments. AKS contributed in the data analysis and performance analysis. JW contributed to the overview of the proposed approach and decision analysis. H-jK contributed to the algorithm design and data sources. All authors read and approved the final manuscript.

Ethics approval and consent to participate

All the authors of this manuscript would like to declare that they mutually agreed no conflict of interest. All the results of this paper are with respect to the experiments on human subject.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

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

Author details

¹Department of Electrical and Computer Engineering, Shiraz University of Technology, Shiraz, Iran. ²School of Computing Science and Engineering, Vellore Institute of Technology (VIT), Vellore 632014, India. ³School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, China. ⁴Business Administration Research Institute, Sungshin Women's University, Seoul, Republic of Korea.

Received: 30 May 2018 Accepted: 13 August 2018

Published online: 29 August 2018

References

- G. Han, C. Zhang, J. Jiang, X. Yang, M. Guizani, Mobile anchor nodes path planning algorithms using network-density-based clustering in wireless sensor networks. *J. Netw. Comput. Appl.* **85**, 64–75 (2017)
- S. Hacoheh, S. Shoval, N. Shvalbc, Applying probability navigation function in dynamic uncertain environments. *Robot. Auton. Syst.* **87**, 237–246 (2017)
- S. Herzog, F. Wörgötter, T. Kulvicus, Generation of movements with boundary conditions based on optimal control theory. *Robot. Auton. Syst.* **94**, 1–11 (2017)
- R. Iraj, S.M. Ghadami, Aids control using state-dependent Riccati equation. *Sci. Int.* **27**, 1183–1188 (2015)
- Q. Jia, Y. Liu, G. Chen, H. Sun, State-dependent Riccati equation control for motion planning of space manipulator carrying a heavy payload. *Open Mech. Eng. J.* **9**(1), 992–999 (2015)
- K. Sinan, Tracking control of elastic joint parallel robots via state-dependent Riccati equation. *Turk. J. Elec. Eng. Comput. Sci.* **23**, 522–538 (2015)
- F. Sağlam, Y.S. Ünlüsoy, State dependent Riccati equation control of an active hydro-pneumatic suspension system. *J. Autom. Control Res.* **1**, 1–10 (2014)
- A. Khamis, D.S. Naidu, Finite-horizon optimal nonlinear regulation and tracking using differential state dependent Riccati equation. *Int. J. Control.* 1–18 (2014) <https://doi.org/10.1080/00207179.2014.977823>
- S.R. Nekoo, Nonlinear closed loop optimal control: a modified state-dependent Riccati equation. *ISA Trans.* **52**, 285–290 (2013)
- M.H. Korayem, S.R. Nekoo, State-dependent differential Riccati equation to track control of time-varying systems with state and control nonlinearities. *ISA Trans.* **57**, 117–135 (2015)
- Y.L. Kuo, T.L. Wu, A suboptimal tracking control approach based on the state-dependent Riccati equation method. *J. Comput. Theor. Nanosci.* **13**, 1013–1021 (2016)
- J. Slávka, S. Ján, in *IEEE 11th International Symposium on Intelligent Systems and Informatics*. Application of the state-dependent Riccati equation method in nonlinear control design for inverted pendulum systems (2013), pp. 26–28
- F. Liccardo, S. Strano, M. Terzo, in *World Congress on Engineering*. Optimal control using state-dependent Riccati equation (SDRE) for a hydraulic actuator (2013), pp. 3–5
- S.M.H. Rostami, M. Ghazaani, State dependent Riccati equation tracking control for a two link robot. *J. Comput. Theor. Nanosci.* **15**, 1490–1494 (2018)
- A. Hamdache, S. Saadi, A stochastic nominal control optimizing the adoptive immunotherapy for cancer using tumor-infiltrating lymphocytes. *Int. J. Dynam. Control.* **5**, 783–798 (2017)
- R. Padmanabhan, N. Meskin, W.M. Haddad, Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment. *Math. Biosci.* **293**, 11–20 (2017)
- A. Behboodi, S.M. Salehi, SDRE controller for motion design of cable-suspended robot with uncertainties and moving obstacles. *Mater. Sci. Eng.* **248**, 1–6 (2017)
- M.P.P. Reddy, J. Jacob, Vibration control of flexible link manipulator using SDRE controller and Kalman filtering. *Stud. Inform. Control* **2**, 143–150 (2017)
- X. Wang, E.E. Yazb, S.C. Schneider, Y.I. Yazb, H₂-H_∞ control of continuous-time nonlinear systems using the state-dependent Riccati equation approach. *Syst. Sci. Control Eng.* **5**(1), 224–231 (2017)
- Y. Chen, H. Peng, J. Grizzle, Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Trans. Control Syst. Technol.* **99**, 1–13 (2018)
- M.G. Plessen, D. Bernardini, H. Esen, A. Bemporad, Spatial-based predictive control and geometric corridor planning for adaptive cruise control coupled with obstacle avoidance. *IEEE Trans. Control Syst. Technol.* PP(99), 1–13 (2018)
- D. Han, H. Nie, J. Chen, M. Chen, Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection. *Robot. Comput. Integr. Manuf.* **49**, 98–104 (2018)
- M. Mendesy, A.P. Coimbray, M.M. Crisostomoy, Assis - cicerone robot with visual obstacle avoidance using a stack of odometric data. *IAENG Int. J. Comput. Sci.* **45**, 219–227 (2018)
- P.D.H. Nguyen, C.T. Recchiuto, A. Sgorbissa, Real-time path generation and obstacle avoidance for multirotors: a novel approach. *J. Intell. Robot. Syst.* **89**, 27–49 (2018)
- Ç. Tayfun, in *The International Federation of Automatic Control*. State-dependent Riccati equation (SDRE) control: a survey (2008), pp. 6–11
- P.K. Menon, T. Lam, L.S. Crawford, V.H.L. Cheng, in *American Control Conference*. Real-time computational methods for SDRE nonlinear control of missiles (2002), pp. 1–17
- M.H. Hurni, P. Sekhavat, I.M. Ross, in *AIAA Infotech Aerospace Conference AIAA*. Issues on robot optimal motion planning and obstacle avoidance (2009), pp. 1–20
- S.S. Mohammadi, H. Khaloozade, in *4th international conference on control, instrumentation, and automation*. Optimal motion planning of unmanned ground vehicle using SDRE controller in the presence of obstacles (2016), pp. 167–171
- S. Xie, P. Wu, Y. Peng, J. Luo, J. Gu, D. Qu, Q. Li, in *Proceeding of the IEEE International Conference on Information and Automation/Hailar*. The obstacle avoidance planning of USV based on improved artificial potential field (2014), pp. 746–751