Polynomial Optimal Trajectory Planning and Obstacle Aviodance for Omni-directional Mobile Robots in Dynamic Environments

Fatemeh Tohfeh [1], Ahmad Fakharian [2]*

Received: 2015/9/15 Accepted: 2015/11/22

Abstract

This paper presents a parameterization method to optimal trajectory planning and dynamic obstacle avoidance for Omni-directional robots. The aim of trajectory planning is minimizing a quadratic cost function while a maximum limitation on velocity and acceleration of robot is considered. First, we parameterize the trajectory using polynomial functions with unknown coefficients which transforms trajectory planning to an optimization problem. Then we use a novel method to solving the optimization problem and obtaining the unknown parameters. Finally, the efficiency of proposed approach is confirmed by simulation.

Keywords: obstacle avoidance; Omni-directional robot; optimization problem; polynomial trajectory; trajectory planning.

1. Introduction

Trajectory optimization is one of the most important issues in the research on mobile robots. A mobile robot must be able to move in his workspace from any initial location to any specific goal location while minimizing a performance index and avoiding collision with obstacles. The *trajectory optimization* and *optimal control problem* terms can be used interchangeably. For classical problems and some special weakly nonlinear low dimensional systems, the solution can be obtained analytically using the necessary and sufficient conditions of optimality. In [1] and [2], a new analytical solution and a reduced-order analytical solution to mobile robot trajectory generation in the presence of moving obstacles are proposed. For dynamic systems described by strongly nonlinear differential equations, numerical methods must be used to obtain solution of optimal control problem. A classification of various techniques for solving trajectory optimization problems numerically has been described in [3]. Numerical methods for solving optimal control problems are divided into two main groups: *indirect* and *direct* methods. A Survey of direct and indirect methods for trajectory optimization are presented in [4] and [5]. In an indirect method, the calculus of variation is used to determine the first-order optimality conditions. Indeed, the indirect approach solves the problem indirectly by converting the optimal control problem to a boundary-value problem [6]-[9]. In direct approaches the optimal control problem is transformed into a nonlinear programing problem (NLP). The advantage of the direct approach is that the user does not have to be concerned with adjoint variables or switching structures. One disadvantage of direct methods is that they produce less accurate solutions than indirect methods [4]. The approaches of converting an optimal control problem to a NLP are classified in three broad categories: *State parameterization methods, control parameterization methods and state and control parameterization methods.* There are several methods which utilize these three approaches to transcribe an optimal control problem to a NLP. Examples include *direct collocation methods* [10-12], *direct single and multiple shooting methods* [13-15].

In this paper we deal with nonlinear differential equations and constraints so employ a numerical method to trajectory optimization. To transformation of optimal control problem to NLP, use a direct state parameterization approach and present an optimal polynomial trajectory planning and obstacle avoidance for Omni-directional mobile robots. The trajectory must be able to move the robot from any specific initial position to any known desired position. An optimal performance index is set up to parameterized trajectory stays close to the shortest path and minimum energy. Maximum velocity and acceleration constraints of robot which exist in practical take into account. Combining with the obstacle avoidance, the practical optimal collusion-free trajectory can be generated. This trajectory is presented by a parameterized polynomial which is obtained from solving an optimization problem. This paper is along the same line of the work [17] with a different approach to solve optimization problem. This solution method is so faster than previous solution method.

The paper is organized as follow. Problem statement and model of the Omni-directional robot is presented in section II. In section III, with some assumption, formulation of the problem is stated completely. A polynomial trajectory planning and obstacle avoidance is proposed in Section IV. Trajectory planning and obstacle avoidance procedure is examined by simulations in section V and finally, the paper is concluded in section VI.

1 M.Sc. of Control Engineering, Department of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Email: f.tohfeh@qiau.ac.ir

2 Assistant Professor, Department of Electrical, Biomedical and Mechatronics Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Corresponding Author Email: ahmad.fakharian@qiau.ac.ir

2. Problem statement and robot model

A. Problem statement

Find a trajectory between two points A and B which with considering maximum limitation on velocity and acceleration of robot, avoids collusion with obstacles and minimizes a performance index as shown in Figure 1.

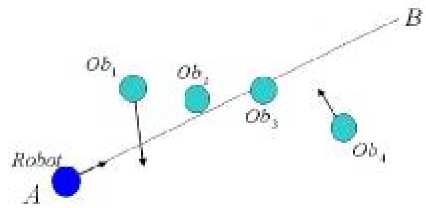

Figure 1. Mobile robot in dynamic environment with moving obstacles

B. Robot's model

In [17], we used a linear state space model as follows:

$$\dot{X}(t) = AX(t) + Bu(t), \qquad (1)$$

Where

$$X = \begin{bmatrix} x_1(\mathrm{t}) \\ x_2(\mathrm{t}) \\ x_3(\mathrm{t}) \\ x_4(\mathrm{t}) \end{bmatrix},\ U(\mathrm{t}) = \begin{bmatrix} u_1(\mathrm{t}) \\ u_2(\mathrm{t}) \end{bmatrix}, \qquad (2)$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad (3)$$

Which $x_1(\mathrm{t})$ and $x_2(\mathrm{t})$ are positions of robot in x and y Cartesian coordinates, also $x_3(\mathrm{t})$ and $x_4(\mathrm{t})$ are velocities of robot in x and y axis respectively. $u_1(\mathrm{t})$ and $u_2(\mathrm{t})$ are the control signals in x and y axis too.

3. Problem formulation and assumptions

To obtain optimal trajectory, we often need to solve an optimal control problem. The problem to be solved is:

Find the inputs to a dynamical system in $\mathrm{t} \in [\mathrm{t}_0, \mathrm{t}_1]$ which can make minimum the quadratic performance index while satisfying any constraints on the trajectory planning. It can be described into mathematical forms which are written as follows:

$$\begin{cases} \min J = \int_{t_0}^{t_1} \frac{1}{2} \begin{bmatrix} X^T(t) Q X(t) \\ + u^T(t) R u(t) \end{bmatrix} dt, \\ s.t: \dot{X}(t) = AX(t) + Bu(t), \\ X(t_0) = X_0, X(t_1) = X_1, \\ c_i(\mathrm{X(t)}, \dot{\mathrm{X}}(\mathrm{t}), \mathrm{U(t)}, \mathrm{t}) \le 0 \end{cases} \qquad (4)$$

Where $X(t)$ is state vector, $u(t)$ is input vector, $Q \in R^{n\times n}$ is a semi-definite matrix, $R \in R^{r\times r}$ is a definite matrix, X_0 is initial condition vector, X_1 is final condition vector and $c_i(\mathrm{X(t)}, \dot{\mathrm{X}}(\mathrm{t}), \mathrm{U(t)}, \mathrm{t}) \le 0$ are linear or nonlinear inequality constraints of states, inputs and their derivatives.

To trajectory planning in this paper, we have three set of nonlinear inequality constraints due to obstacle avoidance, maximum velocity constraint and maximum acceleration constraint.

Nonlinear inequality constraints due to obstacle avoidance are as follows:

$$W_j(\mathrm{X(t), t}) \ge 0,\ j = 1, 2, \dots, m \qquad (5)$$

Where m is number of obstacles which robot must avoid collusion with them.

Also Nonlinear inequality constraints due to maximum velocity and acceleration constraints are as follows:

$$(\mathrm{x}_3^2(t) + \mathrm{x}_4^2(t))^{\frac{1}{2}} \le v_{\mathrm{m}} \qquad (6)$$

$$(\dot{\mathrm{x}}_3^2(t) + \dot{\mathrm{x}}_4^2(t))^{\frac{1}{2}} \le a_{\mathrm{m}} \qquad (7)$$

Where v_{m} and a_{m} are maximum limitation on velocity and acceleration of robot respectively.

In next subsection we make the following assumptions on the trajectory planning.

- Final velocity of robot must be zero to avoid discontinuities in the solution when reaching close to the desired final state [16].
- All obstacles are considered as circular robots with radius r_j, center of $\left(x_{c_{1_j}} \quad x_{c_{2_j}}\right)^T$ and constant velocity $\left(v_{c_{1_j}} \quad v_{c_{2_j}}\right)^T$.
- Weight of states and energy in the performance index be equal. Then we consider Q and R as elementary matrixes with appropriate dimensions.

By these assumptions, mathematical form of optimal control problem associated with the trajectory planning can be rewritten as follows:

$$\begin{cases} \min J = \int_{t_0}^{t_f} \frac{1}{2}\begin{bmatrix} x_1^2(\mathrm{t}) + x_2^2(\mathrm{t}) + x_3^2(\mathrm{t}) \\ + x_4^2(\mathrm{t}) + u_1^2(\mathrm{t}) + u_2^2(\mathrm{t}) \end{bmatrix} dt, \\ s.t: \\ \begin{cases} \dot{x}_1(\mathrm{t}) = x_3(\mathrm{t}) = v_x(\mathrm{t}) \\ \dot{x}_2(\mathrm{t}) = x_4(\mathrm{t}) = v_y(\mathrm{t}) \\ \dot{x}_3(\mathrm{t}) = u_1(\mathrm{t}) = a_x(\mathrm{t}) \\ \dot{x}_4(\mathrm{t}) = u_2(\mathrm{t}) = a_y(\mathrm{t}) \end{cases} \\ X(t_0) = \begin{bmatrix} x_{0_1} & x_{0_2} & x_{0_3} & x_{0_4} \end{bmatrix}^T \\ X(t_1) = \begin{bmatrix} x_{1_1} & x_{1_2} & 0 & 0 \end{bmatrix}^T \\ \begin{pmatrix} (x_1(t) - (x_{c_{1_j}} + v_{c_{1_j}} t))^2 \\ + (x_2(t) - (x_{c_{2_j}} + v_{c_{2_j}} t))^2 \geq r_j{}^2 \end{pmatrix} \\ (\mathrm{x}_3^2(t) + \mathrm{x}_4^2(t))^{\frac{1}{2}} \leq v_{\mathrm{m}}, \\ (\dot{x}_3^2(t) + \dot{x}_4^2(t))^{\frac{1}{2}} \leq a_{\mathrm{m}} \end{cases} \tag{8}$$

Where $\begin{pmatrix} x_{c_{1_j}} & x_{c_{2_j}} \end{pmatrix}^T$ and $\begin{pmatrix} v_{c_{1_j}} & v_{c_{2_j}} \end{pmatrix}^T$ are initial position of obstacles and constant velocity of obstacles respectively.

4. Polynomial trajectory planning and obstacle avoidance

A. Polynomial trajectories

In this section firstly, polynomial trajectories are presented. Two polynomials are used for each of the Cartesian coordinates x and y as follows:

$$x_1(t) = x(\mathrm{t}) = a_0 + a_1 t + a_2 t^2 + ... + a_n t^n, \tag{9}$$

$$x_2(t) = \mathrm{y}(\mathrm{t}) = b_0 + b_1 t + b_2 t^2 + ... + b_n t^n \tag{10}$$

Where $a_0, a_1, ..., a_n$ and $b_0, b_1, ..., b_n$ are the unknown coefficients of the polynomials.

Then from (1)-(3) we get:

$$x_3(t) = \dot{x}(\mathrm{t}) = a_1 + 2a_2 t + ... + n a_n t^{(n-1)}, \tag{11}$$

$$x_4(t) = \dot{y}(\mathrm{t}) = b_1 + 2b_2 t + ... + n b_n t^{(n-1)}, \tag{12}$$

$$u_1(t) = 2a_2 + 6a_3\,\mathrm{t} + ... + n(n-1) a_n t^{(n-2)} \tag{13}$$

$$u_2(t) = 2b_2 + 6b_3\,\mathrm{t} + ... + n(n-1) b_n t^{(n-2)}, \tag{14}$$

In the next, we choice appropriate degree of polynomial trajectories such that all boundary conditions and constraints can be satisfied.

Since we have eight boundary conditions, we need eight coefficients to fulfill them. Moreover for minimization of performance index subject to the constraints, at least two coefficients are required. So we use a fourth-order polynomials for each trajectory. We get:

$$x_1(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4, \tag{15}$$

$$x_2(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4, \tag{16}$$

$$x_3(t) = \dot{x}(\mathrm{t}) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3, \tag{17}$$

$$x_4(t) = \dot{y}(\mathrm{t}) = b_1 + 2b_2 t + 3b_3 t^2 + 4b_4 t^3, \tag{18}$$

$$u_1(t) = \ddot{x}(\mathrm{t}) = 2a_2 + 6a_3\,\mathrm{t} + 12a_4\,\mathrm{t}^2, \tag{19}$$

$$u_2(t) = \ddot{y}(\mathrm{t}) = 2b_2 + 6b_3\,\mathrm{t} + 12b_4\,\mathrm{t}^2 \tag{20}$$

B. Transformation of the optimal control problem to parametric optimization problem

After substituting (15)-(20) in (8), the optimal control problem (8) converts to a parametric optimization problem with ten unknown coefficients. As mentioned, coefficients $a_0, a_1, a_2, a_3, \mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2$ and b_3 are obtained to satisfy boundary conditions. Then the problem can be rewritten as follows:

$$\begin{cases} \min J = \frac{1}{2}\int_{t_0}^{t_f} \begin{pmatrix} x_1^2(a_4, b_4, \mathrm{t}) \\ + x_2^2(a_4, b_4, \mathrm{t}) \\ + x_3^2(a_4, b_4, \mathrm{t}) \\ + x_4^2(a_4, b_4, \mathrm{t}) \\ + u_1^2(a_4, b_4, \mathrm{t}) \\ + u_2^2(a_4, b_4, \mathrm{t}) \end{pmatrix} \mathrm{dt}, \\ s.t: \\ [x_1(a_4, b_4, \mathrm{t}) - (x_{c_{1_j}} + v_{c_{1_j}} t)]^2 + \\ [x_2(a_4, b_4, \mathrm{t}) - (x_{c_{2_j}} + v_{c_{2_j}} t)]^2 \geq r_j{}^2, \\ [\mathrm{x}_3^2(a_4, b_4, \mathrm{t}) + \mathrm{x}_4^2(a_4, b_4, \mathrm{t})]^{\frac{1}{2}} \leq v_{\mathrm{m}}, \\ [\dot{x}_3^2(a_4, b_4, \mathrm{t}) + \dot{x}_4^2(a_4, b_4, \mathrm{t})]^{\frac{1}{2}} \leq a_{\mathrm{m}} \end{cases} \tag{21}$$

C. Solution method of parametric optimization problem

In [17], we used an approach for obtaining the optimization problem which solve the problem in whole of time (t_0, t_f). this make increase simulation time. In this paper we utilize a novel approach that decrease simulation time dramatically. In previous method we check all the inequality constraints for throughout of interval of time. Here we check these constraints only for once in all the time interval. Firstly, we convert the constraints to non-positive or non-negative constraints. Then determine minimum and maximum of them. For checking non-positive constraints, we compute maximum of the constraints on time and unknown coefficients are obtained such that the minimum be non-positive. Also For checking non-negative constraints, maximum of the constraints are computed on time and unknown coefficients are acquired such that the maximum be non-negative. Therefore, (21) can be modified as follows:

$$\begin{cases} \min J = \dfrac{1}{2}\displaystyle\int_{t_0}^{t_f} \begin{pmatrix} x_1^2(a_4,b_4,t) \\ +x_2^2(a_4,b_4,t) \\ +x_3^2(a_4,b_4,t) \\ +x_4^2(a_4,b_4,t) \\ +u_1^2(a_4,b_4,t) \\ +u_2^2(a_4,b_4,t) \end{pmatrix} dt, \\ s.t: \\ \min\Big\{[x_1(a_4,b_4,t)-(x_{c_{1_j}}+v_{c_{1_j}}t)]^2 + \\ [x_2(a_4,b_4,t)-(x_{c_{1_j}}+v_{c_{2_j}}t)]^2 - {r_j}^2\Big\} \geq 0, \\ \max\begin{pmatrix} [x_3^2(a_4,b_4,t) \\ +x_4^2(a_4,b_4,t)]^{\frac{1}{2}} - v_m \end{pmatrix} \leq 0, \\ \max\begin{pmatrix} [\dot{x}_3^2(a_4,b_4,t) \\ +\dot{x}_4^2(a_4,b_4,t)]^{\frac{1}{2}} - a_m \end{pmatrix} \leq 0 \end{cases} \quad (22)$$

5. Simulation results

To illustrate the efficiency of the proposed method, we have performed Matlab simulations to generate optimal collision-free trajectories. Also to better survey, we compare simulation results in present method and previous method was presented in [17].

A. Scenario 1

A representative simulation is discussed, where the following initial and final conditions is used:

$$x_{0_1} = x_{0_2} = 0\,\text{m},\ x_{1_1} = 2\,\text{m}, x_{1_2} = 1\,\text{m}, \quad x_{0_3} = x_{0_4} = 0\,\text{m/s} \quad (23)$$

Maximum velocity and acceleration of robot are equal to:

$$v_m = 2\,\text{m/s},\ a_m = 3\,\text{m/s}^2, \quad (24)$$

Also, time interval is considered $t \in [0 \quad 4]$. Table 1 shows simulation data for optimal trajectory planning and multi obstacle avoidance.

Obtained polynomial functions are as follows:

$$x_1(t) = -0.1546t^2 + 0.2023t^3 - 0.0331t^4, \quad (25)$$

$$x_2(t) = 0.2195t^2 - 0.0473t^3 + 0.002t^4, \quad (26)$$

$$x_3(t) = 0.3092t + 0.6069t^2 - 0.1324t^3, \quad (27)$$

$$x_4(t) = 0.439t - 0.1417t^2 + 0.008t^3, \quad (28)$$

$$u_1(t) = -0.3972t^2 + 1.2138t - 0.3092, \quad (29)$$

$$u_2(t) = \ddot{x}(t) = 0.024t^2 - 0.2835t + 0.439 \quad (30)$$

Figure 2 depicts diagrams of position, velocity and acceleration of robot. The curve of optimal trajectory and multi moving obstacle avoidance is shown in Figure 3. Total velocity and acceleration of robot are illustrated in Figure 4.

Table 1. Multi obstacle avoidance simulation data (Scenario 1)

Positions and Velocities of obstacles	Position (m)		Velocity (m/s)		Radius (m)
	x_{c_1}	x_{c_2}	v_{c_1}	v_{c_2}	
Obstacle 1	1	1.3	0.18	–0.19	0.16
Obstacle 2	0.75	1	0.1	–0.25	0.18
Obstacle 3	0.4	0.8	0.2	–0.4	0.12

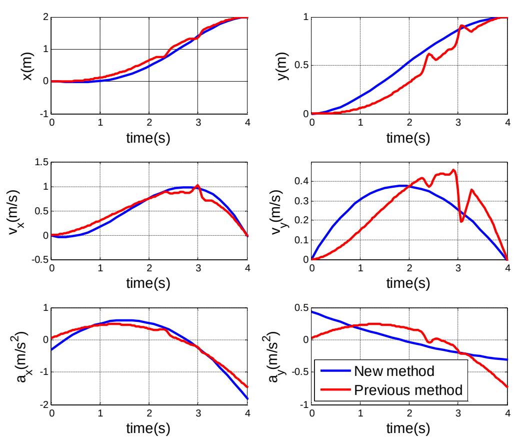

Figure 2. Diagrams of position, velocity and acceleration of robot for optimal trajectory planning and multi moving obstacle avoidance (scenario 1)

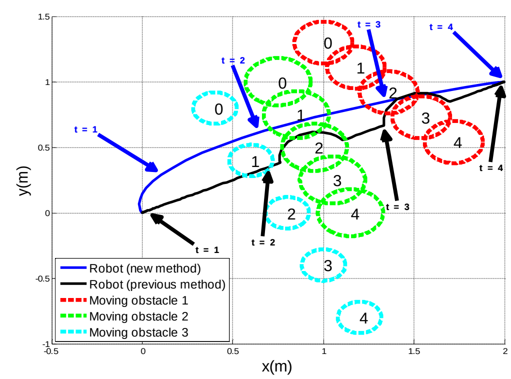

Figure 3. Optimal trajectory and multi moving obstacle avoidance (scenario 1)

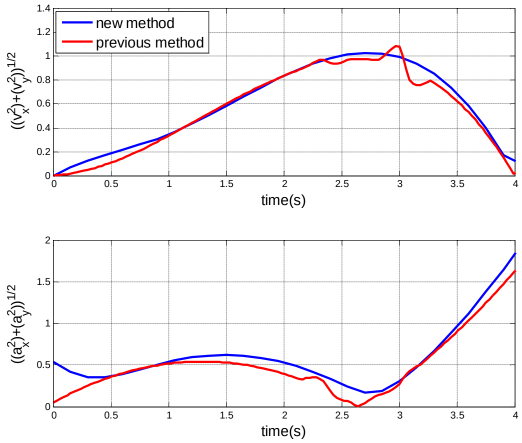

Figure 4. Total velocity and acceleration of robot (scenario 1)

As can be seen in Figure 3, the robot avoids collision with multiple moving obstacles by two method effectively.

As shown in Figure 4, limitations on maximum velocity and acceleration of robot are satisfied.

Run time of simulation for new method is equal to 0.1s and this time for previous method was obtained 11.63s. Furthermore, the performance index value for new method and previous method are equal to 4.69 and 4.48 respectively.

B. Scenario 2

In this scenario, we consider initial and final conditions as follows:

$$x_{0_1} = x_{0_2} = 0\,\text{m},\ x_{1_1} = x_{1_2} = 3\,\text{m},\quad x_{0_3} = x_{0_4} = 0\,\text{m/s} \tag{31}$$

Maximum velocity and acceleration of robot are equal to:

$$v_{\text{m}} = 2\,\text{m/s},\ a_{\text{m}} = 3\,\text{m/s}^2, \tag{32}$$

Time interval is considered $t \in [0 \quad 5]$. Simulation data for trajectory planning and multi obstacle avoidance are given in Table 2.

Table 2. Multi obstacle avoidance simulation data (Scenario 2)

Positions and Velocities of obstacles	Position (m)		Velocity (m/s)		Radius (m)
	x_{c_1}	x_{c_2}	v_{c_1}	v_{c_2}	
Obstacle 1	1.5	−0.9	0	0.7	0.22
Obstacle 2	0.2	2	0.3	−0.5	0.25
Obstacle 3	2.5	0.5	−0.3	0.6	0.15
Obstacle 4	3	3	−0.3	−0.35	0.2

Obtained polynomial functions are as follows:

$$x_1(t) = 0.4825t^2 - 0.097t^3 + 0.0049t^4, \tag{33}$$

$$x_2(t) = 0.13t^2 + 0.044t^3 - 0.0092t^4, \tag{34}$$

$$x_3(t) = 0.965t - 0.291t^2 + 0.0196t^3, \tag{35}$$

$$x_4(t) = 0.26t + 0.132t^2 - 0.0368t^3, \tag{36}$$

$$u_1(t) = 0.0588t^2 - 0.582t + 0.965, \tag{37}$$

$$u_2(t) = -0.1104t^2 + 0.264t + 0.26 \tag{38}$$

Figure 5 shows diagrams of position, velocity and acceleration of robot. The curves of optimal trajectory and total velocity and acceleration of robot are illustrated in Figure 6 and Figure 7.

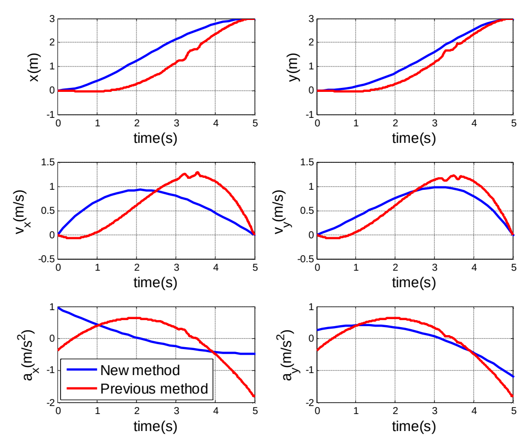

Figure 5. Diagrams of position, velocity and acceleration of robot for optimal trajectory planning and multi moving obstacle avoidance (scenario 2)

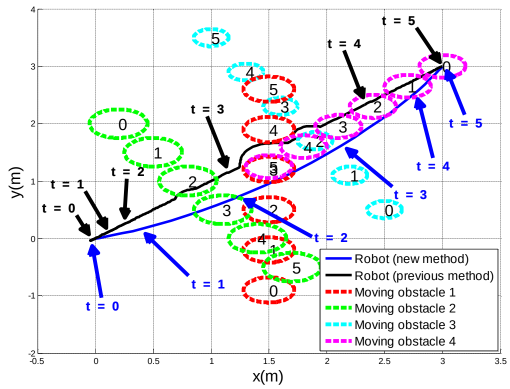

Figure 6. Optimal trajectory and multi moving obstacle avoidance (scenario 2)

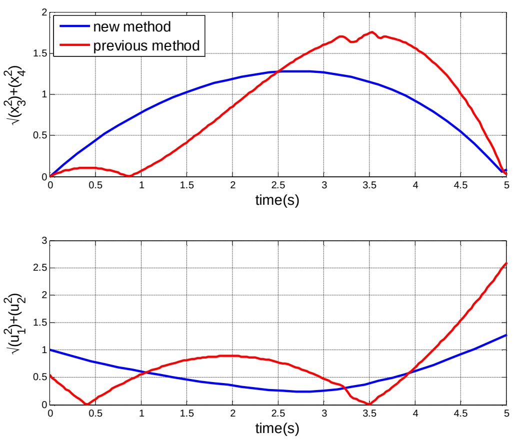

Figure 7. Total velocity and acceleration of robot (scenario 2)

As shown in Figure 6 and Figure 7, the robot avoids collision with multiple moving obstacles by two method effectively while limitations on maximum velocity and acceleration of robot are satisfied.

Simulation time for new method is 0.05s, whilst Simulation time for previous method was obtained 9.35s. Also the value of cost functions for new method and previous method are 19.45 and 16.3 respectively. On the other side, control effort for new method is lower than previous method.

7. Conclusion

In this paper was presented a procedure for optimal trajectory planning and obstacle avoidance of Omni-directional mobile robots in dynamic environments using polynomials of fourth degree. The obtained trajectory minimizes a quadratic performance index while satisfying velocity and acceleration constraints. First, by choice of appropriate objective function the problem was formulated as an optimal control problem. After that, by parameterization of trajectories, the problem was converted to a nonlinear programing problem. Then, to solve the problem a new method has been employed. Finally, a simulation has been performed and effectiveness of proposed method was illustrated by simulation results. The most significant advantages of this method are that it reduces implementation time markedly and is much low cost computationally.

References

[1] Qu, Zhihua, Jing Wang, and Clinton E. Plaisted. “A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles.” *Robotics, IEEE Transaction on 20.6 (2004)*: 978-993.

[2] Wang, Jing, et al. “A reduced-order analytical solution to mobile robot trajectory generation in the presence of moving obstacles. “*Robotics and Automation, 2004. Proceedings. ICRA04. 2004 IEEE International Conference on. Vol. 5. IEEE, 2004.*

[3] Betts, John T. “Survey of numerical methods for trajectory optimization.” *Journal of guidance, control, and dynamics* 21.2 (1998): 193-207.

[4] Von Stryk, Oskar, and Roland Bulirsch. “Direct and indirect methods for trajectory optimization.” *Annals of operations research 37.1* (1992): 357-373.

[5] Rao, Anil V. “A survey of numerical methods for optimal control.” *Advances in the Astronautial Sciences* 135.1 (2009): 497-528.

[6] Hartl, Richard F., Suresh P. Sethi, and Raymond G. Vickson. “A survey of the maximum principles for optimal control problems with state constraints.” *SIAM review* 37.2 (1995): 181-218.

[7] Ross, I. Michael. “Trajectory optimization by indirect scpectral collocation methods.” (2000).

[8] Pepy, Romain, and Bruno Hérissé. “An Indirect Methods for Optimal Guidance of a Glidar.” *19th IFAC World Congress.* 2014.

[9] Barron, Roger L., and Cleon M. Chick. “Improved indirect method for air_vehicle trajectory optimization.” *Journal of guidance, control, and dynamics* 29.3 (2006): 643-652.

[10] Von Stryk, Dipl Math Oskar. Numerical solution of optimal control problems by direct collocation. Birkhäuser Basel, 1993.

[11] Benson, David A., et al. “Direct trajectory optimization and costate estimation via an orthogonal collocation method.” *journal of Guidance, Control, and Dynamics* 29.6 (2006): 1435-1440.

[12] Tohidi, Emran, and Hassan Saberi Nik. “A Bessel collocation method for solving fractional optimal control problems.” *Applied Mathematical Modelling* 39.2 (2015): 455-465.

[13] Kirches, Christian. “The Direct Multiple Shooting Method for Optimal control.” *Fast Numerical Methods for Mixed-integer Nonlinear Model-Predictive control* (2011): 13-29.

[14] Bock, H. G., Diehl, M. M., Leineweber, D. B., & Schloder, J. P. “A Direct Multiple Shooting

Method for Real-time Optimization." *Nonlinear Model Predictive Control* 26 (2012): 245.

[15] Assassa, Fady, and Wolfgana Marquardt. "Dynamic optimization using adaptive direct multiple shooting." *Computers & Chemical Engineering* 60 (2014): 242-259.

[16] O. Purwin and R. D'Andrea, "Trajectory generation for four wheeled omnidirectional vehicles," in Proceedings of the American Control Conference, vol. 7, pp. 4979-4984, June 2005.

[17] Tohfeh, Fatemeh, and Ahmad fakharian. "Polynomial based optimal trajectory planning and obstacle avoidance for an omni-directional robot." *AI and Robotics, 20155th Iranian Conference on, (accepted)* IEEE, 2015.