



Motion Planning Method for Home Service Robot Based on Bezier Curve

Yu Shu¹, Donghui Zhao^{1(✉)}, Zihao Yang¹, Junyou Yang¹, and Yokoi Hiroshi²

¹ Shenyang University of Technology, Shenyang 110870, China
zhaodonghui@sut.edu.cn

² The University of Electro-Communications, Tokyo 1820021, Japan

Abstract. Aiming at the problems of collision, non-smoothness, and discontinuity of the trajectory generated by the motion planning of the service robot in the home environment, this paper proposes a robot motion planning system consisting of three parts: path planning, trajectory generation, and trajectory optimization. First, use the A* algorithm based on graph search to quickly plan a passable global path in a complex home environment as the initial value of trajectory generation; secondly, construct an objective function based on Minimum Snap to generate the initial trajectory to be optimized; finally, Using the convex hull property of the Bezier curve, the safety corridor is constructed and the time distribution is adjusted by the trapezoidal velocity curve method, which solves the overshoot phenomenon that occurs in the solution process of the trajectory generation. The Minimum Snap method based on the Bezier Curve is constructed to optimize the trajectory and finally generate A continuous and smooth motion trajectory with minimal energy loss suitable for service robots. The feasibility and effectiveness of this method are proved by simulation experiments.

Keywords: Service robot · Trajectory optimization · Bezier curve

1 Introduction

In recent years, with the increase in the number of the aging population, service robots play a great advantage in taking care of the daily life of the elderly and are widely used in the home environment [1]. The home environment is relatively complex and unstructured, which puts higher requirements on the motion planning of service robots. The motion planning of mobile robots consists of two parts: path planning and trajectory optimization.

At present, the path planning algorithms of mobile robots mainly include two categories: search-based and sampling-based. Search-based methods mainly include A* algorithm [2,3], D* algorithm [4], the main feature of this kind of algorithm is to expand from the starting point to the endpoint until it reaches the endpoint. The sampling-based method is another kind of commonly used path planning method, among which the most classic is the rapidly exploring

random tree algorithm [5], but since it uses random sampling, the planned path is not the optimal path. Due to the high center of gravity of the service robot, it is necessary to ensure the stability of the robot's movement, which puts forward higher requirements for the trajectory optimization of the robot. Through the differential flat method [6], polynomials can be used to optimize the trajectory of various robots [7]. However, the generated trajectory only considered the smoothness and continuity constraints of the intermediate points, and did not consider the shape of the trajectory, that is, whether a collision will occur.

In this paper, a motion planning method for service robots in the home environment is proposed. By imposing security corridor constraints on position and redistributing time by trapezoidal velocity method, a trajectory optimization method based on the Minimum Snap of Bezier curve is constructed, which effectively solves the motion planning problem of robots in the home environment.

2 Motion Planning

2.1 Overall Structure

The motion planning method of the home service robot proposed in this paper consists of three parts: path planning, trajectory generation, and trajectory optimization, and the overall structure block diagram is shown in Fig. 1. Firstly, the A* algorithm is used to realize the initial path planning and get a set of discrete waypoints that do not collide with the obstacles. On this basis, an objective function based on Minimum Snap is constructed to generate a continuous, smooth, and to-be-optimized trajectory. However, the generated trajectory suffers from a certain amount of overshoot. Therefore, the Bezier curve is introduced, the safety corridor is constructed according to the obstacle distribution, the time is redistributed using the trapezoidal velocity curve method, and the trajectory optimization of Minimum Snap based on the Bezier curve is constructed using the transformation relationship between the Bezier and polynomial functions to finally generate a continuous, smooth motion with minimum energy loss applicable to the home service robot trajectory. The trajectory is input to the robot trajectory tracker, and the service robot can then track the trajectory.

2.2 Trajectory Generation Based on Minimum Snap

When the home environment information is known, given the initial state and the target state, the A* algorithm is used to obtain a path consisting of a series of discrete waypoints that are sparse and do not contain temporal information. To better control the service robot's motion, the sparse waypoints need to be transformed into continuous and smooth curves. In this paper, we use a higher-order polynomial function for trajectory generation. The polynomial coefficients are calculated so that the overall polynomial trajectory satisfies the continuity and smoothness constraints while minimizing the energy function. To be able to restrict the snap and satisfy the equation semi-positive definite, construct a 7th order polynomial objective function.

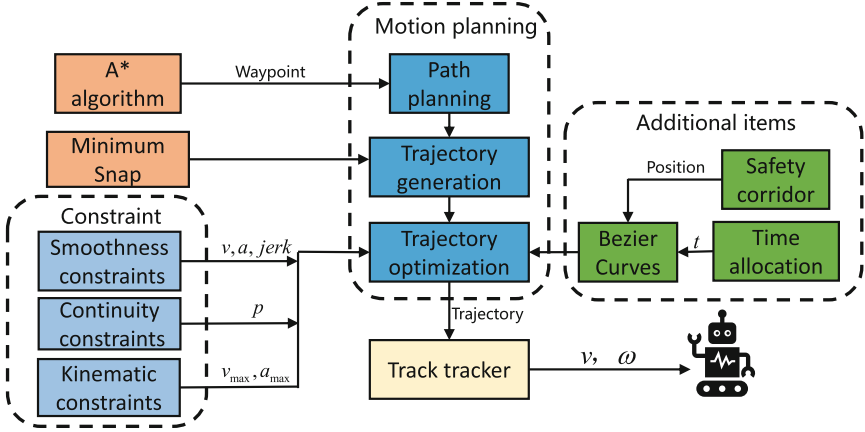


Fig. 1. Block diagram of the overall structure of motion planning.

Let the path be divided into a total of h segments, then any of these segments can be expressed as:

$$q(t) = q_0 + q_1t + q_2t^2 + \dots + q_7t^7 = \sum_{i=0}^7 q_i t^i \quad (1)$$

where $q(i)$ is the coefficient of the polynomial ($i = 0, 1, \dots, 7$), and t is the time of current trajectory.

The Minimum Snap minimization objective function of the whole trajectory is constructed, that is $Z(t)$, the integral of the square of the change of snap in the corresponding time segment of each trajectory, and its expression is:

$$Z(T) = \int_{T_{j-1}}^{T_j} \left(f^{(4)}(t) \right)^2 dt = Q_j^T L_j Q_j \quad (2)$$

where Q_j is the coefficient matrix of j^{th} trajectory polynomial; L_j is the information matrix of j^{th} trajectory polynomial.

The problem of minimization of the objective function is modeled as a mathematical Quadratic Programming (QP) problem. After constructing the objective function to be optimized, to make the final generated trajectory continuous, the continuity equation constraint is constructed at the end of the j^{th} segment trajectory and the beginning of the $(j + 1)^{th}$ segment trajectory as:

$$f_j^{(k)}(T_j) = f_{j+1}^{(k)}(T_j) \quad (3)$$

The position of the trajectory at the path junction of each segment of the trajectory is relatively fixed, and the smoothness equation constraint is constructed:

$$f_j^{(k)}(T_j) = c_j^{(k)} \quad (4)$$

where c represents the x and y axes.

Formula (3) and Formula (4) are substituted for Formula (2), and the coefficients q of each trajectory satisfying the constraint conditions are calculated. The coefficients are substituted into the state equations of each segment to solve the state quantities of the whole trajectory in all directions, and finally, the trajectory to be optimized is obtained.

2.3 Trajectory Optimization Based on Bezier Curve

The trajectory generated based on the Minimum snap method does not impose restrictions on the whole trajectory position. The generated trajectory may collide with the obstacle because of the dense obstacles in the home environment. And the trajectory generated based on the Minimum snap method does not consider the speed and acceleration running limit of the robot itself. Based on the above problems, the trajectory optimization is carried out using the Minimum snap method based on the Bezier Curve, and the mathematical properties of the Bezier Curve are used to construct a safe corridor, and kinematic constraints are imposed according to the parameters of the robot, and the time is redistributed using the trapezoidal velocity curve. The Bezier curve polynomial is established as follows:

$$E_j(t) = p_j^0 e_n^0(t) + p_j^1 e_n^1(t) + \cdots + p_j^n e_n^n(t), t \in [0, 1] \quad (5)$$

where p_j^i is the $(i + 1)^{th}$ control points of a convex hull polygon, $e_n^i(t)$ is the Bernstein basis function.

There is a transformation relationship between the Bezier curve and the polynomial curve, such as the formula (6). Therefore, we can use the Bezier property to impose additional velocity and acceleration constraints, and then transform them into polynomial curves to solve them. This will greatly shorten the calculation time and improve the solution quality.

$$\sum_{i=0}^n p_i \frac{n!}{(n-i)!i!} (1-t)^{n-i} t^i = q_0 + q_1 t + \cdots + q_n t^n \quad (6)$$

The objective function of the original trajectory will start at the first control point and end at the last control point after the Bezier Curve transformation. By restricting the control points of the Bezier curve to the rectangular safety zone, the generated trajectory will be surrounded by the convex envelope formed by the control points due to the convex envelope nature of the Bezier curve [8], and then the whole trajectory must be located within the safety zone and must be collision-free. To avoid collision between trajectory and obstacles, the trajectory expected to be planned must be in Corridor. If the passable safety region is added to the QP problem as a constraint, the trajectory of the solution will naturally be in Corridor. As shown in Fig. 2, a collision-free safe corridor is constructed with the waypoints searched by the A* algorithm as the center, and a trajectory is generated in the corridor, so the trajectory must be safe and collision-free. The inequality constraint expression is as follows:

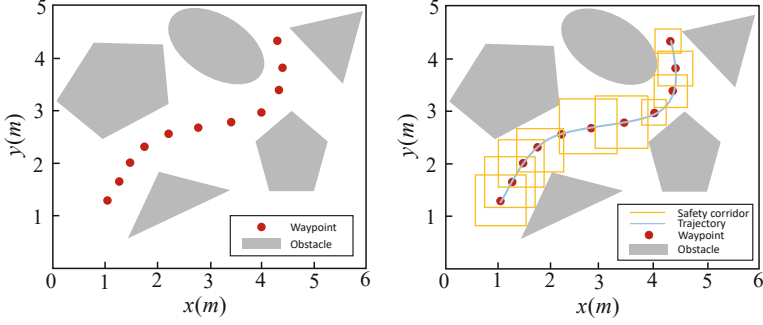


Fig. 2. Establishment of global security corridor.

$$\delta_{\mu_j}^- \leq d_{\mu_j}^{0,i} m_j \leq \delta_{\mu_j}^+ \tag{7}$$

where $d_{\mu_j}^{0,i}$ is the control point of the Bezier curve of order r^{th} and i on the j^{th} trajectory on the μ axis, $\delta_{\mu_j}^-, \delta_{\mu_j}^+$ is the minimum and maximum limit position of safety corridor.

Based on the linear relationship between the high-order control points and the low-order control points of the Bezier curve, the dynamic inequality constraints are constructed as follows:

$$\begin{cases} v_m^- \leq d_{\mu_j}^{1,i} \leq v_m^+ \\ a_m^- \leq d_{\mu_j}^{2,i} m^{-1} \leq a_m^+ \end{cases} \tag{8}$$

where v_m^-, v_m^+ is the minimum and maximum speed; a_m^-, a_m^+ is the minimum and maximum acceleration.

Equality constraints are imposed on the position, velocity, acceleration and jerk of the starting point and end point, which are expressed as:

$$\begin{cases} d_{\mu_j}^{r,i} m_j^{1-r} = g_{\mu_j}^{(r)} \\ d_{\mu_j}^{r,i} = \frac{n!}{(n-r)!} (d_{\mu_j}^{r-1,i+1} - d_{\mu_j}^{r-1,i}) \end{cases} \quad r \geq 1 \tag{9}$$

where m_j^{1-r} is the scalar coefficient of order $1 - r$ of the j^{th} segment, $g_{\mu_j}^{(l)}$ is the j^{th} segment r^{th} order on the axis is a certain value.

To ensure the position, velocity, acceleration, and jerk of every two Bezier Curve joints, a continuity equality constraint is imposed, which is expressed as:

$$d_{\mu_j}^{r,n} m_j^{1-r} = d_{\mu_{j+1}}^{r,0} m_{j+1}^{1-r} \tag{10}$$

Most current time allocation methods use uniform motion to calculate time, as shown in Fig. 3(a), time is allocated in proportion to the path length [9], but the solver is not easy to solve. In Fig. 3(c), the blue and orange colors are different time allocations to produce different trajectories. In this paper, the trapezoidal

velocity profile method is used to allocate time, where the mobile robot starts from the starting point, accelerates smoothly to the maximum speed, moves at a uniform speed, and decelerates smoothly near the endpoint. The velocity profile is shown in Fig. 3(b), and the robot acceleration function is designed as follows:

$$a = g_s (1 - \cos(t)) \quad (11)$$

where g_s is the expected maximum jerk value.

By constructing the objective function of minimizing energy and imposing linear equality constraints and linear inequality constraints, the whole trajectory optimization problem is expressed as:

$$\begin{aligned} \min \quad & P^T Q_o P \\ \text{s.t.} \quad & A_{eq} P = B_{eq} \\ & A_{ie} P \leq B_{ie} \\ & P_l \in \Omega_l, \quad l = 1, 2, \dots, h \end{aligned} \quad (12)$$

where $P = [P_1, P_2, \dots, P_h]$ consists of the feasible region Ω_l in section l that satisfies the waypoint constraint and the safety corridor constraint.

The QP solver is used to solve the above problems, and finally, a continuous, smooth, collision-free, and executable trajectory suitable for home service robot is generated.

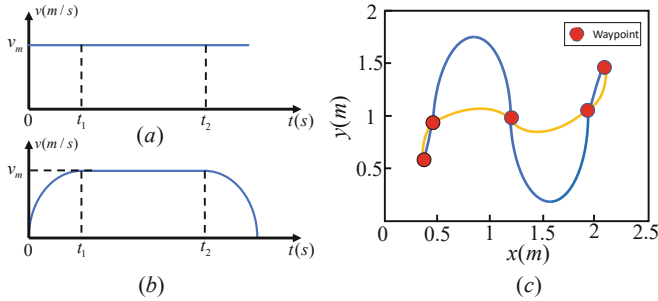


Fig. 3. Time allocation. (a) Uniform time allocation velocity curve. (b) Trapezoidal time allocation velocity curve. (c) Trajectories produced by different time allocation. (Color figure online)

3 Experiment

To validate the proposed method, Ubuntu 16.04 is used as the bottom operating system, and C++ programming is used in the ROS stage simulation environment on I5-7200HQ 8G memory portable computer. In the ROS Stage simulator, To build the robot navigation environment model in the home environment, the Gmapping algorithm is used to build the map, and the map can be started by configuring the parameter file and providing the related topic topics subscribed.

Using Rviz visualization tool in ROS to display simulation results. Service robot $v_{\max} = 0.25 \text{ m/s}$, $a_{\max} = 0.25 \text{ m/s}^2$. Firstly, the starting point and the target point are selected, and then the robot trajectory is planned by using the A* algorithm and the trajectory optimization algorithm proposed in this paper under simple obstacles and complex obstacles. The simulation results in a simple environment and a complex environment are shown in Figs. 4 and 5.

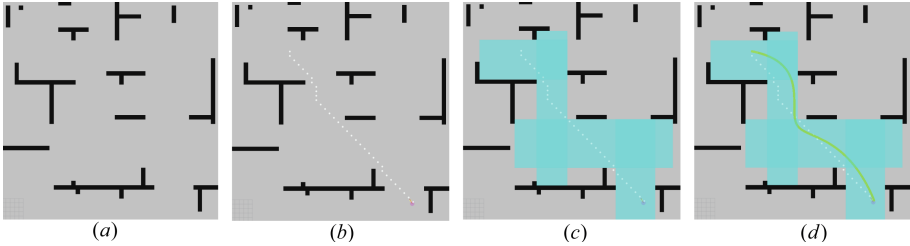


Fig. 4. Simulation results in a simple environment. (a) Environment map. (b) A* algorithm planning path. (c) Safe corridor generation. (d) Optimized trajectory.

In the simulation results, the white discrete points are the paths planned by A*, the light blue area is the safety corridor extended according to waypoints, and the green trajectory is the optimized trajectory within the safety corridor. Analyzing the simulation results, There are 24 obstacles in a simple environment, the path planned by the A* algorithm has 7 step path turns, which is not good for robot execution, while in the optimized path there are only 2 smooth turns, which is good for the robot's stable driving. There are 49 obstacles in a complex environment, the path planned by the A* algorithm has 12 step turns and 90° turns, while the optimized path has only 3 smooth turns, which greatly improves the stability of the robot.

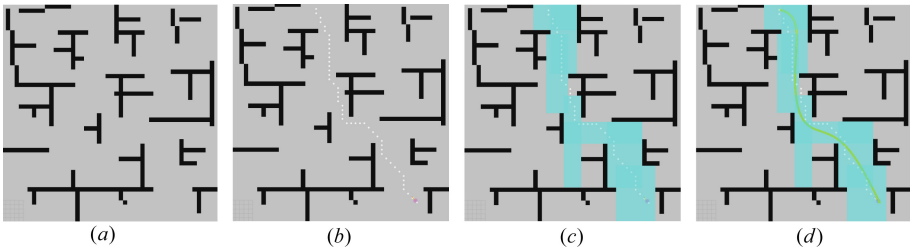


Fig. 5. Simulation results in a complex environment. (a) Environment map. (b) A* algorithm planning path. (c) Safe corridor generation. (d) Optimized trajectory.

The simulation results show that the optimized trajectory of the motion planning method proposed in this paper is better than the traditional A* planning in terms of smoothness and safety, and the obtained trajectory is far away

from obstacles and improves safety. Especially in an environment with complex obstacles, the optimization effect is more obvious and more beneficial to the application of service robots in the home environment.

4 Conclusion

In this paper, a motion planning method for service robots in the home environment is proposed. Trajectory optimization based on Bezier curve Minimum Snap is constructed, security corridor constraint is imposed, and trapezoidal velocity curve is introduced to redistribute time. The feasibility of this method is tested in ROS. By comparing the effect of this method with the traditional method, we can see that the proposed motion planning method can plan a smoother and safer path in both simple and complex environments, and is suitable for the motion planning of service robots in the home environment.

References

1. Tagliavini, L., et al.: On the suspension design of paquitop, a novel service robot for home assistance applications. *Machines* **9**(3), 52 (2021)
2. Guruji, A.K., Agarwal, H., Parsediya, D.K.: Time-efficient A* algorithm for robot path planning. *Procedia Technol.* **23**, 144–149 (2016)
3. Yang, R.J., Cheng, L.: Path planning of restaurant service robot based on a-star algorithms with updated weights. In: 2019 12th International Symposium on Computational Intelligence and Design (ISCID), pp. 292–295. IEEE, Hangzhou (2019)
4. Ferguson, D., Stentz, A.: Using interpolation to improve path planning: the field D* algorithm. *J. Field Robot.* **23**(2), 79–101 (2006)
5. Wang, W., et al.: An improved RRT path planning algorithm for service robot. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1824–1828. IEEE, Chongqing (2020)
6. Zhou, J., et al.: Pushing revisited: differential flatness, trajectory planning, and stabilization. *Int. J. Robot. Res.* **38**(12–13), 1477–1489 (2019)
7. Shomin, M., et al.: Fast, dynamic trajectory planning for a dynamically stable mobile robot. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3636–3641. IEEE, Illinois (2014)
8. Preiss, J.A., et al.: Downwash-aware trajectory planning for large quadrotor teams. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 250–257. IEEE, Vancouver (2017)
9. Mellinger, D., et al.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2520–2525. IEEE, Shanghai (2011)