Proceedings of the 2014 IEEE
International Conference on Robotics and Biomimetics
December 5-10, 2014, Bali, Indonesia

Smooth Path Planning for a Home Service Robot Using η^3-Splines

Sen Zhang[1, 2], Lei Sun[1, 2,*], Zhongliang Chen[1, 2], Xiang Lu[1, 2], and Jingtai Liu[1, 2]

Abstract—**This paper presents a smooth path planning algorithm for a home service robot in presence of a known map and static obstacles. The algorithm proceeds in two steps. Firstly, the MAKLINK Graph based global path planner generates the shortest linear path which consists of several line segments. By introducing the 2D code landmarks into the global path planner, the path passes the landmarks as many as possible to reduce the motor encoder error. In the second step, the waypoints of the linear path are smoothly connected using the η^3 -splines. Considering the obstacle-avoiding issue, the feasible area is proposed in this paper to generate the path in a collision-free region. At last, the simulation and experiment results validate the feasibility of the smooth path planning algorithm.**

I. Introduction

GENERALLY, there are two frameworks regarding the appropriate path generation for autonomous mobile robots. Motion planning considers the problem of joining two given points according to the known map and obstacles. Motion generation, on the other hand, considers the planning problem with strong dynamic environment. This paper focuses on the path planning problem for a home service robot in presence of a known map and static obstacles.

Since Dubin [1] first generated the shortest length path using linear segments and circular arcs, many pieces of work on the path planning problem have been published. T. Fraichard et al. [2] presented the first steering method for a car-like vehicle, which considers the generation of continuous curvature paths. J. Villagra et al. [3] proposed an obstacle-avoiding path planning algorithm for high velocity wheeled mobile robots with uniform quartic B-splines.

In recent years, A. Piazzi et al. published a kind of seventh order polynomial splines, i.e. the η^3-splines [4]. Using the η^3-splines and island parallel genetic algorithm, a flexible collision-free smooth path planning scheme is presented in [5]. However, four parameters are set to zero to simplify the η^3-splines. Regarding the optimization of the η^3-splines, C. Guarino et al. [6] generated paths with minimum curvature derivative and G. Lini et al. [7] proposed a multi-optimization approach to the autonomous parking problem.

Recently, F. Ghilardelli et al. [8] published the η^4 -splines to generate high-quality drive paths for a truck and trailer vehicle. The η^4 -splines fit well with the path generation of articulated vehicles.

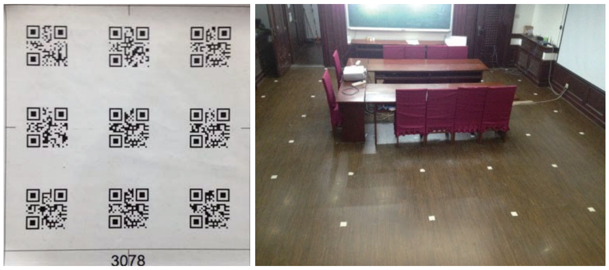
Based on the η^3 -splines, we study the smooth path planning problem for a two-wheeled differential driving home service robot. The contributions of this paper can be concluded as follows. Firstly, the 2D code landmarks are brought into the global path planner so that as many landmarks can be chosen as possible. With the robot location calculated from the landmarks, the motor encoder error can be reduced. Secondly, the obstacle-avoiding issue in the application of the η^3 -splines is considered in the paper. Under the constraint of the feasible area, the generated path is both smooth and collision-free.

The paper is organized as follows. Section II introduces our preliminary work on the application of the 2D code landmarks in the home environment. Section III explicitly presents the smooth path planning algorithm. Section IV briefly describes the three-layer Home Information Center (HIC) architecture and explains how the path planning algorithm works based on the HIC architecture. Section V presents the experiment result and concluding remarks are drawn in Section VI.

II. Preliminary Work on the 2D Code Landmarks

A. 2-Dimensional Code Landmarks

Quick Response (QR) code is a kind of cheap and maturing 2D code and possesses outstanding error correcting capacity. Therefore, we utilize the most commonly used commercial QR codes as the artificial landmarks. To eliminate the effect of the modifications to the floor, the 2D code landmarks can be invisible and seen under UV illumination.



(a) (b)

Fig. 1. 2D code landmark and deployment.

Manuscript received July 25, 2014. This work is supported by the National High Technology Research and Development Program (863 Program) of China under Grant 2012AA041403.

Sen Zhang, Lei Sun, Zhongliang Chen, Xiang Lu and Jingtai Liu are with1, 2 (e-mail: {zhangs, sunl, chenzl, liujt} @ nankai.edu.cn).

1. Institute of Robotics and Automatic Information System, Nankai University, Tianjin, 300071.

2. Tianjin Key Laboratory of Intelligent Robotics.

Address all correspondence to Lei Sun (phone: 13512967601; fax: 022-23500172).

978-1-4799-7397-2/14/$31.00 © 2014 IEEE

As in Fig. 1(a), each landmark label contains 9 QR codes so that the robot is more likely to detect the landmark while moving. The landmarks are deployed at interval of 1m (see Fig. 1(b)), and the position of each landmark is corrected with a laser horizontal instrument.

B. Initial Location Calibration with the 2D Code Landmarks

In our preliminary research, we proposed a low cost and convenient initial location calibration solution for the home service robot. As in Fig. 2, the robot searches for the nearest 2D code landmark to calculate its initial location.

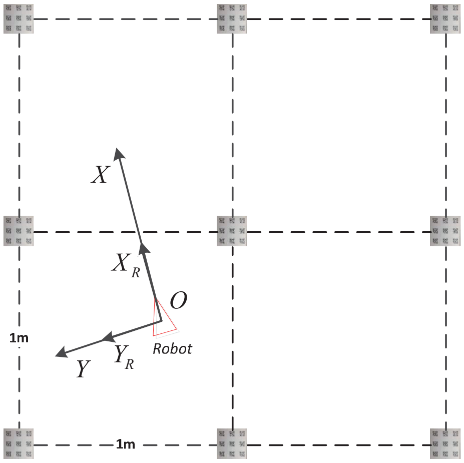


Fig. 2. Schematic of the initial location calibration.

Two search strategies are proposed: given no information of the robot's surroundings, the robot adopts the blind search strategy and tracks a spiral trajectory to find the nearest landmark; with the help of a Kinect RGB camera, the robot estimates the location of the nearest landmark according to the landmarks in the Kinect vision. The experiment results have validated the feasibility of the proposed solution.

C. Motor Encoder Calibration with the 2D Code Landmarks

It is well known that a motor encoder is likely to drift after a long time of running. With the help of the 2D code landmarks, the motor encoder can be calibrated with accurate location of the robot when the robot moves over a landmark. In this way, the motor encoder error ran be reduced. And the 1m interval between two adjacent landmarks ensures that the motor encoder can be calibrated in time.

Based on the important role that the 2D code landmarks can play in the application of home service robots, the 2D code landmarks are introduced into the smooth path planning algorithm. The algorithm is presented in the rest of the paper.

III. Smooth Path Planning

The smooth path planning algorithm proposed in this paper proceeds in two steps. In the first step, the global path planner generates the shortest linear path. Then the waypoints of the linear path are smoothly connected to generate a collision-free path. The algorithm is shown in Fig. 3.

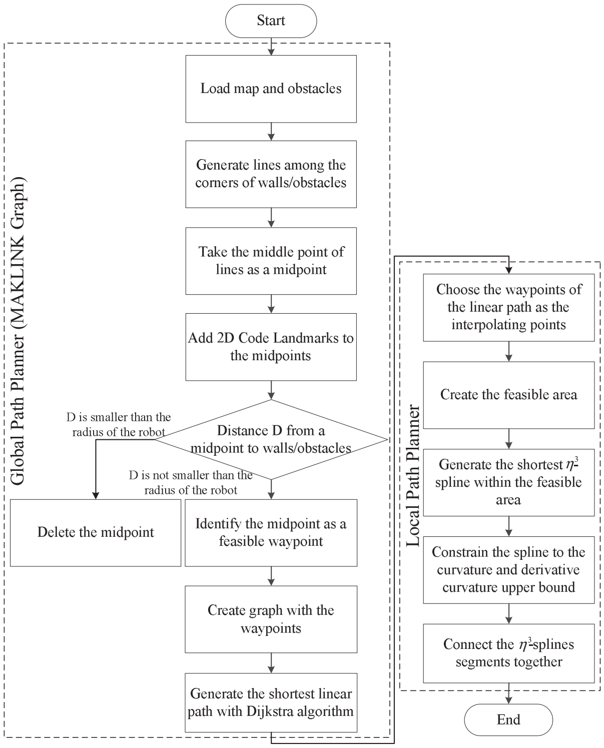

Start
Global Path Planner (MAKLINK Graph)
Load map and obstacles
Generate lines among the corners of walls/obstacles
Take the middle point of lines as a midpoint
Add 2D Code Landmarks to the midpoints
Distance D from a midpoint to walls/obstacles
D is smaller than the radius of the robot
D is not smaller than the radius of the robot
Delete the midpoint
Identify the midpoint as a feasible waypoint
Create graph with the waypoints
Generate the shortest linear path with Dijkstra algorithm
Local Path Planner
Choose the waypoints of the linear path as the interpolating points
Create the feasible area
Generate the shortest η^3-spline within the feasible area
Constrain the spline to the curvature and derivative curvature upper bound
Connect the η^3-splines segments together
End

Fig. 3. The smooth path planning algorithm.

A. Global Path Planner

As in Fig. 3, the global path planner is based on the MAKLINK Graph algorithm [9]. The corners of the environment or the obstacles are connected with straight lines. If the distance from a line's midpoint to a wall or an obstacle is larger than the robot's diameter, then the midpoint is considered as a feasible waypoint. Using the Dijkstra algorithm, the shortest linear path is generated and the global path planning result is shown in Fig. 4.

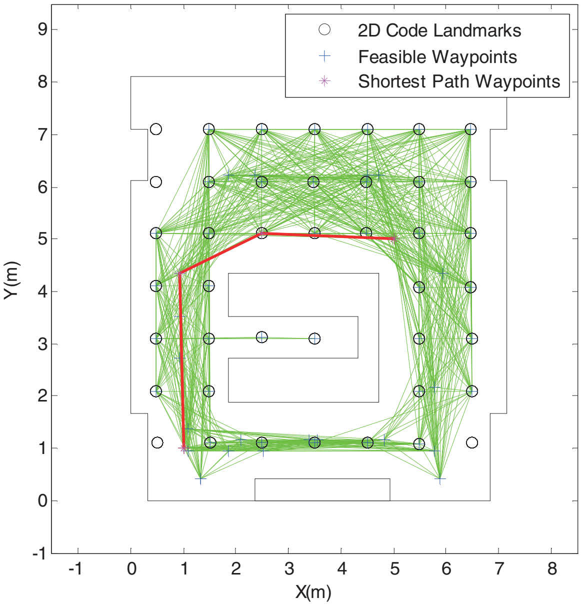


Fig. 4. Global path planning result without the 2D code landmarks.

As in Fig. 4, the landmarks are not necessarily to be passed, so the situation in Fig. 4 may happen occasionally. To include more landmarks to the generated linear path, the 2D code landmarks are introduced into the classic MAKLINK Graph algorithm as midpoints. After the same procedures, several landmarks are identified as the feasible waypoints. Compared with the path generated in Fig. 4, more landmarks are passed in Fig. 5. As mentioned in Section II, the path in Fig. 5 is more helpful to the motor encoder calibration.

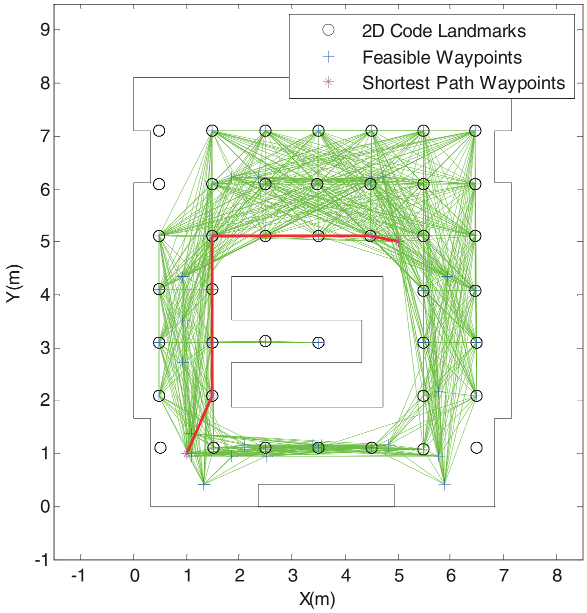


Fig. 5. Global path planning result with the 2D code landmarks. By modifying the graph weight related to the landmarks, more landmarks can be included to the generated linear path.

B. Local Path Planner

As in Fig. 5, the path generated by the global path planner is made up of several line segments. In practical applications, the robot has to stop at the interpolating point to change its orientation owing to the discontinuous curvature. Therefore, the waypoints of the linear path must to be smoothly connected.

a. The η^3-splines and correlated characteristics

The η^3-splines are seventh order polynomial curves and permit the smooth connection with arbitrary interpolating condition. As in [4], the η^3-splines can be parameterized by $p(u)=[x(u),y(u)]^T, u\in[0,1]$. The η^3-splines are defined as (1), where $\alpha_i,\beta_i, i=0,\ldots,7$ represents the parameters of the seventh order polynomial.

$$\begin{aligned} x(u) &= \alpha_0+\alpha_1 u+\alpha_2 u^2+\alpha_3 u^3+\alpha_4 u^4+\alpha_5 u^5+\alpha_6 u^6+\alpha_7 u^7 \\ y(u) &= \beta_0+\beta_1 u+\beta_2 u^2+\beta_3 u^3+\beta_4 u^4+\beta_5 u^5+\beta_6 u^6+\beta_7 u^7 \end{aligned} \qquad (1)$$

The $\alpha_i,\beta_i, i=0,\ldots,7$ are the functions of the interpolating parameters and the shaping parameters. The interpolating parameters include the position, the unit tangent vector, curvature and curvature derivative of the endpoints, while the shaping parameters are the $\eta_i, i=1,\ldots,6$. Please refer to [4] for detailed explanations.

The η^3-splines are well-suited for the application where two points need to be smoothly connected with arbitrary interpolating conditions. Given the 4 points A, B, C, D in Fig. 6, when the interpolating condition are set, they can be smoothly connected using the η^3-splines. Each curve can be reshaped with different shaping parameters $\eta_i, i=1,\ldots,6$.

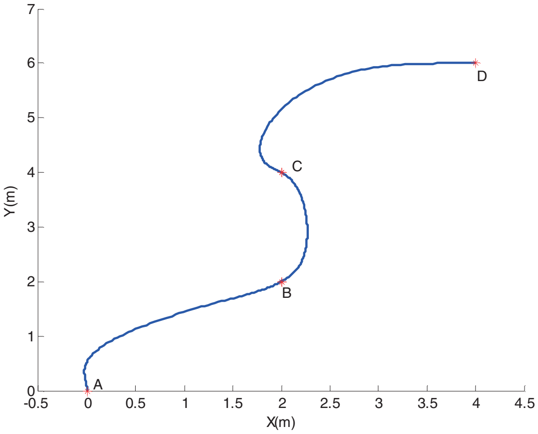


Fig. 6. Smooth connection of A, B, C, D using the η^3-splines.

The interpolating conditions and the shaping parameters of each curve are shown in Table I and Table II. It is worth noting that the interpolating conditions at the joint of two η^3-splines must agree so that the path at the interpolating point will be continuous and smooth.

TABLE I
THE INTERPOLATING CONDITION OF EACH CURVE

Curve	P_S	P_E	θ_S	θ_E	K_S	K_E	$\dot{K}_S$	$\dot{K}_E$
AB	0 0	2 2	$\frac{\pi}{2}$	$\frac{\pi}{4}$	1	1.5	1	-1.5
BC	2 2	2 4	$\frac{\pi}{4}$	$\frac{3\pi}{4}$	1.5	2	-1.5	3
CD	2 4	4 6	$\frac{3\pi}{4}$	0	2	0	3	0

Table I presents the interpolating parameters of each curve. Note that the interpolating conditions at the joint of two curves must agree.

TABLE II
PARAMETERS η_i USED TO GENERATE η^3-SPLINES

Curve	η_1	η_2	η_3	η_4	η_5	η_6
AB	2	2	0	0	0	0
BC	1	2	2	1	3	4
CD	1	5	2	2	4	5

Table II shows the shaping parameters of each curve. The shape of the curve is greatly influenced by the choice of $\eta_i, i=1,\ldots,6$.

b. *Generate the shortest smooth path using the η^3-splines*

Based on the outstanding connectivity of the η^3-splines, the waypoints of the shortest linear path can be connected using the η^3-splines to compose a smooth and continuous path. In order to minimize the arc length of each η^3-spline, and limit the curvature and curvature derivative to the upper bound, the optimal problem can be described as (2) – (4), where $k_{\max}$ and $\sigma_{\max}$ separately represent the upper bound of curvature and curvature derivative.

$$\min_{\eta\in\mathcal{H}} \int_{u_0}^{u} \sqrt{\dot{x}(\xi)^2 + \dot{y}(\xi)^2}\, d\xi \qquad (2)$$

$$\left\| k_c(u) \right\| \le k_{\max} \qquad (3)$$

$$\left\| \dot{k}_c(u) \right\| \le \sigma_{\max} \qquad (4)$$

The curvature $k_c(u)$ and the first derivative of $k_c(u)$ with respect to u are described in (5) – (6),

$$k_c(u) = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}+\dot{y})^{3/2}} \qquad (5)$$

$$\dot{k}_c(u) = \frac{(\dot{x}\dddot{y} - \dddot{x}\dot{y})(\dot{x}+\dot{y})^2 - 3(\dot{x}\ddot{y} - \ddot{x}\dot{y})(\dot{x}\ddot{x} + \dot{y}\ddot{y})}{(\dot{x}+\dot{y})^{5/2}} \qquad (6)$$

The reciprocal of curvature represents the radius of the current curve, while the derivative curvature is related to the steering speed. A large derivative curvature may cause lateral solicitation when the robot moves at high velocity. Unlike the path planning for car-like vehicles [2] and high velocity robots [3], the two-wheeled differential driving home service robot has no requirement for high velocity. Therefore, the influence of the lateral solicitation phenomenon is limited. To make sure that there are solutions to the optimal problem, the curvature upper bound is defined as the radius of the robot and the curvature derivative is not strictly constrained.

A smooth and continuous path is shown in Fig. 7 under the constraint of (2) – (4). However, in order to satisfy the curvature and curvature derivative requirement, the path which is generated using the η^3-splines is not necessarily collision-free. In the worst case (as in Fig. 7), the path may collide with a wall or an obstacle in the environment. Consequently, the obstacle-avoiding problem is an important issue to be considered.

c. *Generate the collision-free path using the η^3-splines*

In order to solve the obstacle-avoiding problem mentioned above, the feasible area is introduced to the local path planner to generate the path in a collision-free area.

As mentioned in the global path planner, the feasible waypoints are generated based on the distance from a midpoint to a wall or an obstacle. The midpoint is considered

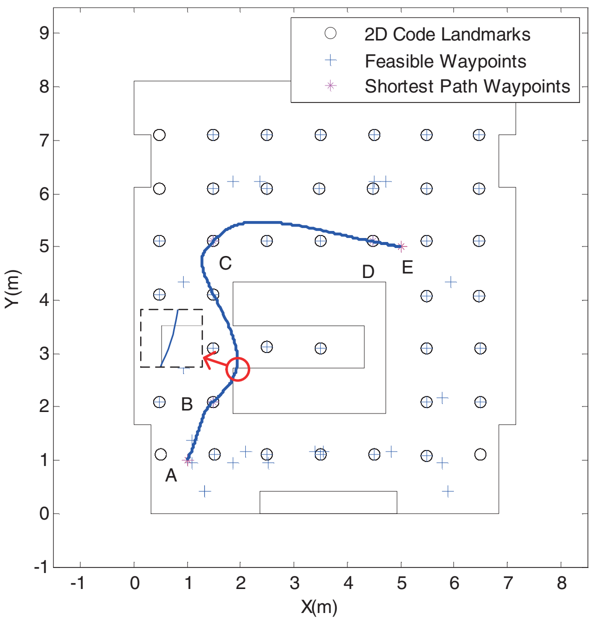


Fig. 7. Path AE collides with an obstacle. Although the path is smooth and continuous, it still collides with an obstacle.

as a feasible waypoint only when the distance is larger than the robot's diameter. Based on this thought, the feasible area is constructed with the two parallels of every linear path segment (see in Fig. 8), and the robot's radius is chosen as the vertical interval from the linear path to the parallels. In this way, the feasible area guarantees the generated path to be collision-free. The planning result is shown in Fig.8.

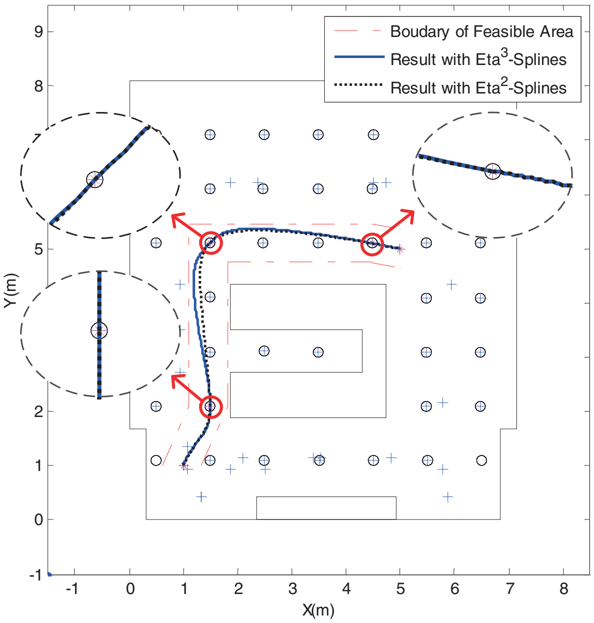


Fig. 8. Smooth path planning result. The parallels of each linear path segment compose the boundary of the feasible area.

According to the constraints (2) – (6), the shaping parameters of each η^3-spline are solved based on the Sequential Quadratic Programming (SQP) algorithm (see the shaping parameters in Table III).

TABLE III
PARAMETERS η_i USED TO GENERATE η^3-SPLINES

Curve	η_1	η_2	η_3	η_4	η_5	η_6
AB	1.204	1.204	0	0	0	0
BC	1.500	1.916	0.500	0.500	0.500	3.000
CD	1.500	1.500	0.500	2.990	0.500	0.500
DE	0.530	0.530	0	0	0	0

The parameters η_i in Table III are calculated based on the Sequential Quadratic Programming (SQP) algorithm.

The planning result using the η^2-splines [10] is also shown in Fig. 8 (see the dotted line) for comparison. As in Fig. 9, the curvatures of the two paths are both continuous and bounded. However, the curvature of the path, which is based on the η^3-splines, changes more gently.

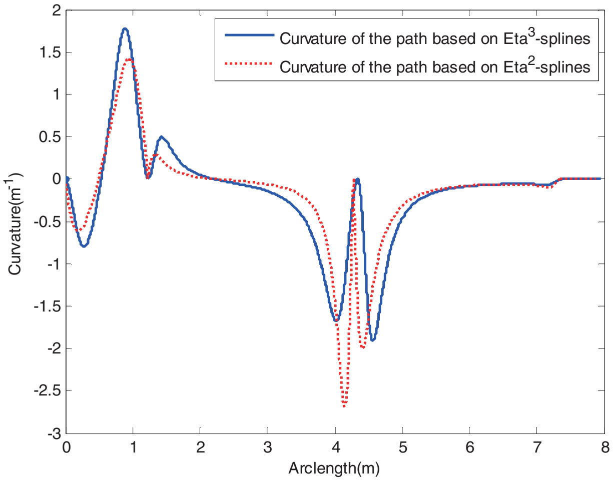


Fig. 9. The curvatures of the generated paths in Fig. 8.

The derivative curvatures of the two paths in Fig. 8 are shown in Fig. 10. The η^3-splines guarantee that the derivative curvature of the path is not only bounded but also continuous.

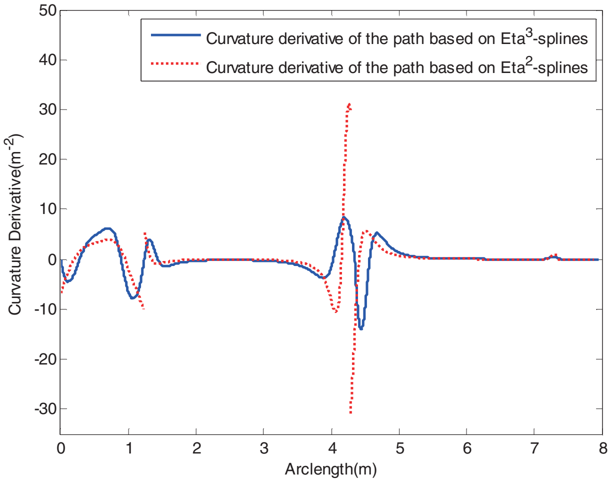


Fig. 10. The derivative curvatures of the generated paths in Fig. 8.

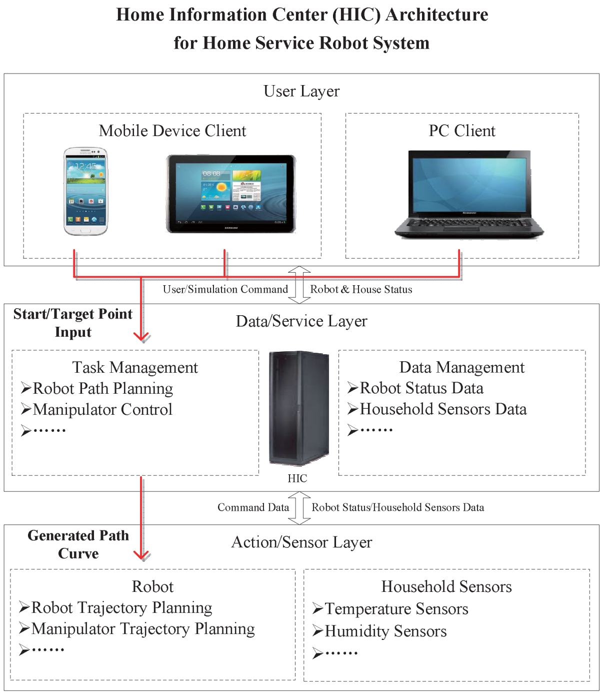


Fig. 11. Three-layer home information center (HIC) architecture.

IV. HIERARCHICAL PATH PLANNING STRUCTURE

Based on the light-client design, a kind of three-layer Home Information Center (HIC) architecture (see in Fig. 11) is proposed in this paper.

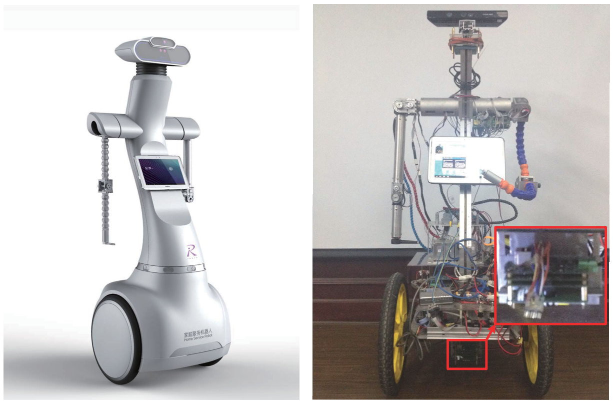
As in Fig. 11, the HIC architecture consists of User Layer, Data/Service Layer and Action/Sensor Layer. HIC is the core of the architecture. By means of networked technology, the HIC receive and transmit the user command or system status data to realize robot control and status query. Based on the modularized and hierarchical design, the HIC encapsulates modules according to the type of each task and divides tasks into upper tasks and lower tasks. The upper tasks are operated in the HIC, while the robot is simplified to an actuator to run the lower tasks. By introducing the HIC into the home service robot system, the robot becomes lighter and faster.

Regarding the path planning of the home service robot, the smooth path planning algorithm is implemented on the Data/Service Layer. Through wireless network, the generated path is transmitted to the robot, and the trajectory planning is finished by the robot (see the red arrow in Fig. 11).

V. PATH PLANNING EXPERIMENT RESULT

A. Home Service Robot Prototype

The home service robot (see Fig. 12(a)) mentioned in this paper is a two-wheeled differential driving robot with a supporting roller on the rear. As in Fig. 12(b), a Kinect RGB camera is mounted on the top for gesture recognition and indoor 3D mapping, and an Android pad is used for robot control and human-robot interaction. The decoding system is installed under the chassis for landmarks decoding (see the rectangle in Fig. 12(b)).



(a) (b)

Fig. 12. The home service robot prototype. Fig. 12(b) is the interior structure of Fig. 12(a). As in Fig. 12(b), the decoding system consists of a CMOS imaging sensor and DSP embedded system.

B. *Experiment Result*

Based on the three-layer HIC architecture proposed in Section IV, the smooth path planning algorithm is validated with the home service robot prototype. As in Fig. 13, the robot starts from point A and stops at point B. The real path and simulation path are both smooth and continuous. Due to the space constraints, the velocity and angular velocity are not presented. However, they are both continuous and bounded.

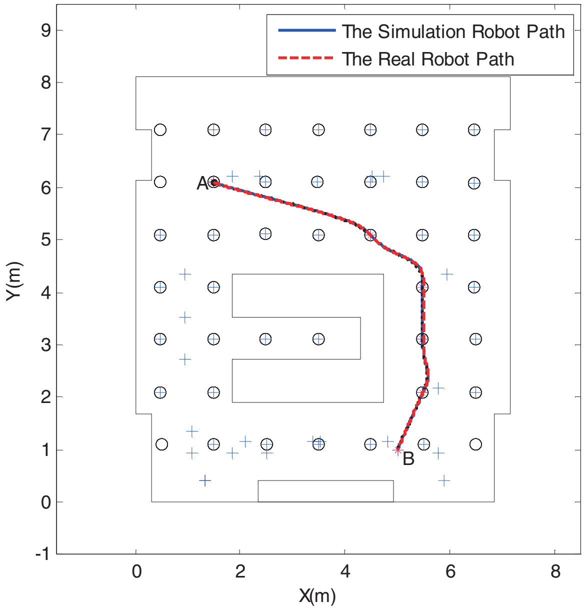


Fig. 13. The simulation and experiment results.

VI. Conclusion

This paper presents a smooth path planning algorithm for the home service robot in presence of a known map and static obstacles. By introducing the 2D code landmarks and the η^3-splines to the algorithm, a smooth path is generated under the constraint of upper-bounded curvature and upper-bounded curvature derivative. With the constraint of the feasible area, the generated path is not only smooth but also collision-free. The simulation and the experiment results validate the feasibility of the smooth path planning algorithm.

The smooth path planning algorithm proposed in this paper has achieved initial results, our future efforts will focus on the following aspects.

Firstly, the values of the curvature and the curvature derivative are not discussed in this paper. Since finding the solutions to the SQP problem is rather difficult, the curvature and the curvature derivative at the endpoints are set to zero. Therefore, the optimization of the curvature and the curvature derivative will be the first issue to work on.

Secondly, the feasible area mentioned in this paper is a static region. By means of sonar sensors or laser rangefinders, the feasible area will change dynamically according to the surroundings. The generated path can be further improved.

At last, the path planning algorithm is based on a known map and static obstacles, and the algorithm is limited in the people-aware navigation. As a consequence, planning in a dynamic and unknown environment will be our next focus.

References

[1] L. Dubins, “On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents,” *Amer. J. Math.*, vol. 79, pp. 497–517, 1957.

[2] T.Fraichard and A. Scheuer, “From Reeds and Shepp’s to continuous -curvature paths,” *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 1025–1035, Dec. 2004.

[3] J. Villagra and H. Mounier, “Obstacle-avoiding path planning for high velocity wheeled mobile robots,” *IFAC World Congress.*, Prague, Czech Republic, Jul, 2005.

[4] A. Piazzi, C. Guarino Lo Bianco, and M. Romano, “ η^3 -splines for the smooth path generation of wheeled mobile robots,” *IEEE Trans. Robotics.*, vol. 23, no. 5, pp. 1089–1095, Oct. 2007.

[5] H.C. Chang and J.S. Liu, “High-quality path planning for autonomous mobile robots with η^3 -splines and parallel genetic algorithms,” in *Proc. 2008 IEEE Int. Conf. on Robotics and Biomimetics*, Bangkok, Thailand, Feb. 2009, pp. 1671-1677.

[6] C. Guarino Lo Bianco and O. Gerelli, “Generation of paths with minimum curvature derivative with η^3 -splines,” *IEEE Trans. Automation Science and Engineering.* , vol. 7, no. 2, pp. 249-256, Apr. 2010.

[7] G. Lini, A. Piazzi, and L. Consolini, “Multi-optimization of η^3 -splines for autonomous parking ,” in *Proc. 50th IEEE Conf. Decision Control/Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 6367-6372.

[8] F. Ghilardelli, G. Lini, and A. Piazzi, “Path generation using η^4 -splines for a truck and trailer vehicle,” *IEEE Trans. Automation Science and Engineering.* , vol. 11, no. 1, pp. 187-203, Jan. 2014.

[9] M.K. Habib and H. Asama. “Efficient method to generate collision free paths for autonomous mobile robot based on new free space structuring approach,” in *Proc. IEEE/RSJ International Workshop on Robots and Systems*, pp. 563-567, 1991.

[10] C. Guarino Lo Bianco and A. Piazzi, “Optimal trajectory planning with quantic G^2 -splines,” in *Proc. IEEE Intelligent Vehicles Sympoium.*, Dearborn, MI, USA, Oct. 2000, pp. 620–625.