World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:6, No:11, 2012

International Science Index, Computer and Information Engineering Vol:6, No:11, 2012 waset.org/Publication/15391

Sensor-based Motion Planning for a Car-like Robot based on Bug Family Algorithms

Dong-Hyung Kim, Ji Yeong Lee, and Chang-Soo Han

Abstract**—**This paper presents a sensor-based motion planning algorithm for 3-DOF car-like robots with a nonholonomic constraint. Similar to the classic Bug family algorithms, the proposed algorithm enables the car-like robot to navigate in a completely unknown environment using only the range sensor information. The car-like robot uses the local range sensor view to determine the local path so that it moves towards the goal. To guarantee that the robot can approach the goal, the two modes of motion are repeated, termed motion-to-goal and wall-following. The motion-to-goal behavior lets the robot directly move toward the goal, and the wall-following behavior makes the robot circumnavigate the obstacle boundary until it meets the leaving condition. For each behavior, the nonholonomic motion for the car-like robot is planned in terms of the instantaneous turning radius. The proposed algorithm is implemented to the real robot and the experimental results show the performance of proposed algorithm.

Keywords**—**Motion planning, car-like robot, bug algorithm, autonomous motion planning, nonholonomic constraint.

I. Introduction

AUTONOMOUS navigation for car-like robots is a challenging field because the car-like robot must depend on only the local information from the sensor during the motion planning. Thus the navigation algorithm needs to guarantee that the robot approaches the goal location only using the local information of the environment. Another difficulty for navigation occurs because of the motion constraints, like the nonholonomic constraint that prevents a side-slip motion in the perpendicular direction, and the minimum turning radius. In order to design a motion planner for the car-like robot, the nonholonomic motion generated from the motion planning algorithm should satisfy these motion constraints.

This paper proposes a sensor-based motion planning algorithm for the car-like robot based on the Bug family algorithms. For the point robot with a contact sensor or a zero range sensor, the simplest motion planning algorithms called Bug1, Bug2 algorithms was introduced [1], whereas the Tangent Bug algorithm is used when the robot has a finite range sensor with a 360° infinite orientation resolution [2]. Basically, the two behaviors, the motion-to-goal and the wall-following, are repeated until the robot reaches the goal or a loop is detected, in which case the goal is not reachable. The motion-to-goal moves the robot in the direction of the shortest path to the goal. As the robot avoids the obstacle, the distance to the goal may reach a local minimum and begin to increase. If this happens, the motion-to-goal is terminated and the wall-following is executed so that the robot circumnavigates the obstacle boundary until the distance to the goal becomes smaller than the recorded minimum distance to the goal. These works have been applied to various motion planning tasks [3],[6]. Extensions to classical Bug-like algorithms are introduced in [3]–[5]; the problem of the planetary rover navigation with a limited field of view (FOV) is addressed by the WedgeBug algorithm. This motion planning algorithm minimizes the requirement to sense and store data using autonomous gaze control [3]. As a variant of TangentBug, CautiousBug was introduced in [4]. Although the structure of the algorithm is the same as Bug-like algorithm, CautiousBug algorithm uses a cautious search when it determines the direction to follow on obstacle boundaries. For on-line robot navigation, CBug algorithm was designed for the navigation of disk robot and its competitiveness in terms of path stability has been established [5]. However, the above works consider the navigation of robot without accounting for motion constraints. The minimum turning radius of car-like robot limits the instantaneous reachable space; thus, the convergence to goal point for the motion planning is not guaranteed.

Compared to the two degree-of-freedom point robot, the car-like robot is modeled with three degree-of-freedom, including not only position but also orientation, and it also has a nonholonomic constraint. A number of approaches have been proposed recently to cope with this motion constraint for the car-like robot navigation. The mapping from the configuration space to the specific configuration space was suggested so that the well-known reactive collision avoidance method for a holonomic robot like a potential field can be applied to the nonholonomic robot in that space [7]. Thus, the transformation between two spaces is necessary for executing the reactive motion planning method. Nonholonomic path deformation methods were introduced in [8] which simultaneously execute the real-time path planning and obstacle avoidance. Given an initial path, the methods in [8] define the potential field over the space of a path. The obstacle potential is generated from the ob-

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the 'Advanced Robot Manipulation Research Center' support program supervised by the NIPA(National IT Industry Promotion Agency), (grant number NIPA-2012-H1502-12-1002); and also supported by the Ministry of Knowledge Economy, Korea, under the Industrial Foundation Technology Development Program supervised by the Korea Evaluation Institute of Industrial Technology (Grant number 10038660, Development of the control technology with sensor fusion based recognition for dual-arm work and the technology of manufacturing process with multi-robot cooperation).

D. Kim and J. Lee are with the Department of Mechanical Engineering, Hanyang University Seoul campus, Seoul, Korea (e-mail: fileman@hanyang.ac.kr; jiyeongl@hanyang.ac.kr).

C. Han is with the Department of Mechanical Engineering, Hanyang University ERICA campus, Gyeonggi-do, Korea (phone: +82-31-400-5247; fax: +82-31-406-6398; e-mail: cshan@hanyang.ac.kr).

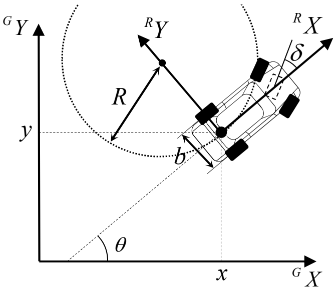

Fig. 1 Kinematic model of a car-like robot

stacle and determines the direction of the path deformation. Since these methods compute the obstacle potential field along the path, it requires numerous computations by the robot's real-time controller. These reactive approaches also suffer from local minima which causes the robot to converge to a non-goal region.

In this paper, we focus on the motion planning algorithm which provides a nonholonomic path to a goal for a car-like robot that ensures the global convergence using only locally available sensor information. The proposed motion planning algorithm generates the local nonholonomic motion using the instantaneous turning radius which satisfies the motion constraints on the robot. That is, even though the notions of the motion-to-goal and the wall-following from the Bug family algorithms are analogously used, we provide the local feasible motion to the car-like robot.

II. Sensor-based Motion Planning Algorithm

The proposed algorithm navigates the car-like robot in a planar unknown environment populated by stationary obstacles. The car-like robot is a point robot with motion constraints, a nonholonomic constraint and a minimum turning radius. Only forward motion is allowed during the motion planning. We also assume that the robot is equipped with a 2D range sensor.

Let us briefly describe the sensor-based motion planning algorithm. The proposed motion planning algorithm uses the two basic behaviors, the motion-to-goal and the wall-following. For each behavior, the instantaneous nonholonomic motion is generated based on the Pure pursuit method [11],[12] so that the motion constraints of the robot are satisfied.

During the motion-to-goal behavior, the robot moves toward the goal point by following the nonholonomic path. When the distance to the goal point begins to increase during the motion-to-goal behavior, the local minima is determined and the behavior switches to the wall-following behavior. The robot circumnavigates the obstacle boundary until the current distance to the goal becomes smaller than the recorded minimal distance to the goal point. Then the robot leaves the obstacle boundary by switching the behavior to the motion-to-goal.

A. Basic Notions

1) Kinematic model

In Fig. 1, the global coordinate system is a fixed coordinate system, and each axis is denoted by the superscript, *G*. The robot coordinate system, denoted by the superscript *R*, is a moving coordinate system attached to the robot, with the x-axis aligned with the heading of the robot. If the position and the orientation of the car-like robot are expressed by (x, y) and θ, respectively, the configuration of the car-like robot can be defined as $\mathbf{q}=[x\ y\ \theta]^T$. Therefore, the kinematic model of the car-like robot is expressed by

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \tag{1}$$

where $\mathbf{u}=[v\ \omega]^T$ is the control input vector that contains the linear velocity v and the angular velocity ω. Here, the backward motion of the car-like robot is not considered in this work: that is, v is positive.

In this kinematic model, the motion of the car-like robot constrained by the following nonholonomic constraint is describes as:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0. \tag{2}$$

That is, it is assumed that the car-like robot is reasonably slow such that the longitudinal traction and lateral force exerted on the tires do not exceed the maximum static friction between the tires and the floor. This is called a no-slip condition. At the low speed of the car-like robot, the kinematic steering is determined from the Ackerman turning geometry as $\delta=b/R=b\kappa$ where δ is the equivalent steering angle for the front wheels, b is the wheelbase, and R is the turning radius. Thus with the maximum steering angle for the front wheel $\delta_{\max}$, the lower limit on the turning radius is represented by $R_{\lim}$ which satisfies $\delta_{\max}=b/R_{\lim}$.

2) Local Information from the 2D Laser Rangefinder

The maximum sensing range for 2D laser rangefinder is defined as $r_{\max}$ and the FOV of the laser rangefinder is also limited by 2β. Even if the laser rangefinder provides the set of the discrete points with respect to the angle resolution of the sensor, they are considered as a continuous line or curve if the distance between the two points is close. Let the set of the end points of the obstacle boundary be the set E. For example, in Fig. 2, $E=\{e_1, e_2, e_3, e_4\}$.

Let $A(\mathbf{q}, r_{\max}, \beta)$ be the area within the sensing distance and FOV. The arc boundary of A at radius $r_{\max}$ is defined as ∂A, and B is the union of the two bounding rays. Then the local free spa-

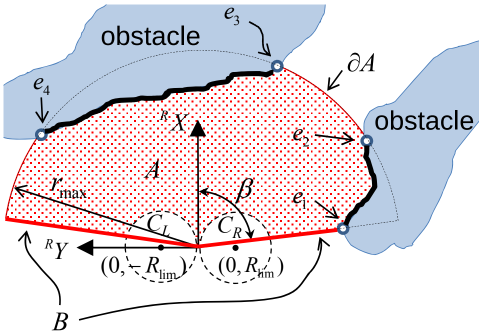

Fig. 2 Local view from 2D laser rangefinder of the robot

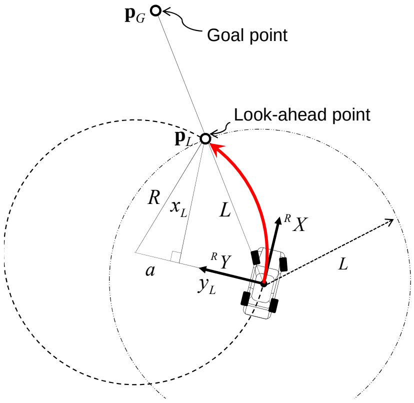

Fig. 3 Geometry of Pure pursuit method

ce is

$$F_{free} = A - \sum_{i=1}^{n} WO_i \tag{3}$$

where WO_i is an i-th obstacle and n is the number of the obstacle. Then the local feasible space is defined as

$$F_{feas} = F_{free} - (C_L \cup C_R) \tag{4}$$

where C_L and C_R are the interior of the circles centered at (0, $-R_{\text{lim}}$), (0, R_{lim}), respectively. Thus, to guarantee the collision-free motion of the car-like robot, the instantaneous motion should be planned inside of F_{feas} during motion planning.

3) Pure Pursuit Method for the Motion Planning Algorithm

For each behavior of the motion-to-goal and the wall-following, the proposed algorithm uses the Pure pursuit method [12] to plan the local nonholonomic motion. Figure 3 shows the geometry of the Pure pursuit for the car-like robot. In Fig. 3, an arc that joins the current position of the robot and the look-ahead point $\mathbf{p}_L$ can be constructed. The chord length of this arc is the look-ahead distance L. Let the location of the look-ahead point be (x_L, y_L) with respect to the robot coordinate. Then from Fig. 3, $x_L^2 + y_L^2 = L^2$ and $R = a + y_L$ can be geometrically obtained where a satisfies $x_L^2 + a^2 = R^2$. From these equations, $R=L^2/2y_L$ can be obtained. Thus with the goal point $\mathbf{p}_G$ shown in Fig. 3, the turning radius is calculated using this equation.

Figure 4 shows the set of feasible trajectory with Pure pursuit method. If the orientation of the car-like robot is exactly toward the instantaneous goal point, then the look-ahead point lies on the intersection of the X-axis of the robot coordinate and the circumference of the circle with radius L. As the orientation of the robot moves away from the instantaneous goal point, the look-ahead point moves toward the point A or B along the circumference of the circle with radius L. When the look-ahead point coincides with the point A or B, then the look-ahead point is located on the Y-axis of the robot coordinate. For this case, the pure pursuit method calculates the minimum value of the turning radius min(R) as R_{min}. Then the look-ahead distance L and the minimum turning radius have the relationship given by

$$L = 2\min(R) = 2R_{\min} \quad \text{or} \quad R_{\min} = \frac{L}{2}. \tag{5}$$

thus, the look-ahead distance L needs to be selected so that it satisfies $R_{\text{min}}>R_{\text{lim}}$ where R_{lim} is the lower limit on the turning radius given in Sec. II. *A*.1.

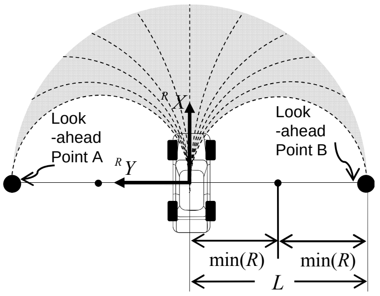

Fig. 4 Set of feasible trajectory with Pure pursuit method

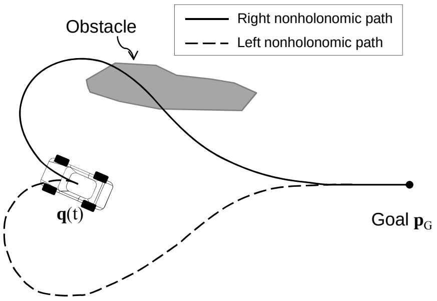

Fig. 5 Left and right nonholonomic paths.

B. Motion-to-goal Behavior

1) Blocking Obstacle with the Nonholonomic Path

The left and right nonholonomic paths are the paths to the goal point starting with the right turn and the left turn, respectively. That is, the car-like robot has the two choices shown in Fig. 5, the left nonholonomic path (dotted line) and the right nonholonomic path (solid line). These paths are obtained based on the Pure pursuit method so that it satisfies the motion constraint of the car-like robot. The car-like robot selects the one which does not intersect with the obstacle and smaller than the other. The detail of algorithm is not described due to the space limitation.

The blocking obstacle is detected when the both of path intersect with the obstacle. In the example of Fig. 5, the car-like robot can select the left nonholonomic path even the length of this path is longer than the other path. Thus the obstacle WO_1 in Fig. 5 is not the blocking obstacle. Otherwise, if there is no collision-free path, the blocking obstacle requires the termination of the direct motion-to-goal and the beginning of the slide motion-to-goal to avoid the obstacle.

2) Direct Motion-to-Goal Behavior

Let S_i be a point where the car-like robot starts the i-th direct motion-to-goal. The direct motion-to-goal behavior is required when there are no blocking obstacles and $(\mathbf{p}_G - \mathbf{p}) \cdot \mathbf{X}_R \geq 0$ at S_i.

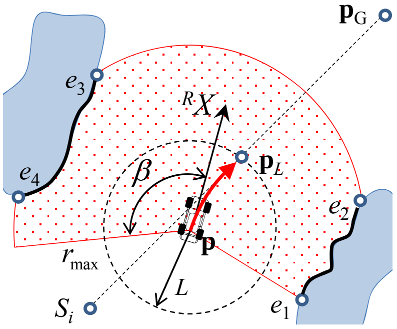

Fig. 6 Direct motion-to-goal when $(\mathbf{p}_G - \mathbf{p})\cdot\mathbf{X}_R \geq 0$ at S_i

During this behavior, as shown in Fig. 6, the look-ahead point $\mathbf{p}_L$ lies on the line between S_i and $\mathbf{p}_G$ where it satisfies $d(\mathbf{p}, \mathbf{p}_L)+d(\mathbf{p}_L, \mathbf{p}_G)=d(\mathbf{p}, \mathbf{p}_G)$. Then, to determine the instant motion of the car-like robot, the local goal point is defined as the focus point F'. Therefore, with this focus point, the instantaneous motion for the direct motion-to-goal is determined using the Pure pursuit method. The direct motion-to-goal behavior is switched to the slide motion-to-goal when the blocking obstacle is detected.

3) Slide motion-to-goal behavior

Figure 7 shows the two possible cases when $e_{min} \in \partial A$. The left and right side of Fig. 7 shows that focus point F exists on the arc boundary of obstacle B or the interior of the sensing area A, respectively. To guarantee the shortest path to the goal point, the focus point F is determined as F=e_{min} with the following equation:

$$e_{\min} = \arg\min_{e_i \in E} \left\{ d(\mathbf{p}, e_i) + d(e_i, \mathbf{p}_G) \right\}. \quad (6)$$

Let define the line connecting $\mathbf{p}$ and F, and rotate this line with the angle $2\sin^{-1}(d_{safe}/d(\mathbf{p}, e_{min}))$ to the direction where the line is not overlapped with the obstacle boundary. Then the focus point becomes F' and the look-ahead point $\mathbf{p}_L$ lies on the line between $\mathbf{p}$ and $\mathbf{p}_G$.

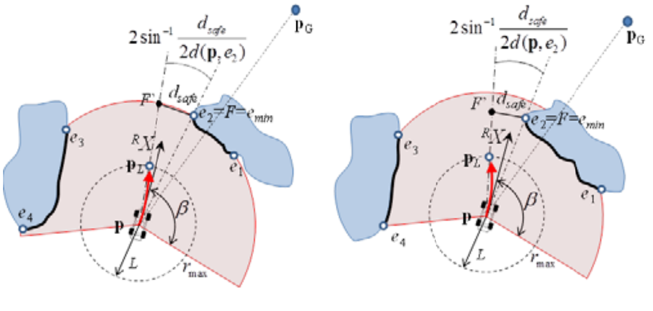

Fig. 7 Slide motion-to-goal when $F \in \partial$A(left) and $F \in$ int(A) (right)

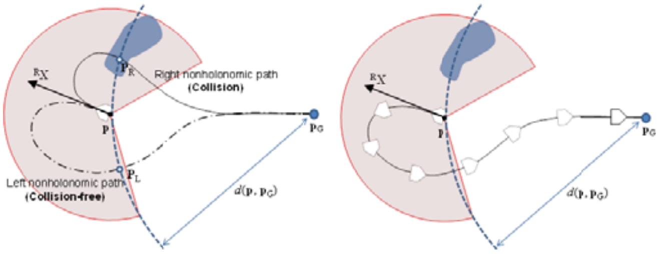

Fig. 8 Fist case: the robot can see the both paths to points $\mathrm{P_R}$, $\mathrm{P_L}$. After the nonholonomic path is determined (left), the robot follows the nonholonomic path (right)

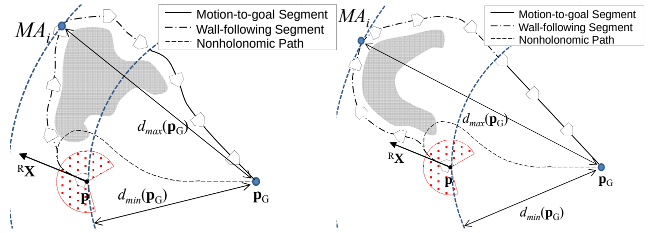

Fig. 9 Second case: the robot cannot see the both paths to points $\mathrm{P_R}$, $\mathrm{P_L}$. The some part of blocking obstacle is included within the circle radius $d_{min}(\mathbf{p}_G)$ (left) or not (right) at the switching point to the motion-to-goal behavior

The car-like robot must avoid the obstacle boundary with the fixed travel direction during the slide motion-to-goal behavior. To prevent the sudden change of focus point location, the algorithm records the previous location of F, e_i for a single obstacle boundary. Thus, the end point which is close to the previous one on the single obstacle boundary is selected. Particularly, if e_{min} lies on ∂A, $e_{min} \in \partial A$, the location of F is determined from the following equation:

$$e_{a\min} = \arg\min_{e_i \in E} \cos^{-1} \frac{{}^R X \cdot \mathbf{t}_{ei}}{\left|{}^R X\right| \left|\mathbf{t}_{ei}\right|} \quad (7)$$

where $\mathbf{t}_{ei}$ is the tangent vector at e_i, and e_i is the end point lying on the followed obstacle boundary. Because the set of obstacle point is obtained from the 2D laser rangefinder, $\mathbf{t}_{ei}$ is obtained using the two obstacle points that e_i and the obstacle point lies on the obstacle boundary and closest to e_i.

The slide motion-to-goal behavior is terminated and the direct motion-to-goal begins if the collision-free nonholonomic path to $\mathbf{p}_G$ is found. Otherwise, it is switched to the wall-following behavior if the distance to goal $d(\mathbf{p}, \mathbf{p}_G)$ begins to increase. In this case, the i-th local minimum point is determined as M_i=$\mathbf{p}(t)$ and the minimum distance to the goal is recorded as $d_{\min}$=$d(M_i, \mathbf{p}_G)$. This is the leaving condition from the motion-to-goal behavior to the wall-following behavior.

C. Temporary Nonholonomic Behavior

Temporary nonholonomic behavior is required when $(\mathbf{p}_G - \mathbf{p})\cdot{}^R X < 0$ is satisfied at the beginning of the motion-to-goal since the Euclidian distance to goal increases due to the minimum turning radius of the car-like robot. Otherwise, the robot just follows the direct motion-to-goal.

The algorithm is divided into two cases. First, the car-like robot can see both paths to points $\mathrm{P_R}$ or $\mathrm{P_L}$, where $\mathrm{P_R}$ and $\mathrm{P_L}$ are the intersection points of the left and right nonholonomic paths, respectively. If the algorithm finds the nonholonomic path to $\mathrm{P_R}$ or $\mathrm{P_L}$ as like the left side of Fig. 8, the car-like robot follows the nonholonomic path shown in the right side of Fig. 8 until $d(\mathbf{p}, \mathbf{p}_G) < d(\mathrm{P_L}, \mathbf{p}_G,)$ or $d(\mathbf{p}, \mathbf{p}_G) < d(\mathrm{P_R}, \mathbf{p}_G,)$ is satisfied. Then, the followed temporary motion behavior becomes the motion-to-goal path segment. Otherwise, if the blocking obstacle is detected so that both the left and right nonholonomic path are not collision-free, the temporary nonholonomic behavior then becomes wall-following behavior.

Second, the car-like robot cannot see the nonholonomic paths from **p** to P_R or P_L shown in Fig. 9 If the car-like robot can reach P_R or P_L without meeting an obstacle, then the followed temporary motion segment becomes the motion-to-goal segment. However, the car-like robot may meet an unexpected obstacle on the path when the car-like robot moves along the nonholonomic path. For this case, $d_{\min}(\mathbf{p}_G)$ is defined as the minimum distance to the goal during this temporary nonholonomic motion behavior. We define $d_{\max}(\mathbf{p}_G)$ as the maximum distance to goal before the car-like robot satisfies $d(\mathbf{p}, \mathbf{p}_G) < d_{\min}(\mathbf{p}_G)$. Therefore, $d_{\min}(\mathbf{p}_G) < d_{\max}(\mathbf{p}_G)$ is also satisfied. For this special case, the detail of the temporary nonholonomic motion behavior is determined as follows: the car-like robot follows the wall using the focus point *F*' for the wall-following behavior and continuously updates the distance $d_{\max}(\mathbf{p}_G)$. If the distance to goal $d(\mathbf{p}, \mathbf{p}_G)$ begins to decrease, then $d_{\max}(\mathbf{p}_G)$ is determined. Then we define the local maximum point, MA_i. While the algorithm continues the wall-following segment, the focus point *F*' for the motion-to-goal mode is used until the car-like robot reaches another local minima point or it satisfies $d(\mathbf{p}, \mathbf{p}_G) < d_{\min}(\mathbf{p}_G)$. In the case of the local minima detection, the car-like robot performs the wall-following again until it detects MA_i. The car-like robot directly move towards the goal point by switching to the motion-to-goal behavior when $d(\mathbf{p}, \mathbf{p}_G) < d_{\min}(\mathbf{p}_G)$ is satisfied.

D. Wall-following Behavior

The car-like robot continues the wall-following behavior until it satisfies the leaving condition. Due to space limitation, we only provide the brief description of algorithm. If the focus point *F'* is founded where the distance from *F'* to $\mathbf{p}_G$ is smaller than $d_{\min}(\mathbf{p}_G)$ from the 2D laser rangefinder, then the car-like robot leaves the obstacle boundary. Then the robot moves toward to $\mathbf{p}_G$ until it reaches to the leaving point Z that satisfies $d(Z, \mathbf{p}_G) < d_{\min}(\mathbf{p}_G)$. While the car-like robot approaches to the leaving point, the motion is determined with the same behavior of the motion-to-goal. When the car-like robot meets the leaving point, it resumes the motion-to-goal behavior.

III. Analysis of the Algorithm

As the motion planning algorithm uses the notions from the Bug family algorithms, the proof of completeness is analogous to the Tangent Bug or Wedge Bug algorithms. Let us remind the points S_i to be the *i*-th point where the robot starts the direct motion-to-goal, P_i the *i*-th point where the robot first senses the blocking obstacle and starts the slide motion-to-goal, M_i the *i*-th point where the local minimum point is determined (the start point of the wall-following behavior), Z_i the *i*-th point where the robot leaves the obstacle

Lemma 1 The distance to a goal decreases each wall-following segment, and between the wall-following segments.

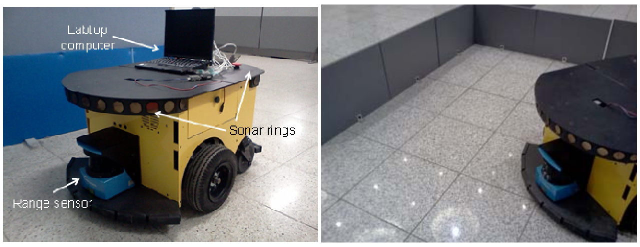

Fig. 10 Mobile robot equipped with the laptop computer, 2D laser rangefinder (left) and the environment made of the series of panel (right).

Proof During the wall-following behavior, the robot circumnavigates the obstacle boundary with a fixed direction until it reaches the leaving point where $d(\mathbf{p}, \mathbf{p}_G) < d_{\min}$ or the algorithm detects the loop. Thus, according to the switching condition, $d(Z_i, \mathbf{p}_G) < d(M_i, \mathbf{p}_G) = d_{\min}$ is guaranteed.

Lemma 2 The distance to the goal decreases between the two motion-to-goal segments.

Proof We considers the two cases for proving the convergence to goal between the two motion-to-goal segments. For the first case, if $(\mathbf{p}_G - \mathbf{p}) \cdot {}^R X \geq 0$ at S_i, the conditionally determined motion behavior is executed. In this case, the distance to goal decreases at the beginning of this behavior in this case. If the robot does not meet blocking obstacle, the convergence of Pure pursuit method guarantees that $d(\mathbf{p}, \mathbf{p}_G)$ becomes zero [11]. Otherwise, if the blocking obstacles appear during the motion-to-goal behavior, the behavior mode is switched to the wall-following behavior by determining the local minimum point M_i. This case then guarantees that $d(S_i, \mathbf{p}_G) > d(M_i, \mathbf{p}_G)$.

For the second case, if $(\mathbf{p}_G - \mathbf{p}) \cdot {}^R X < 0$ at S_i, the conditionally determined motion behavior is considered. Depending on the value of $r_{\max}$ and β for the laser rangefinder, this behavior is divided into two cases. In the first case with the large sensing range, the algorithm determines whether the robot can approach to the point P_L or P_R (Fig. 8) where it satisfies $d(\mathbf{p}, \mathbf{p}_G) < d(S_i, \mathbf{p}_G)$. Otherwise, because the wall-following starts instead of the motion-to-goal, the proof of finite length path is guaranteed from Lemma 1. In the second case with the small sensing range, the robot starts with the wall-following. During this behavior, the robot repeats the motion-to-goal and wall-following while it is determining the local maximum point MA_i and the local minimum point M_i. Thus, each path segment guarantees that the distance to goal is decreased at the end of this behavior.

IV. Experimental Results

We tested the proposed motion planning algorithm with a differential drive mobile robot named PowerBot produced by ActiveMedia Robotics (Fig. 10). The low-level control like the wheel speed control is done with the on-board controller of the robot. And the proposed motion planning algorithm works on a laptop computer onto the robot, or wireless communication can enable operation of the robot without including the laptop. For the experiments, And the environment was made of the series of panel as shown in Fig. 10, and the size of each panels is 600 mm×400 mm×20 mm.

For all experiments, the reference velocity of the robot was fixed as 0.6 m/s, with a safe distance at 0.7m. The measurable distance of the 2D laser rangefinder is about 9 m; however, we ignore the measured obstacle point out of 5 m because the robot does not need to react to obstacles far in the distance. Therefore, the maximum sensing range and angle for the 2D laser rangefinder are 5 m and 180 degrees, respectively. If the look-ahead distance for the robot is 2 m then the minimum turning radius is determined as 1 m. Although the PowerBot has a differential drive rather than a steering device, we limit the velocity commands to the right and left wheels so that this robot has a minimum turning radius like a car-like robot. As the robot's system corrects the orientation using the gyro sensor, and the long range motion planning is not considered, we rely on the onboard odometry by ignoring the small amount of localization error.

Figure 11 shows the result in the simple environment from the initial configuration $\mathbf{q}(0)=[0\ 0\ 0]^T$ to $\mathbf{p}_G=[16\text{m}\ 0]^T$. The running time for this experiment was about 36 seconds. The circles attached to the robot present the minimum size of turn. The robot's initial orientation is directed to the goal point, so the distance to the goal decreases at the beginning, as denoted in Fig. 12, where the flag equals 0 is the motion-to-goal behavior and 1 is the wall-following. Thus, we can see the only one local minima is determined.

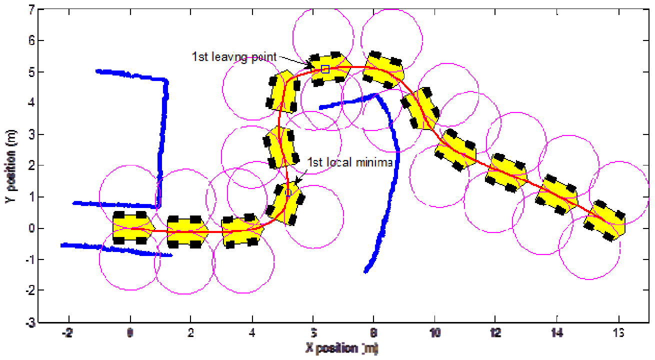

Fig. 11 Experimental results in simple environment

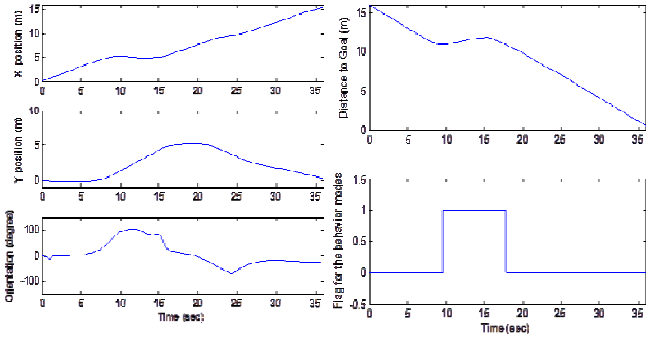

Fig. 12 Robot configuration (left), the distance to goal and the flag of behavior modes (right) in simple environment

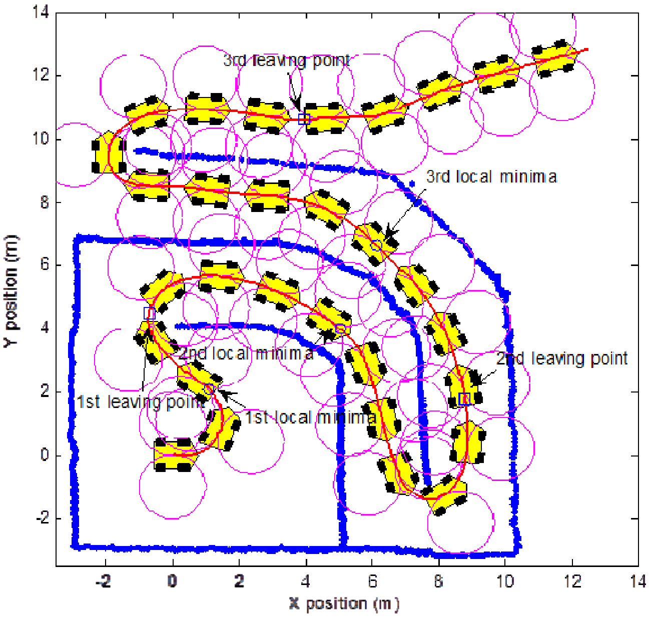

Fig. 13 Experimental results in maze-like environment

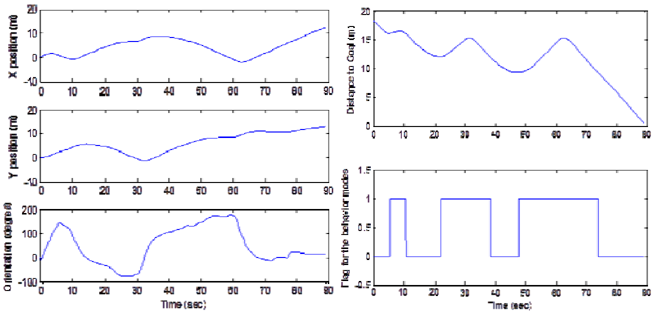

Fig. 14 Robot configuration (left), the distance to goal and the flag for behavior modes (right) in maze-like environment

We tested the motion planning algorithm with a maze-like environment. In this second experiment, the initial configuration and the goal point are given by $\mathbf{q}(0)=[0\ 0\ 0]^T$ and $\mathbf{p}_G=[13\text{m}\ 13\ \text{m}]^T$. It took about 90 seconds to reach to the goal point. Figure 13 represent the trajectory of the robot, and the graphs in Fig. 14 are the robot configuration, the distance to goal, and the status of the flag.

V. Conclusion

This paper introduces a motion planning algorithm for a car-like robot. To be more realistic, the car-like robot is equipped with a 2D laser rangefinder with a limited range and angle. The proposed motion planning method is based on the classic Bug family algorithms which are the simplest path planner for robot motion planning. Therefore, our work does not require a local path planner to obtain the nonholonomic path, thereby avoiding heavy computation requirements. For nonholonomic mobile robot motion planning, different from the previous motion planning algorithms, the proposed algorithm provides instantaneous nonholonomic motion to the car-like robot in real time. By testing the motion planning algorithm a simple environment and the complex maze-like environment, we showed that car-like robot can approach the goal point without violating the motion constraint.

REFERENCES

[1] V. J. Lumelsky and A. Stepanov, "Path planning strategies for point automaton moving amidst unknown obstacles of arbitrary shape,"*Algorithmica*,vol. 2, pp.403–430, 1987.

[2] I. Kamon, E. Rimon and E. Rivlin, "Tangentbug: A range-sensor based navigation algorithm," *Int. J. Robot. Res*., vol. 17, p.934–953, 1998.

[3] S. Laubach and J. Burdick, "An autonomous sensor-based path planner for planetary microrovers," in *Proc. IEEE ICRA*, Detroit, MI, 1999, pp. 347–354.

[4] E. Magid and E. Rivlin, "CAUTIOUSBUG: A competitive algorithm for sensory-based robot navigation," in *Proc. IEEE/RSJ Int. Conf. IROS*, Sendai, Japan, Oct. 2004, pp. 2757–2762.

[5] Y. Gabriely and E. Rimon, "CBUG: A Quadratically Competitive Mobile Robot Navigation Algorithm," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1451–1457, Dec. 2008.

[6] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "Robust Navigation in an Unknown Environment with Minimal Sensing and Representation," *IEEE Trans.on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 39, no. 1, Feb. 2009.

[7] J. Minguez and L. Montano, "Extending Collision Avoidance Methods to Consider the Vehicle Shape, Kinematics, and Dynamics of a Mobile Robot," *IEEE Trans. Robot*., vol. 25, no. 2, pp. 367–381, Apr. 2009.

[8] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive Path Deformation for Nonholonomic Mobile Robots," *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 967–977, Dec. 2004.

[9] K.Nagatani, Y. Iwai, and Y. Tanaka, "Sensor-based navigation for car-like mobile robots based on a generalized Voronoi graph," Advanced Robotics, vol. 17, no. 5, pp. 385–401, 2003.

[10] C. Lanzoni and A. Sanchez, "Sensor-based motion planning for car-like mobile robots in unknown environments," *in Proc. IEEE ICRA*, Taipei, Taiwan, Sep.2003,pp. 4258–4263.

[11] A. Ollero and G. Heredia, "Stability Analysis of Mobile Robot Path Tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., Pittsburgh, PA, 1995, vol. 3, pp. 461–466.

[12] Thomas Hellström, Ola Ringdahl, "Follow the past - a path tracking algorithm for autonomous vehicles", in *Int. J. Vehicle Autonomous Systems*, vol. 4, pp. 216–224, 2006.