Polynomial Optimal Trajectory Planning and Obstacle Avoidance 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 its 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 and 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](image)

   B. Robot’s model
   In [17], we used a linear state space model as follows:
   \[
   \dot{X}(t) = AX(t) + Bu(t),
   \]
   \[
   \text{Where } X = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}, \quad U(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix},
   \]
   \[
   A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},
   \]
   Which \( x_1(t) \) and \( x_2(t) \) are positions of robot in \( x \) and \( y \) Cartesian coordinates, also \( x_3(t) \) and \( x_4(t) \) are velocities of robot in \( x \) and \( y \) axis respectively. \( u_1(t) \) and \( u_2(t) \) 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 \( t \in [t_0, t_f] \) 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:
   \[
   \min J = \int_{t_0}^{t_f} \left[ \frac{1}{2} X^T(t)QX(t) + \frac{1}{2} u^T(t)Ru(t) \right] dt,
   \]
   \[
   \text{s.t. } \dot{X}(t) = AX(t) + Bu(t),
   \]
   \[
   X(t_0) = X_0, X(t_f) = X_1,
   \]
   \[
   c_j(X(t), \dot{X}(t), U(t), t) \leq 0
   \]
   Where \( X(t) \) is state vector, \( u(t) \) is input vector, \( Q \in \mathbb{R}^{m \times m} \) is a semi-definite matrix, \( R \in \mathbb{R}^{p \times p} \) is a definite matrix, \( X_0 \) is initial condition vector, \( X_1 \) is final condition vector and \( c_j(X(t), \dot{X}(t), U(t), t) \leq 0 \) 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:
   \[
   W_j(X(t), t) \geq 0, \quad j = 1, 2, ..., m
   \]
   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:
   \[
   (x_3^2(t) + x_4^2(t)) \leq v_m^2
   \]
   \[
   (\dot{x}_3^2(t) + \dot{x}_4^2(t)) \leq a_m^2
   \]
   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 \( x_{c1j}, x_{c2j} \) and constant velocity \( v_{xj} \).
   - 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:
\[
\begin{align*}
\min J = \int_0^T \left[ \frac{x_1^2(t) + x_2^2(t) + x_3^2(t)}{2} + x_4^2(t) + u_1^2(t) + u_2^2(t) \right] dt,
\end{align*}
\]
subject to:
\[
\begin{align*}
\mathbf{x}(t_0) &= \begin{bmatrix} x_{10} & x_{20} & x_{30} & x_{04} \end{bmatrix}^T, \\
\mathbf{x}(t_f) &= \begin{bmatrix} x_{1f} & x_{2f} & 0 & 0 \end{bmatrix}^T, \\
\left(x_1(t) - (x_{1i} + v_{ci} t)\right)^2 + \left(x_2(t) - (x_{2i} + v_{ci} t)\right)^2 &\geq r_f^2, \\
\frac{1}{\left(\Delta x_1^2 + x_2^2(t)\right)^2} &\leq v_m, \\
\frac{1}{\left(\Delta x_1^2 + x_2^2(t)\right)^2} &\leq a_m.
\end{align*}
\]
Where \((x_{1i}, x_{2i})^T\) and \((v_{ci}, v_{ci})^T\) 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:
\[
\begin{align*}
x_1(t) &= x(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n, \\
x_2(t) &= y(t) = b_0 + b_1 t + b_2 t^2 + \ldots + b_n t^n,
\end{align*}
\]
where \(a_0, a_1, \ldots, a_n\) and \(b_0, b_1, \ldots, b_n\) are the unknown coefficients of the polynomials.

Then from (1)-(3) we get:
\[
\begin{align*}
x_3(t) &= \mathbf{\dot{x}}(t) = a_1 + 2a_2 t + \ldots + n a_n t^{n-1}, \\
x_4(t) &= \mathbf{\ddot{x}}(t) = b_1 + 2b_2 t + \ldots + n(n-1) b_n t^{n-2}, \\
u_1(t) &= 2a_3 + 6a_4 t + \ldots + n(n-1)(n-2) a_n t^{n-3}, \\
u_2(t) &= 2b_3 + 6b_4 t + \ldots + n(n-1)(n-2) b_n t^{n-3}.
\end{align*}
\]

In the next, we choose 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:
\[
\begin{align*}
x_1(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4, \\
x_2(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4,
\end{align*}
\]
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 \(a_0, a_1, a_2, a_3, b_0, b_1, b_2\) and \(b_4\) are obtained to satisfy boundary conditions. Then the problem can be rewritten as follows:
\[
\begin{align*}
\min J = \int_0^T \left[ \frac{x_1^2(a_4, b_4, t) + x_2^2(a_4, b_4, t)}{2} + x_3^2(a_4, b_4, t) + x_4^2(a_4, b_4, t) + u_1^2(a_4, b_4, t) + u_2^2(a_4, b_4, t) \right] dt,
\end{align*}
\]
subject to:
\[
\begin{align*}
[x_1(a_4, b_4, t) - (x_{1i} + v_{ci} t)]^2 + [x_2(a_4, b_4, t) - (x_{2i} + v_{ci} t)]^2 &\geq r_f^2, \\
\frac{1}{[\Delta x_1^2 + x_2^2(a_4, b_4, t)]^2} &\leq v_m, \\
\frac{1}{[\Delta x_1^2 + x_2^2(a_4, b_4, t)]^2} &\leq a_m.
\end{align*}
\]

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 intervals \((t_0, t_f)\). 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:

\[
\min J = \frac{1}{2} \int_0^T dt \left[ x_1^2(a_4,b_4,t) + x_2^2(a_4,b_4,t) + x_3^2(a_4,b_4,t) + x_4^2(a_4,b_4,t) + u_1^2(a_4,b_4,t) + u_2^2(a_4,b_4,t) \right]
\]

\[
s.t: \quad \begin{cases} 
\min \left\{ \left[ x_1(a_4,b_4,t) - (x_{c_{ij}} + v_{c_{ij}} t) \right]^2 + \left[ x_2(a_4,b_4,t) - (x_{c_{ij}} + v_{c_{ij}} t) \right]^2 - r_j^2 \right\} \geq 0, \\
\max \left\{ \frac{1}{2} \left[ x_3^2(a_4,b_4,t) + x_4^2(a_4,b_4,t) \right] - v_m \right\} \leq 0, \\
\max \left\{ \frac{1}{2} \left[ \dot{x}_1(a_4,b_4,t) + \dot{x}_2(a_4,b_4,t) \right] - a_m \right\} \leq 0
\end{cases}
\]

**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:
\[
\begin{align*}
  x_0 &= x_{0_1} = 0 \text{m}, \quad x_1 = 2 \text{m}, \quad x_2 = 1 \text{m}, \\
  v_0 &= x_{0_1} = 0 \text{m/s}.
\end{align*}
\]

Maximum velocity and acceleration of robot are equal to:
\[
\begin{align*}
  v_m &= 2 \text{m/s}, \\
  a_m &= 3 \text{m/s}^2.
\end{align*}
\]

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

Obtained polynomial functions are as follows:
\[
\begin{align*}
  x_1(t) &= -0.1546r^2 + 0.2023r^3 - 0.0331t^4, \\
  x_2(t) &= 0.2195t^2 - 0.0473r^3 + 0.0024t^4, \\
  x_3(t) &= 0.3092r + 0.6069r^2 - 0.1324r^3, \\
  x_4(t) &= 0.439t - 0.1417t^2 + 0.008t^3, \\
  u_1(t) &= -0.3972t^2 + 1.2138t - 0.3092, \\
  u_2(t) &= \dot{\text{\#\#\#}} = 0.024r^2 - 0.2835t + 0.439
\end{align*}
\]

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.

\[
\text{Table 1. Multi obstacle avoidance simulation data (Scenario 1)}
\]

<table>
<thead>
<tr>
<th>Positions and Velocities of obstacles</th>
<th>Position (m) $x_1$</th>
<th>$x_2$</th>
<th>Velocity (m/s) $v_1$</th>
<th>$v_2$</th>
<th>Radius (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obstacle 1</td>
<td>1</td>
<td>1.3</td>
<td>0.18</td>
<td>-0.19</td>
<td>0.16</td>
</tr>
<tr>
<td>Obstacle 2</td>
<td>0.75</td>
<td>1</td>
<td>0.1</td>
<td>-0.25</td>
<td>0.18</td>
</tr>
<tr>
<td>Obstacle 3</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>-0.4</td>
<td>0.12</td>
</tr>
</tbody>
</table>
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:

\[
\begin{align*}
    x_0 &= x_2 = 0 \text{ m}, & x_1 &= x_3 = 3 \text{ m}, \\
    x_0 &= x_4 = 0 \text{ m/s}.
\end{align*}
\]

Maximum velocity and acceleration of robot are equal to:

\[
v_m = 2 \text{ m/s}, \quad a_m = 3 \text{ m/s}^2.
\]

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)

<table>
<thead>
<tr>
<th>Positions and Velocities of obstacles</th>
<th>Position (m)</th>
<th>Velocity (m/s)</th>
<th>Radius (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{c1} )</td>
<td>( x_{c2} )</td>
<td>( v_{c1} )</td>
<td>( v_{c2} )</td>
</tr>
<tr>
<td>Obstacle 1</td>
<td>1.5</td>
<td>-0.9</td>
<td>0</td>
</tr>
<tr>
<td>Obstacle 2</td>
<td>0.2</td>
<td>2</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Obtained polynomial functions are as follows:

\[
x_1(t) = 0.4825t^2 - 0.097t^3 + 0.0049t^4, \quad (33)
\]

\[
x_2(t) = 0.13t^2 + 0.044t^3 - 0.092t^4, \quad (34)
\]

\[
x_3(t) = 0.965t - 0.29t^2 + 0.0196t^3, \quad (35)
\]

\[
x_4(t) = 0.26t + 0.132t^2 - 0.0368t^3, \quad (36)
\]

\[
u_1(t) = 0.0588t^2 - 0.582t + 0.965, \quad (37)
\]

\[
u_2(t) = -0.1104t^2 + 0.264t + 0.26 \quad (38)
\]

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

Figure 5. Diagrams of position, velocity and acceleration of robot for optimal trajectory planning and multi moving obstacle avoidance (scenario 2)
illustrated by simulation results. The most significant performed and effectiveness of proposed method was been employed. Finally, a simulation has been problem. Then, to solve the problem a new method has been converted to a nonlinear programing problem. First, by choice of appropriate objective function the problem was formulated as an optimal control problem. Using polynomials of fourth degree. The obtained directional mobile robots in dynamic environments trajectory planning and obstacle avoidance of Omni.

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

