2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)
27-28 January 2016, Qazvin Islamic Azad University, Qazvin, Iran

Optimal Trajectory planning for Omni-directional Mobile Robots using Direct Solution of Optimal Control Problem

Naser Azim Mohseni
Department of Electrical, Biomedical, and Mechatronic Engineering
Qazvin Branch, Islamic Azad University
Qazvin, Iran
N.azimmohseni@qiau.ac.ir

Ahmad Fakharian
Department of Electrical, Biomedical, and Mechatronic Engineering
Qazvin Branch, Islamic Azad University
Qazvin, Iran
ahmad.fakharian@qiau.ac.ir

Abstract— **In this paper we address the problem of finding trajectory for Omni-directional mobile robots. The objective of the trajectory planning is moving the robot from its initial position to a final position in the presence of static obstacles while minimizing a quadratic index of performance. Along the trajectory, the robot requires to observe certain velocity and acceleration limitations. This problem can be formulated as a constraint nonlinear optimal control problem. To solve this problem, we employ direct method of numerical solution in which the trajectories are parameterized by parametric polynomial functions. By this transforming, the main optimal control problem converts to a nonlinear programming problem (NLP) by lower computational cost. To solve the NLP and obtaining the trajectories, we utilize a new approach with too small run time. The performance and effectiveness of the proposed method are tested in simulations.**

Keywords— omni-directional robot; parametric polynomials; static obstacles; trajectory planning; velocity and acceleration limitation

I. Introduction

Omni-Directional mobile robots provide a wide range of applications due to their ability to maneuver in different environments. One of the main problem in the research on mobile robots is finding a trajectory in order to robot motion. The trajectory planning problem can be formulated as an open loop optimal control problem. In general, the optimal control problems are nonlinear and finding an exact analytical solution of this problem is impossible. But there are the various techniques for obtaining an approximate analytical solution or numerical solution. The numerical solution can be classified into two major categories: indirect methods and direct methods [1-3]. Indirect methods are based on the calculus of variations and the maximum principle [4]. Direct methods operate by parameterization of state variables, control variables, and both of them. In the case where only the state variables are parameterized, the method is called a *state parameterization method,* when the control variables are parameterized, the method is called a *control parameterization method,* and when the state and control variables are parameterized, the method is called a *state and control parameterization method.* By this transformation, the optimal control problem is converted to a *nonlinear optimization problem* or *nonlinear programming problem* (NLP) [1]. Then the NLP can be solved using well known optimization techniques. One of the most basic direct method for solving optimal control problem is the *direct shooting method* which the control variables are parameterized using a specified functional form. A modified method is called *direct multiple shooting method* [5-7] which has been developed in order to overcome the numerical difficulties of the single shooting method. In this method, the time interval is divided into $M+1$ subintervals. Then the single shooting method is applied over each subinterval. Ref [8] proposed direct multiple shooting based methods with long prediction horizons for nonlinear model predictive control. *Direct collocation method* [9-11] is another direct method which operate by parameterization of both the state and control variables. Two most common forms of direct collocation methods are *local collocation* and *global collocation.* A local collocation method based on Legendre-Gauss-Radau is developed in [12]. Ref [13] presented a method for direct trajectory optimization and co-state estimation for general finite–horizon and infinite-horizon optimal control problems using global collocation at Legendre-Gauss-Radau points. An overview of direct collocation methods have been presented in [14].

In this paper we present an optimal trajectory planning. The aim of the trajectory generation is moving robot from its initial position to a terminal position subject to minimizing a quadratic cost function. Throughout the trajectory, robot should satisfy the maximum velocity and acceleration limitations.

Due to existence of nonlinear inequality constraints and differential constraints on states and controls in our problem formulation, indirect method is resulted to appear some nonlinear functions such as *Heaviside function*. These functions make complicate the calculations and increase the computational cost. Thus in this paper we use the direct method to solve the optimal control problem and finding desired trajectory. A state parameterization is utilized to convert nonlinear optimal control problem to a NLP. In [15], we presented a solution method which solves the NLP in arbitrary points of time interval $t \in (0, t_f)$. The method determines solution in all arbitrary points of the time interval and desired trajectories are obtained by putting together these solutions. In the current paper, we solve the NLP in a novel method which the solution is obtained at once.

The paper is organized as follows. In section II, we present dynamic model of Omni-directional mobile robot. In section III, by description of problem and some assumptions, the problem is formulated completely. Parameterization of trajectories, choice of polynomials degree, and new solution approach are presented in Section IV. Simulations are given in Section V. Finally, the paper ends with some concluding remarks in section VI.

II. The Omni-Directional Mobile Robot Model

This section presents the model of Omni-directional mobile robot. Let the states vector of the robot model be $X = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$, which $x, y, \dot{x},$ and $\dot{y}$, are the location of the center of mass (CM) of robot and velocity of robot in x and y direction, respectively. Also let consider the inputs vector of the system be $U = \begin{bmatrix} u_x & u_y \end{bmatrix}^T$, which u_x and u_y are the control efforts in x and y direction, respectively. By denoting $x_1(\mathrm{t}) \triangleq x$, $x_2(\mathrm{t}) \triangleq y$, $x_3(\mathrm{t}) = \dot{x}$, $x_4(\mathrm{t}) = \dot{y}$, $u_1(\mathrm{t}) \triangleq u_x(\mathrm{t})$, and $u_2(\mathrm{t}) \triangleq u_y(\mathrm{t})$, the state space equations of system can be written as follows:

$$\begin{cases} \dot{x}_1(\mathrm{t}) = x_3(\mathrm{t}) = v_x(\mathrm{t}) \\ \dot{x}_2(\mathrm{t}) = x_4(\mathrm{t}) = v_y(\mathrm{t}) \\ \dot{x}_3(\mathrm{t}) = u_1(\mathrm{t}) = a_x(\mathrm{t}) \\ \dot{x}_4(\mathrm{t}) = u_2(\mathrm{t}) = a_y(\mathrm{t}) \end{cases} \tag{1}$$

III. Problem Statement and Formulation

A. Problem Statement

We define the trajectory planning problem as follows:

Moving the Omni-directional robot from known initial location to known final location with minimizing a quadratic cost function of states and energy. There are static obstacles in the environment which the robot should avoid from collision with these obstacles. Also the velocity and acceleration of robot are limited.

B. Problem Formulation

The trajectory planning problem can be formulated as an optimal control problem as follows:

Find the control $U^*(\mathrm{t})$ in $t_0 \le t \le t_f$ which minimizes the cost function J :

$$J = \frac{1}{2}\int_{t_0}^{t_f} (\mathrm{x}(t)^T Qx(t) + u(t)^T Ru(t))\,\mathrm{dt} \tag{2}$$

Subject to robot dynamics (1),

Initial and final conditions:

$$\begin{aligned} &x_1(t_0) = x_{1_0},\ x_2(t_0) = x_{2_0}, \\ &x_3(t_0) = x_{3_0},\ x_4(t_0) = x_{4_0}, \end{aligned} \tag{3}$$

$$\begin{aligned} &x_1(t_f) = x_{1_f},\ x_2(t_f) = x_{2_f}, \\ &x_3(t_f) = x_{3_f},\ x_4(t_f) = x_{4_f}, \end{aligned} \tag{4}$$

State inequality constraints due to the static obstacles:

$$\bigcup_{i=1}^{k} \left[S_i(\mathrm{X(t), t}) \ge 0 \right] \tag{5}$$

State and control inequality constraints due to Maximum limitation on velocity and acceleration of the robot:

$$\sqrt{x_3^2(\mathrm{t}) + x_4^2(\mathrm{t})} \le v_{\max}, \tag{6}$$

$$\sqrt{u_1^2(\mathrm{t}) + u_2^2(\mathrm{t})} \le a_{\max} \tag{7}$$

Which Q is a symmetric positive semi-definite matrix and R is a symmetric positive definite matrix; x_{j_0} and x_{j_f}, $(j = 1, \ldots, 4)$ are the initial and final positions and velocities of robot; $S_i(\mathrm{X(t), t})$ represents the time-varying boundaries of the static obstacles, $v_{\max}$ and $a_{\max}$ are the maximum velocity and acceleration of robot, respectively.

We make the following assumptions without loss of any generality:

Assumption 1: The initial and final velocity of robot should be zero.

Assumption 2: The initial time and position of robot are considered at origin.

Assumption 3: Obstacles are presented by circles with radius r_i which are centered at $X_{c_i} = \begin{bmatrix} x_{c_i} & y_{c_i} \end{bmatrix}^T$.

Assumption 4: The weight of all states and inputs in the cost function are considered to be equal.

By these assumptions, the problem formulation can be rewritten as follows:

Find the control $U^*(\mathrm{t})$ in $0 \le t \le t_f$ which minimizes the cost function J :

$$J = \frac{1}{2}\int_0^{t_f} (\mathrm{x}_1^2 + \mathrm{x}_2^2 + \mathrm{x}_3^2 + \mathrm{x}_4^2 + u_1^2 + u_2^2)dt, \tag{8}$$

Subject to robot dynamics (1),

Initial and final conditions:

$$\begin{aligned} &x_1(0) = 0,\ x_2(0) = 0, \\ &x_3(0) = 0,\ x_4(0) = 0, \end{aligned} \tag{9}$$

$$x_1(t_f) = x_{1_f},\ x_2(t_f) = x_{2_f},$$
$$x_3(t_f) = 0,\ x_4(t_f) = 0, \tag{10}$$

State inequality constraints due to the circular static obstacles:

$$\bigcup_{i=1}^{k} \left[(x_1(t) - x_{c_i})^2 + (x_2(t) - y_{c_i})^2 \geq r_i^2 \right] \tag{11}$$

State and control inequality constraints (6) and (7).

Which $X_{c_i} = \begin{bmatrix} x_{c_i} \\ y_{c_i} \end{bmatrix}$ is the center and r_i is the radius of i^{th} circular obstacle.

IV. Optimal Trajectory Generation

A. Parameterization of the Trajectories

In this subsection, we present a parametric trajectory generation method which uses n-order polynomial functions of time to represent the desired trajectory between two points $X_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ and $X_f = \begin{bmatrix} x_f \\ y_f \end{bmatrix}$ respect to problem constraints.

The trajectories is described as follows:

$$x_1(t) = \sum_{m=0}^{n} a_m t^m,\ n = 1,2,3,... \tag{12}$$

$$x_2(t) = \sum_{p=0}^{n} b_p t^p,\ n = 1,2,3,... \tag{13}$$

$$x_3(t) = \sum_{m=1}^{n} m a_m t^{m-1}, \tag{14}$$

$$x_4(t) = \sum_{p=1}^{n} p b_p t^{p-1}, \tag{15}$$

$$u_1(t) = \sum_{m=2}^{n} m(m-1) a_m t^{m-2}, \tag{16}$$

$$u_2(t) = \sum_{p=2}^{n} p(p-1) b_p t^{p-2} \tag{17}$$

First, by considering boundary conditions, unknown parameters $a_0, a_1, ..., a_{n-1}$ and $b_0, b_1, ..., b_{n-1}$ are determined and we get the trajectories as functions of unknown parameters a_n and b_n as follows:

$$x_1(t) = f_1(a_n, b_n), x_2(t) = f_2(a_n, b_n),$$
$$x_3(t) = f_3(a_n, b_n), x_4(t) = f_4(a_n, b_n), \tag{18}$$

Then by substituting these trajectories into performance index (8) and computing the definite integral, also by substituting these trajectories into the constraints (6), (7), and (11), the optimal control problem is converted into a NLP as follows:

$$\min \hat{J}(a_n, b_n) \tag{18}$$

Subject to:

$$\bigcup_{i=1}^{n} [L_i(a_n, b_n, t) \geq 0], \tag{19}$$

$$P(a_n, b_n, t) \leq 0, \tag{20}$$

$$Q(a_n, b_n, t) \leq 0 \tag{21}$$

Which $\hat{J}$, L_i, P, and Q are new polynomial functions associated with cost function, obstacle constraints, maximum velocity constraint, and maximum acceleration constraint, respectively.

B. Previous Approach for Solving NLP

In [15], to solve this NLP, we solved the NLP in arbitrary points of time interval $t \in (0, t_f)$ and determined solution in every point of the time interval. Finally, desired trajectories was obtained by putting together these solutions.

C. New Solution Approach for Solving NLP

In this subsection is utilized from a novel approach that decrease simulation time dramatically. In this method the minimum value of $L_i(a_n, b_n, t)$ is computed and unknown coefficients are obtained so that the minimum be non-negative. Also maximum values of $P(a_n, b_n, t)$ and $Q(a_n, b_n, t)$ are computed and unknown coefficients are obtained so that these values be non-positive. Indeed, the NLP can be rewritten as follows:

$$\min \hat{J}(a_n, b_n) \tag{22}$$

Subject to:

$$\min\left(\bigcup_{i=1}^{k} [L_i(a_n, b_n, t)] \right) \geq 0, \tag{23}$$

$$\max\left(P(a_n, b_n, t) \right) \leq 0, \tag{24}$$

$$\max\left(Q(a_n, b_n, t) \right) \leq 0 \tag{25}$$

D. Choice of Polynomial Trajectories's Degree

In this subsection is discussed about appropriate degree of approximation polynomials.

We have four initial conditions and four final conditions. Thus eight coefficients are required to satisfy these boundary conditions which are included parameters $a_0, ..., a_3$ and $b_0, ..., b_3$. Also with assuming being equal degree of $x_1(t)$ and $x_2(t)$, we need at least two coefficients for the NLP. These unknown coefficients are a_4 and b_4.

So we have:

$$x_1(\mathrm{t}) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4, \tag{26}$$

$$x_2(\mathrm{t}) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 \tag{27}$$

V. Simulation Results

In this section, the proposed method is simulated to illustrate its effectiveness. Also simulation results are compared with the result obtained in [15].

Table I shows the simulation data in SI units for scenarios 1 and 2.

Fig.1 depicts the curves of position, velocity and acceleration of robot in scenario 1. The optimal trajectory in 2D coordinate is shown in Fig. 2.

Also total velocity and acceleration of robot in scenario 1 are shown in Fig. 3.

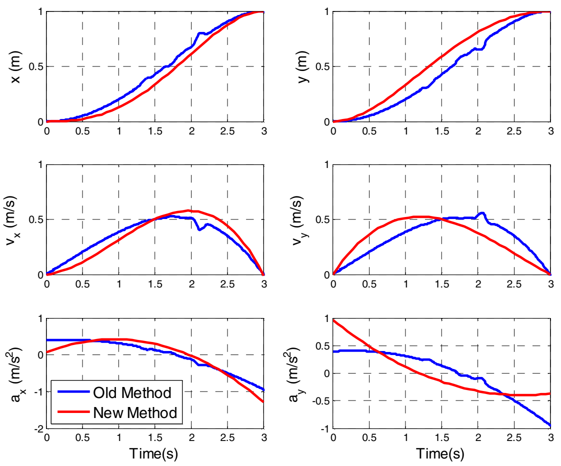

Fig. 1. Curves of position, velocity and acceleration of robot in scenario 1

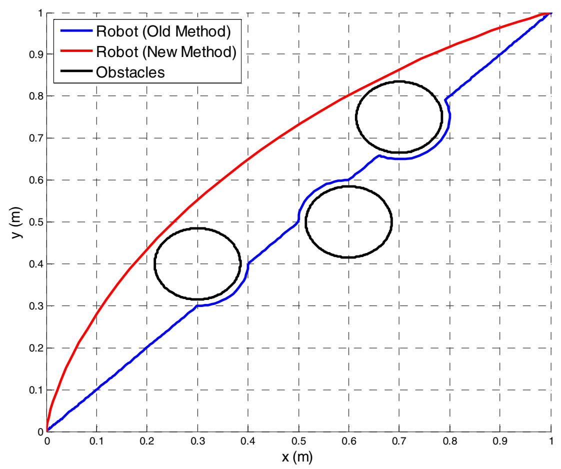

Fig. 2. Optimal trajectory in 2D coordinate in scenario 1

TABLE I. Simulation Data

	$\begin{bmatrix} x_f \\ y_f \end{bmatrix}$	$\begin{bmatrix} x_{c_1} \\ y_{c_1} \end{bmatrix}$	$\begin{bmatrix} x_{c_2} \\ y_{c_2} \end{bmatrix}$	$\begin{bmatrix} x_{c_3} \\ y_{c_3} \end{bmatrix}$	$\begin{bmatrix} x_{c_4} \\ y_{c_4} \end{bmatrix}$	t_f
1	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.6 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.7 \\ 0.75 \end{bmatrix}$	–	3
2	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$	$\begin{bmatrix} 0.4 \\ 0.35 \end{bmatrix}$	$\begin{bmatrix} 0.65 \\ 0.55 \end{bmatrix}$	$\begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix}$	3

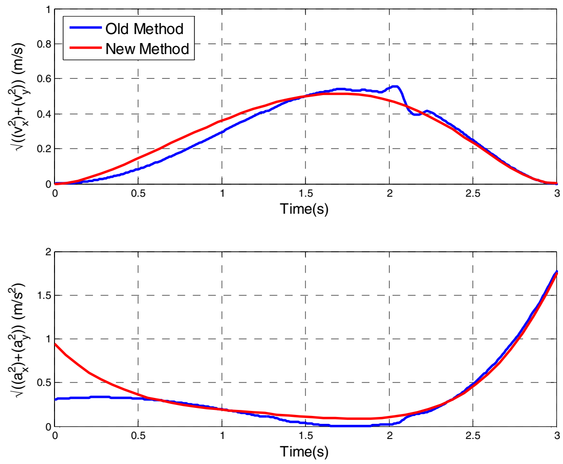

Fig. 3. Total velocity and acceleration of robot in scenario 1

In Fig. 1, we see that the robot reaches to the desired final position while the initial and final conditions on velocity of robot are satisfied and robot stops in destination point. Furthermore seen in Fig. 2 that the robot without any collision with obstacles reaches to its final position. As seen in Fig.3, the motion constraints of robot including maximum velocity and acceleration in scenario 1 are satisfied.

Approximation polynomial trajectories for scenario 1 are obtained as follows:

$$x_1(\mathrm{t}) = x(\mathrm{t}) = 0.0336t^2 + 0.1257t^3 - 0.0333t^4, \tag{28}$$

$$x_2(\mathrm{t}) = y(\mathrm{t}) = 0.4836t^2 - 0.1743t^3 + 0.0167t^4 \tag{29}$$

Also the approximation velocity and acceleration of robot in x and y direction are obtained as follows:

$$x_3(\mathrm{t}) = v_x(\mathrm{t}) = 0.0673t + 0.3772t^2 - 0.1332t^3, \tag{30}$$

$$x_4(\mathrm{t}) = v_y(\mathrm{t}) = 0.9673t - 0.5228t^2 + 0.0668t^3, \tag{31}$$

$$u_1(\mathrm{t}) = a_x(\mathrm{t}) = 0.0673 + 0.7544t - 0.3996t^2, \tag{32}$$

$$u_2(\mathrm{t}) = a_y(\mathrm{t}) = 0.9673 - 1.0456t + 0.2004t^2 \tag{33}$$

As seen, the motion constraints of robot including maximum velocity and acceleration in scenario 1 are satisfied.

Fig.4 shows the curves of position, velocity and acceleration of robot in scenario 2. We can see that the robot reaches to the desired final position while the initial and final conditions on velocity of robot are fulfilled and the robot stops by reaching to destination location. Also the optimal trajectory in 2D coordinate in scenario 2 is demonstrated in Fig.5 which seen the robot passes obstacles and reaches to its desired position.

Furthermore total velocity and acceleration of robot using two solution methods in scenario 2 is depicted in Fig.6 which seen the motion constraints including maximum velocity and acceleration of robot in scenario 2 are satisfied.

Table II compares run times of simulations and the optimal value of cost functions by two solution methods. $t_{run_{new}}$ and $t_{run_{old}}$ present the run time of simulation in proposed method and the method in [15], respectively.

Also J_{old} and J_{new} present the optimal value of performance index in proposed method and the method in [15], respectively.

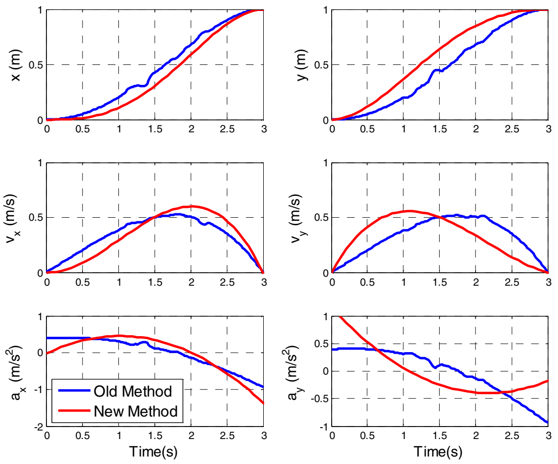

Fig. 4. Curves of position, velocity and acceleration of robot in scenario 2

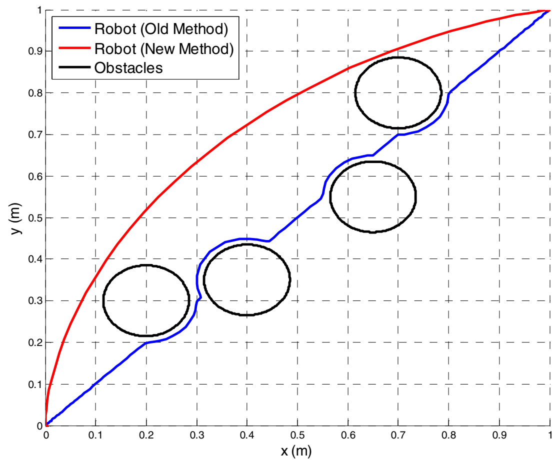

Fig. 5. Optimal trajectory in 2D coordinate in scenario 2

TABLE II. SIMULATION RESULTS

	J_{old}	J_{new}	$t_{run_{old}}$ (s)	$t_{run_{new}}$ (s)
1	7.24	7.48	42.66	0.12
2	7.27	7.69	59.52	0.1

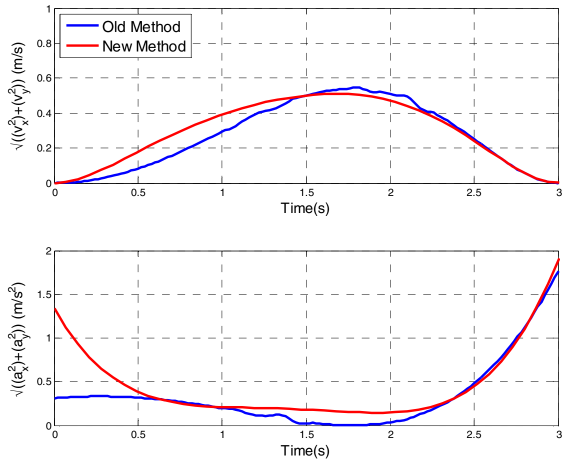

Fig. 6. Total velocity and acceleration of robot in scenario 2

Approximation polynomial trajectories for scenario 2 are obtained as follows:

$$x_1(t) = x(t) = -0.0141t^2 + 0.1575t^3 - 0.0386t^4, \quad (38)$$

$$x_2(t) = y(t) = 0.5781t^2 - 0.2373t^3 + 0.0272t^4 \quad (39)$$

Also the approximation velocity and acceleration of robot in x and y direction for scenario 2 are obtained as follows:

$$x_3(t) = v_x(t) = -0.0281t + 0.4726t^2 - 0.1544t^3, \quad (40)$$

$$x_4(t) = v_y(t) = 1.1563t - 0.7118t^2 + 0.1088t^3, \quad (41)$$

$$u_1(t) = a_x(t) = -0.0281 + 0.9452t - 0.4632t^2, \quad (42)$$

$$u_2(t) = a_y(t) = 1.1563 - 1.4236t + 0.3264t^2 \quad (43)$$

As given in Table II, the run time has been decreased sharply using new solution method in comparison with old solution method. It confirm the proposed method is appropriate for implementation in real world .Also the optimal value of performance index using new solution method has been increased too small in comparison with old solution method.

VI. CONCOLUTION

In this paper a direct method has been proposed for optimal trajectory planning of Omni-directional mobile robot in presence of static obstacles. The optimal trajectory with considering maximum velocity and acceleration of robot, minimizes a quadratic performance index. For this purpose, the problem was formulated as an optimal control problem and by a state parameterization transformed to a nonlinear programing problem (NLP) and finally solved using a novel method. The method was tested on simulation. It is shown that run time decreases dramatically using this method. Simulation results and low execution times demonstrate the viability of the proposed method as a direct numerical approach to optimal trajectory planning of Omni-directional mobile robots.

REFERENCES

[1] Rao, Anli. “A survey of numerical methods for optimal control.” *Advances in the Astronautical Sciences,* vol. 135, pp. 497-528, August 2009.

[2] Betts, John T. “Survey of numerical methods for trajectory optimization.” *Journal of guidance, control, and dynamics,* vol. 21, pp. 193-207, March 1998.

[3] Von Stryk, Oskar, and Roland Bulirsch. “Direct and indirect methods for trajectory optimization.” *Annals of operations research,* vol. 37, pp. 357-373, December 1992.

[4] 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,* vol. 37, pp. 181-218, June 1995.

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

[6] 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.

[7] Assassa, Fady, and Wolfgana Marquardt. “Dynamic optimization using adaptive direct multiple shooting.” *Computers & Chemical Engineering* 60, pp. 242-259, January 2014.

[8] Kirches, C., L. Wirsching, H. G. Bock, and J. P. Schlöder. “Efficient direct multiple shooting for nonlinear model predictive control on long horizons.” *Journal of Process Control*, vol. 22, pp. 540-550, March 2012.

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

[10] Benson, David A., Geoffrey T. Huntington, Tom P. Thorvaldsen, and Anil V. Rao., “Direct trajectory optimization and costate estimation via an orthogonal collocation method.” *journal of Guidance, Control, and Dynamics,* vol. 29, pp. 1435-1440, November 2006.

[11] Tohidi, Emran, and Hassan Saberi Nik. “A Bessel collocation method for solving fractional optimal control problems.” *Applied Mathematical Modelling,* vol. 39, pp. 455-465, January 2015.

[12] Kameswaran, Shivakumar, and Lorenz T. Biegler. “Convergence rates for direct transcription of optimal control problems using collocation at Radau Points.” *Computional Optimization and Applications,* vol. 41, pp. 81-126, September 2008.

[13] Grag, Divya, et al. “Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a Radau pseudospectral method.” *Computional Optimization and Application,* vol. 49, pp. 335-358, June, 2015.

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

[15] Mohseni, Naser Azim, and Ahmad Fakharian. "Optimal trajectory planning for an omni-directional mobile robot with static obstacles: a polynomial based approach." *AI & Robotics (IRANOPEN), 2015*. IEEE, 2015.