Variable-resolution State Roadmap Generation Considering Safety Constraints for Car-like Robot

Jingyu Xiang, Yuichi Tazaki, Tatsuya Suzuki and B. Levedahl
Department of Mechanical Science and Engineering
Graduate School of Engineering, Nagoya University
Nagoya, Aichi, Japan
{k_kou, tazaki, t_su]zuki, blaine}@nuem.nagoya-u.ac.jp

Abstract—This research develops a new graph-map method for autonomous car-like mobile robots based on variable-resolution division of space. Unlike conventional roadmaps, which include position information only, the proposed graph-map also includes orientation information of the car-like robot. In this manner, the robot is able to plan a path in detail. The orientation information of each node is not a fixed value but a range. The range is constructed by dividing the orientation space using variable-resolution, which is obtained from the surrounding situation of links that connect to the node. Finally, the proposed method is evaluated through simulations.

Index Terms—Kinodynamic motion planning, Kinodynamic roadmap, Roadmap, Variable resolution, Car-like robot

I. INTRODUCTION

In the path-planning of mobile robots, roadmap methods, which express the available free space in the configuration space by a graph structure and use graph search methods to find a path to a specific destination, are frequently used[1][2][3]. In the roadmap methods, the workspace is abstracted to nodes, which represent positions in space, and links, which represent connections between nodes in the workspace. There is an important safety constraint that none of the links should intersect obstacles.

Roadmap methods are classified into deterministic roadmap methods and probabilistic roadmap methods. Examples of the former are Voronoi diagrams[4], Visibility roadmap[5] and others. An advantage of these methods is that a path is guaranteed to be obtained, if one exists, since a roadmap is constructed over the whole configuration space. However, if the configuration space is complicated, it means that the configuration space has high dimension or there are many obstacles, the computational complexity is high and path-planning will take time. Examples of the probabilistic methods are Rapidly-exploring Random Tree (RRT)[6] and the Probabilistic Roadmap Method (PRM) [7]. These methods handle the computational complexity efficiently. However, since the dynamics of the robot is not considered, there is no guarantee that the robot can follow the plan without violating constraints (e.g. physical, safety obstacle avoidance, etc.). There are also methods to remake the map after path-planning[8], and calculate a new trajectory by observing environment after path-planning[9].

In kinodynamic motion planning[10], the motion planning considers the robot dynamics by using a state space construction from the configuration of the robot, its velocity, and its acceleration. Planning in the state space is not as easy as planning in position space since, in general, a straight line in the state space is not a valid motion in general[11].

In the area of kinodynamic motion planning, Kinematic Motion Planning by Interior-Exterior Cell Exploration (KPIECE) was proposed by Kavraki [12]. KPIECE is a sampling-based approach, which is capable of motion planning for multiple systems with complex dynamics, and can only be probabilistically complete.

Based on the above background, this research develops a new roadmap generation method for a car-like robot (Fig. 1). The method considers a 3D state space which is constructed from the 2D position of the robot and a 1D orientation space. The map which only considers the position space of a robot is called the Position Roadmap (P-roadmap). The map which considers the state space of a robot is called the Variable-resolution Orientation Roadmap (VOR). A position roadmap in 2D is generated by existing methods, and is extended to the 3D state roadmap by adding orientation information. In contrast to position, orientation information is expressed in the orientation space not by points, but by ranges in variable resolutions. A node in the proposed roadmap is a pair of a configuration point, and a region of orientation space. A pairs of nodes is connected by a directed link if a transition from one node to the other node is possible. The former node is the tail and the latter is the head of the directed link. Using this directed link, the robot can determine whether a path, which is obtained by path-planning, can be physically followed.

The rest of this paper is organized as follows: Section II
presents the car-like mobile robot, which is considered in this study. Section III presents the proposed roadmap generation method. Section IV develops the constraint conditions of the mobile robot. Section V demonstrates a proposed VOR method using a numerical example. Finally, concluding remarks are given in Section VI.

Notation: A series \( \{p_0, p_1, \ldots, p_n\} \) is written as \( \{p_i\}_{0:n} \).

## II. System Description

In this paper, a car-like robot is assumed to move in the 2D workspace. The 3D configuration space then consists of positions in the real space, which is represented by \( p \in P \subset \mathbb{R}^2 \), and an orientation, which is represented by \( \theta \in \Theta \subset \mathbb{R} \). The state of the robot in 2D space is represented by \( v \in V \subset \mathbb{R}^2 \), which consists of a velocity \( v \) and an angular velocity \( \omega \). The final state of the robot is then denoted by \( x = [p, \theta] \in X, \ X = P \times \Theta \times V \).

This remainder of this section is organized as follows. Section II-A presents some terms and variables. A control transform: the local coordinate is then obtained from the following coordinate

\[ \begin{align*}
\mathbf{v}^x &= \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \mathbf{v}^X, \\
\mathbf{v}^y &= \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \mathbf{v}^X, \\
\theta &= \theta - \phi, \\
\phi &= \arctan \left( \frac{p^y - \hat{p}^y}{p^x - \hat{p}^x} \right).
\end{align*} \]

where \( (\hat{v}^X, \hat{v}^Y) \) is the velocity in the global coordinate, and \( (v^x, v^y) \) is the velocity in the local coordinate.

If the robot is driven by constant velocity and constant angular acceleration inputs, as shown in Fig. 3, the dynamics of the robot is given by:

\[ \begin{align*}
\frac{d}{dt} & \begin{pmatrix} x(t) \\ y(t) \\ \theta(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \\ \theta(t) \end{pmatrix} + \begin{pmatrix} \cos \theta & 0 & 0 \\ 0 & \sin \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} u(t),
\end{align*} \]

where \( p(t) = [x(t), y(t)]^T \) is the position of the robot at time \( t \), and \( u(t) = [u_v(t), u_\alpha(t)]^T \) is the input which is the set of velocity and angular acceleration at \( t \). Notice that angular velocity \( \omega \) and the steering angular \( \xi \) have the following condition:

\[ \omega = \tan \xi / l. \]

In this paper, the angular velocity is assumed to be linearly proportional to the steer angle, and robot is assumed to move with constant angular acceleration \( \alpha \) and constant velocity \( v \), which gives

\[ \begin{align*}
\begin{pmatrix} x(t) \\ y(t) \\ \theta(t) \end{pmatrix} &= \begin{pmatrix} e^t \int_0^t \cos \left( \frac{\alpha}{2}s^2 \right) ds \\ e^t \int_0^t \sin \left( \frac{\alpha}{2}s^2 \right) ds \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} v \sqrt{\frac{\alpha}{2}} C\left( \sqrt{\alpha/\pi} t \right) \\ v \sqrt{\frac{\alpha}{2}} S\left( \sqrt{\alpha/\pi} t \right) \\ \alpha t^2 / 2 \end{pmatrix},
\end{align*} \]

where \( C(*) \) and \( S(*) \) are Fresnel integrals, and the resulting curve is called a Clothoid curve. A control law that drives the robot between two \( (x, y, \theta) \) conditions can be defined using a combination of clothoid curves (Fig. 4). Two cases, \( \theta > 0, \theta' < 0 \) and \( [\theta > 0, \theta' \geq 0] \) are explained here. In both cases, following boundary conditions can be specified:

\[ p(0) = (0, 0), \quad p(T) = (d(p, p'), 0), \quad \theta(0) = \theta, \quad \theta(T) = \theta', \]

where \( \theta(t) \) is the orientation of the robot at \( t \), and \( T \in \mathbb{R}^+ \) is the time between the start and end points.

1) case 1: \( \theta > 0 \) and \( \theta' < 0 \): In this case as shown in Fig. 4(a), \( \hat{p} \) is the intersection point of line \( l \), whose slope is \( \tan \theta \) at \( p \), and line \( l' \), whose slope is \( \tan \theta' \) at \( p' \). Also, as shown in the figure, \( \gamma = \pi - |\theta - \theta'| \) and \( L \) is the bisector of the angle \( \gamma \). The solid curve is a combination of two clothoid curves with \( C \) and \( C' \) divided by \( L \). \( C \) is defined by:

\[ \begin{align*}
\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} &= \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} C_x(t) \\ C_y(t) \end{pmatrix}, \\
C_x(t) &= \frac{\pi}{\alpha} \left( C\left( \sqrt{\alpha/\pi} t \right) \right), \\
C_y(t) &= \frac{\pi}{\alpha} \left( S\left( \sqrt{\alpha/\pi} t \right) \right),
\end{align*} \]

where \( \alpha \) is the angular acceleration at \( C \). Similarly, \( C' \) is defined by:

\[ \begin{align*}
\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} &= \begin{pmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{pmatrix} \begin{pmatrix} C'_x(t) \\ C'_y(t) \end{pmatrix}, \\
C'_x(t) &= -\frac{\pi}{\alpha} \left( C\left( \sqrt{\alpha/\pi} t \right) \right), \\
C'_y(t) &= -\frac{\pi}{\alpha} \left( S\left( \sqrt{\alpha/\pi} t \right) \right) + p'_e.
\end{align*} \]
Let \( T' \in \mathbb{R}^+ \) be the transit time from \( p \) to \( p' \), and \( T_c \in \mathbb{R}^+ \) be the transit time from \( p \) to \( A \). The trajectory of the robot is then:

\[
\begin{align*}
\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} C_x(t) \\ C_y(t) \end{bmatrix}, \\
C_x(t) &= v \sqrt{\frac{\pi}{\alpha}} \left( C\left(\sqrt{\frac{\alpha t}{\pi}}\right) - S\left(\sqrt{\frac{\alpha t}{\pi}}\right) \right), \quad \text{if } t \in [0, T_c), \\
C_y(t) &= -v \sqrt{\frac{\pi}{\alpha}} \left( S\left(\sqrt{\frac{\alpha t}{\pi}}\right) + C\left(\sqrt{\frac{\alpha t}{\pi}}\right) \right), \\
C'(x(t)) &= v' \sqrt{\frac{\pi}{\alpha'}} \left( C\left(\sqrt{\frac{\alpha' t}{\pi}}\right) + S\left(\sqrt{\frac{\alpha' t}{\pi}}\right) \right) + p', \\
C'(y(t)) &= -v' \sqrt{\frac{\pi}{\alpha'}} \left( S\left(\sqrt{\frac{\alpha' t}{\pi}}\right) - C\left(\sqrt{\frac{\alpha' t}{\pi}}\right) \right) + p',
\end{align*}
\]

The orientation \( \theta(t) \) and the angular velocity \( \omega(t) \) of the robot must also be continuous at \( A \), which is given by the condition:

\[
\theta(T_c) = \frac{\theta - \theta'}, (10a) \quad |\omega(T_c)| = |\alpha - \alpha'| (T' - T_c). (10b)
\]

The angular velocities should also be \( 0 \) at \( p \) and \( p' \), because \( p \) and \( p' \) are the origins of clothoid curves.

2) case 2: \( \theta > 0 \) and \( \theta' \geq 0 \): In this case, as shown in Fig. 4(b), a new line \( l'' \) whose slope is \( \tan\left(-\left(\theta + \theta'\right)/2\right) \) is constructed at a midpoint between points \( p \) and \( p' \). Let \( \tilde{p} \) be the point of intersection \( l'' \) and \( l \) and let \( \tilde{p}' \) be the point of intersection of \( l'' \) and \( l' \). Also, let \( p'' \) be the midpoint between \( p_{\tilde{p}} \) and \( p'_{\tilde{p}} \), where \( p_{\tilde{p}} \) is the point of intersection of line \( l'' \) and a line that passes through \( p \) and is perpendicular to the bisector of \( \angle p\tilde{p}p' \) (\( p_{\tilde{p}} \) is similarly defined for \( \tilde{p}' \)). Two trajectories, one between \( p \) and \( p'' \) and one between \( p'' \) and \( p' \), can be made using the method in Section II-B1. By defining the orientation \( \theta(t) \) and angular velocity \( \omega(t) \) of the robot to be continuous at \( A \) and \( A' \), the orientation \( \theta(t) = -\left(\theta + \theta'\right)/2 \) and the angular velocity \( \omega(t) = 0 \) at \( p'' \) can be obtained.

The above results in \( u(t) \) and \( T \) being uniquely defined when \( p, \theta, p' \) and \( \theta' \) are specified and \( p\) being uniquely defined when \( u(t) \) is specified. Let \((u, p, \theta, \omega, T) = F(p, \theta, p', \theta') \) be the resulting function.

C. Constraint Condition for Movement

In this section, the local coordinate is also used for explanation. Two constraint conditions are considered during robot movement: a safety constraint, which is for collision safety, and a specification constraint, which arises from the physical limitations of the robot.

As shown in Fig. 2(b), there are two safety constraints given to the robot. The first is a constraint in the perpendicular direction to the line of movement. The second is a constraint in the direction of movement. In the former, to avoid obstacles, the robot should not leave the line of movement more than \( w(p, p') \). In the latter, the robot should not depart from the ends of the line segment by more than \( \epsilon_1(p, p') \) and \( \epsilon_2(p, p') \). These result in the following safety constraints:

\[
\begin{align*}
|y(t)| &\leq w(p, p'), \quad \forall t \in [0, T], (11a) \\
p^z - \epsilon_1(p, p') &\leq x(t) \leq p^z + \epsilon_2(p, p'), \quad \forall t \in [0, T], (11b)
\end{align*}
\]

All robots have an upper limit of velocity and acceleration. Here the limits are considered specification constraints. Let \( V_{\text{max}} \) be the upper limit of translation velocity, \( \Omega_{\text{max}} \) be the upper limit of angular velocity, and \( A_{\text{max}} \) be the upper limit of angular acceleration. The specification constraints are then:

\[
\begin{align*}
|v(t)| \leq V_{\text{max}}, \quad \forall t \in [0, T], (12a) \\
|\omega(t)| \leq \Omega_{\text{max}}, \quad \forall t \in [0, T], (12b) \\
|\alpha(t)| \leq A_{\text{max}}, \quad \forall t \in [0, T]. (12c)
\end{align*}
\]

Using the clothoid curves specified in Section II-B, the trajectory is uniquely identified when the positions and velocities of a starting point and ending point are defined.
Imposing safety and specification constraints, we can define the \textit{feasibility} for a pair of starting and ending points as:

\textbf{Definition 1 (Feasibility)}

A pair of \((p, \theta)\) and \((p', \theta')\) is feasible if and only if the trajectory \((u, p, \theta, T) = F(p, \theta, p', \theta')\) between the pair satisfies constraint conditions \((11)\) and \((12)\). A series \(\{(p_0, \theta_0), (p_1, \theta_1), \ldots, (p_n, \theta_n)\}\) is feasible if and only if \((p_i, \theta_i), (p_{i+1}, \theta_{i+1}), (i = 0, 1, \ldots, n - 1)\) is feasible.

\textbf{Definition 2 (Feasible Region)}

Given \(p, p'\), and \(F(p, p')\), which is the set of \((\theta, \theta')\) such that the pair of \((p, \theta)\) and \((p', \theta')\) is feasible, then \(F(p, p')\) is called a \textit{Feasible Region}.

If \(F(p, p') = \emptyset\), a transition from \(p\) to \(p'\) is impossible using the proposed controller.

\section{III. Orientation Roadmap Generation Method}

This section explains the proposed method to find feasible trajectories. \textbf{Figure 5(a)} shows a path from \(p_1\) to \(p_4\). The robot should select a valid orientation at each position to track the path. If the orientation sequence is not valid, however, the robot is unable to reach \(p_4\) from \(p_1\) as depicted by red dotted lines in \textbf{Fig. 5(a)}. To determine if the orientation sequence is valid, the orientation information can be added to \textbf{Fig. 5(a)}, as shown in \textbf{Fig. 5(b)} and \textbf{5(c)}.

\textbf{A. Augmentation of Orientation Information}

This section discusses how to extend the P-roadmap \(M = (P, L)\) to a VOR \(M_o = (X_o, L_o)\) using the orientation information, where \(X_o\) is the set of nodes and \(x = (p, \theta) \in X_o\) is a pair of a position \(p \in P\) and a orientation region \(\bar{\theta} \subseteq \Theta\), and \(L_o \subseteq [X_o]^2\) is the set of directed links. Let \(O(p) = \{ \theta | (p, \theta) \in X_o\}\) be the set of orientation regions at \(p\), \(\Pi[M_o] = \{ \{(p_i, \theta_i)\}_{0:n} | n \in \mathbb{N}, x_i \in X_o, (x_i, x_{i+1}) \in L_o\}\) be the set of paths in \(M_o\), and \(\Pi[M_o](p, p') = \{ (x_i)_{0:n} \in \Pi[M_o] | p_0 = p, p_n = p'\}\) be the set of paths from \(p\) to \(p'\) in \(M_o\). The connectivity of the nodes can be defined as:

\textbf{Definition 3 (Connectivity of State Nodes)}

The following relation holds for \(x = (p, \theta)\) and \(x' = (p', \theta')\) in \(M_o = (X_o, L_o)\):

\[(x, x') \in L_o \iff (p, p') \in L\] and \((\theta, \theta') \subseteq F(p, p')\).

This means, if \((p, \theta), (p', \theta')\) for \(\forall \theta \in \bar{\theta}, \forall \theta' \in \bar{\theta}'\) is feasible, a directed link from \(x\) to \(x'\) is constructed.

From this definition, the following lemma stated:

\textbf{Lemma 1 For an arbitrary path} \(\{x_i\}_{0:n} \in \Pi[M_o]\), \(x_i = (p_i, \theta_i)\) in \(M_o\), a series \(\{(p_0, \theta_0), (p_1, \theta_1), \ldots, (p_n, \theta_n)\}\), \((\theta_i \in \bar{\theta}, \forall i = 0, 1, \ldots, n - 1)\) is feasible.

That is to say that an arbitrary path in \(M_o\) gives a feasible trajectory.

\textbf{B. Checking feasibility conditions}

This section addresses how to check whether \(\bar{\theta} \times \bar{\theta}' \subseteq F(p, p')\) in generally. As shown in \textbf{Fig. 6}, because in general \(F(p, p')\) is not a convex region, it is difficult to check
In general, there is no prior knowledge about an appropriate global coordinate to divide the orientation space.

This section addresses how to check the constraint conditions. In this section, how to check the constraint conditions only need to be checked if \( \tilde{\theta} \) is a leaf of the tree. Then

Algorithm 1: generate links \((p, p')\)

1: for all \((\tilde{\theta}, \tilde{\theta}') \in \mathcal{O}(p) \times \mathcal{O}(p')\) do
2: \(\mathcal{V}(\tilde{\theta}, \tilde{\theta}') \leftarrow \) set of vertices of \(\tilde{\theta} \times \tilde{\theta}'\)
3: if \(\mathcal{V}(\tilde{\theta}, \tilde{\theta}') \subset \tilde{\mathcal{F}}(p, p')\) then
4: make directed link between \((p, \tilde{\theta})\) and \((p', \tilde{\theta}')\)
5: else if \((\mathcal{V}(\tilde{\theta}, \tilde{\theta}') \cap \tilde{\mathcal{F}}(p, p')) \neq \emptyset\) then
6: if \(\tilde{\theta}\) is a leaf of tree, then
7: \(C = C \cup \{\tilde{\theta}\}\)
8: end if
9: if \(\tilde{\theta}'\) is a leaf of tree, then
10: \(C = C \cup \{\tilde{\theta}'\}\)
11: end if
12: end if
13: end for

resolution. If dividing is too coarse, there may be no path to a destination shown in Fig. 5(b). If partitioning is too fine, there may be many directed links and it will make path-planning computationally expensive. This paper proposes a partitioning method based on variable-resolution. In the proposed method, the variable resolution partitioning of the orientation space is expressed by a binary tree. An example of partitioning at \(p \in \mathcal{P}\) is shown in Fig. 7(a), and an example of partitioning the feasible orientations at an ordered pair of \(p\) and \(p' \in \mathcal{P}\) is shown in Fig. 7(b).

A flow diagram of this partitioning method is shown in Fig. 8. Initially, the orientation space at each \(p\) is divided into quarters (Resolution 1). Next, generate_links\((p, p')\), which is given in Algorithm 1, is performed on all \((p, p') \in \mathcal{L}\). In Algorithm 1, \(C\) is the set that contains the orientation regions. Lines 3-12 of Algorithm 1, determine if each orientation region should be further divided. If the orientation regions are further divided, \(\tilde{\theta} \in \mathcal{V}(p)\) is stored in \(C\). The criteria for dividing a direction region is based on the relationship between vertices in the orientation region \(\mathcal{V}(\tilde{\theta}, \tilde{\theta}')\) and \(\tilde{\mathcal{F}}(p, p')\) as shown in Fig. 7(b). In Fig. 7(b), the blue ellipse depicts \(\tilde{\mathcal{F}}(p, p')\) and, because a part of \(\mathcal{V}(\tilde{\theta}_3, \tilde{\theta}_1')\) is the subset of \(\tilde{\mathcal{F}}(p, p')\), \(\tilde{\theta}_3\) and \(\tilde{\theta}_1'\) are stored into \(C\). After checking for all regions, \(\tilde{\theta}_2\), \(\tilde{\theta}_3\), \(\tilde{\theta}_4\), \(\tilde{\theta}_1'\), \(\tilde{\theta}_2'\), \(\tilde{\theta}_3'\) are divided, respectively. Based on this process, the orientation space is automatically divided into suitable size regions according to the situation.

IV. APPLICATION TO A ROBOT

In this section, the proposed method is applied to the robot which was described in Section II. The variables used were introduced in Fig. 2 and Section II-A. Finding the constraint conditions of the robot is separated to 3 parts, position, angular velocity and angular acceleration. Since the trajectory is a combination of two clothoid curves, the angular acceleration and the velocity can be determined from the constraint conditions of the trajectory. Therefore, only the constraint condition on the y-axis position is explained here.

Considering (11a), which is a y-axis constraint condition (Fig. 9), let \(\theta(p, p')\) be the angle from \(p\) to \(p'\) and \(p_{y_{\text{max}}}\) be the most distant point in the trajectory. Because of symmetry around the x-axis, the following 2 conditions can be demonstrated:
Fig. 9 A region about constraint condition of y-axis

1) \( \theta_A \geq \theta(p,p') \) and \( \theta_A \geq 0 \): Since the trajectory is a clothoid, the following is obtained:

\[
p_{y_{\text{max}}} = v_t \sqrt{\frac{\pi}{\alpha}} \left( \cos \theta' - \sin \theta' \right) \times \\
\left(-C \sqrt{\alpha/\pi t_{\text{max}}} \right) + \left(-S \sqrt{\alpha/\pi t_{\text{max}}} \right) + p_c, \quad (13)
\]

\[
\frac{1}{2} \alpha' t^2_{\text{max}} \leq \left| \frac{1}{2} \alpha' t^2 - \theta_A \right| = |\theta|.
\]

2) \( \theta_A \geq \theta(p,p') \) and \( \theta_A < 0 \): Similarly to the above, the following is obtained:

\[
p_{y_{\text{max}}} = v_t \sqrt{\frac{\pi}{\alpha}} \left( \cos \theta - \sin \theta \right) \times \\
\left(C \sqrt{\alpha/\pi t_{\text{max}}} \right) + \left(-S \sqrt{\alpha/\pi t_{\text{max}}} \right) + p_c, \quad (15)
\]

\[
\frac{1}{2} \alpha t^2_{\text{max}} \leq \left| \frac{1}{2} \alpha t^2 - \theta_A \right| = |\theta|.
\]

V. NUMERICAL SIMULATION

Considering the system defined in Section IV the experimental result shown in Fig. 10 can be obtained. Figure 10(a) depicts a workspace and a P-roadmap. Fig. 10(b) depicts a VOR for the P-roadmap and a trajectory. In Fig. 10(a), the black circles depict each position, the line segments depict links, and the green areas depict obstacles. In Fig. 10(b), the red balls depict each position, the blue curve denotes a calculated trajectory, and the green fans show regions which were selected at each position. In this experiment, the initial state of the robot is \((X, Y, \theta) = (0, 0, 0)\) and the destination is \((X_G, Y_G) = (500, 0)\). The orientation of the robot at the destination was not specified. In the proposed method, since the trajectory is calculated using clothoid curves, the inputs are decided at the path-planning phase.

Fig. 10 Environment for experiment and map constructed by the proposed method

VI. CONCLUSION

In this paper, a new roadmap generation method based on variable-resolution for an autonomous car-like robot has been presented. Since the map generated by the proposed method includes orientation information of the robot, the inputs, which consider the safety and the physical limitations of the robot, can be found simultaneously during the path-planning phase. The method has been tested through numerical simulation and an example was presented. In the future, the algorithm will be tested through experiments in a real environment, and extended to multi-robot systems.

REFERENCES