

Efficient Path Planning with Limit Cycle Avoidance for Mobile Robot Navigation

Yu Lun Ting

tobyting@gmail.com

Jing Shi Liu

liu@iis.sinica.edu.tw

Abstract—Autonomous navigation in an unknown environment may encounter the limit cycle problem causing lower efficiency when a mobile robot attempts to reach the goal while avoiding obstacles on the way. To cover a wider range of navigation tasks, this paper presents a novel waypoint navigation method in polar coordinate. Navigation limit cycle avoidance is handled by creating and memorizing a specific traversable but less preferred areas during waypoint navigation in which the robot changes its orientation to follow from goal-directed to opportunistic goal-repulsive behavior. This allows the robot to be able to predict, thus avoiding upcoming limit cycle situations. The saving of time and memory resources of path computation using the novel polar-coordinate based waypoint navigation has been verified by simulations, and an implementation of the navigation method on a real mobile robot works well in an indoor experiment with demonstrated planning and route selection capabilities for waypoint navigation.

I. INTRODUCTION

Autonomous robots or unmanned vehicles capable of real-time exploring and executing tasks in an unknown environment using onboard sensing, computing and communication devices has been focused and studied by a lot of researchers. This technology is not only for the research challenges, but also required urgently to solve some problems in the human society. For example, using a robot or a group of robots to measure damage level and search for victims in a dangerous building or disaster [1], or border searching and patrol in military missions, or to perform home care and security duties in residential.

No matter what kind of task been assigned to the robot, it is necessary that the robot has the capable of locally planning paths between two locations by assuming that a robot is modeled as a point or an object with simple shape that is able to translate and rotate (e.g. Dubins or car-like vehicles) with a curvature constraint. The A* algorithm in grid maps to search a lowest cost path between two points and its extensions, such as focused D* [2] or Generalized Adaptive A* [3], have been developed to deal with more complicated dynamic environments using path replanning, and could be integrated into robot motion control [20] in real world. On the contrary, the fuzzy-logic control system [4][5][21] is a flexible and having high success rate method for robust obstacle avoidance in the presence of uncertainties [22],[23],[26]. It efficiently handles ambiguous data, processes logical calculation, and outputs a clear and simple answer. Though the answer usually doesn't form a

shortest path, it is still able to adapt to most of environment types, thus making fuzzy-logic system a very popular navigation method. In addition, Ecological robotics is an interesting science. Inspired by the wonders of nature, researchers study and mimic animal or insect behaviors to design some strategies to adapt to unknown environments. For example, navigation using equiangular spiral for target reaching is inspired by the behavior of insects flying toward a candle [19]. Nowadays, ecological robots not only go after animals or insects, they can even perform more difficult tasks or move faster with the help of modern techniques such as Q learning [6] or Kobayashi's landmark identification [11]. Potential field method [9], [10] transforms geographic information into potential fields in which attractive fields could come from targets, while repulsive fields could come from obstacles. The net influence of fields around the robot generates the movement of the robot to avoid the stationary or dynamic obstacles and reach the target.

For limited sensing range, waypoint navigation system [17],[18],[28] is a popular navigation approach by planning a sequence of subgoals that leads the robot to the target. To develop a more efficient waypoint navigation system, a deliberative layer could be integrated in the higher level to provide global planning capability based on accumulated knowledge of the environment and navigation experience stored in memory [17]. In this aspect, some features of ecological robotics, especially those learned from human intelligence in navigation, are helpful. For example, while walking in an unknown environment, we act according to what we see in the limited range, but we won't scan every grid in the vision, like an A* algorithm do, nor we won't evaluate distances of all the surrounding objects, like a fuzzy-logic do. Instead, human find the edges of obstacles, corners of walls, or the end of sight because those are mediators leading to other unknown areas, which helps explore the environment. By chance, human can escape from current situation in an unknown environment.

For sensor-based reactive navigation, it is well known that the potential conflict between the behaviors of obstacle avoidance and reaching the goal can yield local minima or limit cycle situation, mostly due to unawareness of previously visited locations in U-shaped or T-shaped obstacles. There are many researches dealing with this kind of problems. Wall following is one of the main approaches to resolve the limit cycle in navigation, as used in Bug family algorithms. Most of research record the trajectories of the robot or characteristics of the environment [4],[12]-[13],[22]-[29] to recognize visited and unvisited areas represented in Cartesian coordinate, or abrupt change of robot heading of amount larger than a threshold [26], e.g. from turn-to-right to turn-to-left and vice versa [26] for identifying repeated traversal of the same regions. Techniques are developed to guide the robot to escape from limit cycle

This work was supported by National Science Council of ROC under contract NSC 101-2221-E-001-001

Yu Lun Ting, Jing Shi Liu are with Institute of Information Science, Academia Sinica, (e-mail: liu@iis.sinica.edu.tw)

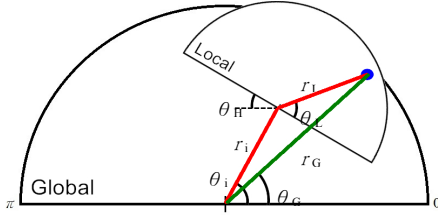


Figure 1. Transformation between global map seen from the target and local map seen from the robot

situation upon detection of a limit cycle path. Some of the approaches are the setup of a virtual wall [24] or virtual target [26], or adaptive rotation using clockwise or counterclockwise turning along a periodic orbit [25]. [26] proposed to separate the deliberative (A*) layer and reliable reactive (DH-bug) layer based on bug algorithm of control system architecture on grid map to escape from traps and improve the efficiency.

Use of polar coordinate in mobile robot navigation offers some advantages in kinematic control of unicycle mobile robot [30],[31], where the distance and direction of the obstacles to the current robot position or goal are explicitly described along with some indication of their size. In view of limited energy and memory in a typical autonomous robotic system for a long distance of travel and a long time of operation in an environment whose size and structural characteristics such as obstacle geometry (convexity and concavity) and distribution are unknown, using polar coordinate system is more suitable for navigation task. Inspired by potential field method, we propose a strategy called Repulsive Area, with the help of the polar coordinate system, the creation of Repulsive Area can be independent of the size of the global map, thus it is more suitable for unknown environments. Furthermore, compared with other methods tackling limit cycle, Repulsive Area occupies less memory spaces, especially significant when the explored areas are growing.

This paper is organized as follows. Section 2 presents the polar coordinate system as working coordinate of mobile robot navigation. Section 3 presents the human-like waypoint selection method with respect to robot's current position. Section 4 presents the novel concept of Repulsive Area and discusses its advantages for limit cycle avoidance. Section 5 presents our simulations results to show the efficiency and memory requirement by using polar coordinate in navigation method for different sensing range. Section 6 shows an indoor experiment result of the proposed waypoint navigation method implemented on a mobile robot equipped with a laser range finder. The final section makes a conclusion.

II. POLAR COORDINATE SYSTEM

In this paper, we propose a full polar coordinate navigation system [31]. In polar coordinate, a global map is the map seen from the target, which records all the known environment information, trajectories of the robot, and the current positions of the robot and the target. A local map is set up at robot to show open areas, obstacles, and the candidate next waypoints within sensing range of the current robot position. As shown in Fig. 1, suppose the current robot position is $W_i = (r_{W_i}, \theta_i)$ with θ_i the heading angle. A local polar coordinate frame whose origin is

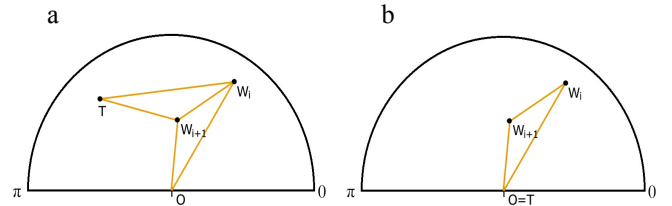


Figure 2. Coordinates calculation

W_i is defined at the current robot position. Given a direction θ , a depth r is defined on every point within the field of view (FOV), thus defining its coordinate as (r, θ) . Let (r_G, θ_G) and (r_L, θ_L) represent the global and local polar coordinates of the specific point, respectively. The transformation of converting a local coordinate to global coordinate is

$$\begin{aligned} \theta_a &= \theta_i - \theta_L - \theta_H \\ r_G &= \sqrt{r_L^2 + r_i^2 - 2r_L r_{W_i} \cos \theta_a} \\ \theta_G &= \theta_i + s \cdot \cos^{-1} \left(\frac{r_{W_i} - r_L \cos \theta_a}{r_G} \right) \end{aligned} \quad (1)$$

where θ_a is the included angle of r_G and r_L , s is a sign function,

$$s = \begin{cases} 1, & 0 < \theta_a \leq \pi \\ -1, & \text{else} \end{cases}$$

From a geometric viewpoint, (1) can be seen as calculating a triangle. To give a further explanation, we take the next waypoint W_{i+1} as an example. Assume W_{i+1} has been calculated in the local map from current robot position W_i (also the origin of the local map). Next, use the transformation to calculate the absolute position of W_{i+1} , since OW_i is also known, it can be calculated from $\triangle OW_i W_{i+1}$ in Fig. 2a. Besides, the relationship between the target T and W_{i+1} should be figured out, because the robot has to know the bearing to the target when it arrives W_{i+1} . The same, it can be calculated from $\triangle TW_i W_{i+1}$ in Fig. 2a. To obtain the absolute position of W_{i+1} , and its relationship with the target T , we have to calculate $\triangle OW_i W_{i+1}$ in Fig. 2b.

III. WAYPOINT SELECTION

In a planar unknown environment $\mathcal{W} \subset R^2$ populated with static obstacles $OB_{i,i=1,}$, guiding a robot effectively to reach the target is the major objective. Besides, a good navigation system should possess the function to keep the robot in a safe, trapped-free state at all time intervals and if possible, it should be smart enough to provide a high efficiency movement strategy. In static environments, waypoint navigation [17],[18], [28] is a popular reactive navigation method in practical applications. The robot follows an ordered set of waypoints which guides the robot to reach the goal, despite its robustness issue to robot localization errors [18]. Waypoint navigation has the advantage of reduced memory storage, since only the sequence of waypoints are stored instead of whole path. The waypoint navigation system we propose here incorporates additional designs: (a) Using polar coordinate as working coordinate, (b) Marking the regions associated with waypoints at which

navigation to goal is blocked, which tells the system trapped-free paths, (c)Traveling as distant as possible a longer reachable distance from a waypoint, to reduce the overall time of target reaching. The detailed procedure and mechanism of our system will be described in the following.

A. Deploy candidate waypoints

Waypoints or sub goals are collision-free locations which intend to guide the robot towards the goal. The robot begins to change its heading as it passes through a waypoint [17], [28]. Before each movement, the waypoint navigation system will deploy several waypoints located in free space within the robot's sensing range, and one of them is chosen as the next waypoint for local navigation. For deployment of waypoints random, or improved random methods [14][15] could deploy massive waypoints for exploration-based navigation. However, over concentrated waypoints without limiting the distances between waypoints could inch the robot, and might guide the robot toward obstacles. Our deployment method is to deploy points at the edges between objects and open area, and the distances are as far as possible [16]. For safe and smooth navigation, all the deployed points must keep a safety distance away from the objects by taking into account the volume and dynamic motion characteristic (e.g. nonholonomy and smoothness) of the robot [Me]and errors in robot and waypoint localization [18] in real navigation.

Let the maximum sensing range and field of view of range sensor be $D>0$ and $FOV = 2\beta \in [0, 2\pi]$, respectively. Let $\Omega_i = \Omega(D, \beta, W_i)$ be the area within the sensing range and FOV at the current robot position of waypoint W_i . The open areas and the obstacles within Ω_i are denoted as $\{OA_j, j \in J\}$ and $\{OB_k, k \in K\}$, respectively. The robot takes a scan to obtain the local environment data Ω_i around W_i

$$\begin{aligned} \Omega_i &= \{OA_j, j \in J\} \cup \{OB_k, k \in K\} \\ &= \{(r, \theta), r \leq D, |\theta| \leq \beta = 0.5FOV\} \end{aligned} \quad (2)$$

and

$$\{OA_j, j \in J\} \cap \{OB_k, k \in K\} = \{\Gamma_l, l \in L\}$$

where Γ_l the variances, i.e. the edges/contours between any OA_j and OB_k , of the geometric discontinuities of Ω_i .

Our current implementation sets the points located at a threshold distance δ away along the direction of geometric variance as the newly created candidate waypoints to allow sufficient space of navigation. Let $V_p = \Gamma_p \pm \delta, p \in P \subseteq L$ denote the collision-free point at a predefined fixed distance δ away from Γ_p in the normal direction of Γ_p (\pm for right or left of the obstacles). δ is less than maximum sensing range (assuming larger than turning radius), which is related to the motion constraint and robot physical size. In addition, let

$$\begin{aligned} \Pi_i &= \{x_\alpha : x_\alpha = \text{random sample}(\{OA_j, j \in J\}), \\ \alpha &= 1, \dots, N\} \end{aligned} \quad (3)$$

be N random sampling points of the free space of current robot location, indicating the collision-free directions toward which the robot desires to turn from current position. One point in each direction is recorded as candidate next waypoint to be traversed. The number of directions directly relates to a tradeoff of the computational effort and the ability to detect openings. In summary, by exploring navigation-related locations when the robot is at the i_{th} waypoint W_i , the set of candidate $(i+1)_{th}$ waypoints C_i in Ω_i is opportunistically selected as

$$C_i \in \{V_p\} \cup \Pi_i \quad (4)$$

Note that this set and Ω_i are updated when the robot reaches a new waypoint. Fig. 3 illustrates the deployment of candidate waypoints.

B. Select the next waypoint

The choice of next waypoint is largely dependent on the maximum sensing range. In [28], the sensing range (a semicircle or a sector) is divided into three equal zones (left, right and front), and the waypoint located in the opening that has the minimum direction difference both from the target and from current robot heading was chosen [28]. In our system, there are three behavior modes to consider: the ordinary modes of reaching the target, avoiding the obstacles in waypoint navigation and one additional mode of avoiding the limit cycles. Thus, the determination of a proper desirable direction θ_d executed by steering control of mobile robot is based on:

(a). Line-of-sight at the target for goal reaching behavior, θ_{target} such that r is decreasing. (b). Directions which allow the robot to escape from limit cycle situations, θ_{leave} ; these could be a set of opportunistic directions. (c). Forward collision-free direction θ_{obs} , chosen from (4) for avoiding obstacles. Thus, θ_d is based on the minimization of a weighted sum over all candidate waypoints θ_{obs} chosen from (4) of current robot position:

$$\begin{aligned} \theta_d &= \arg \min_{\theta_{obs}} E(\theta_{target}) \cdot w(\theta_{target}) \cdot \theta_{target} \\ &+ E(\theta_{leave}) \cdot w(\theta_{leave}) \cdot \theta_{leave} \\ &+ E(\theta_{obs}) \cdot w(\theta_{obs}) \cdot \theta_{obs} \end{aligned} \quad \square(6)$$

where $w(\cdot)$ denotes a weight, $E(\theta)$ is a binary symbol indicating the nullity ($E(\theta)=0$) of a not considered direction θ or otherwise. In the following situations, $E(\theta)$ is determined as 0.

(a). When the robot meets blocking obstacles that cause limit cycle, the priority is to leave this region using some escaping strategy, and $E(\theta_{target})=0$.

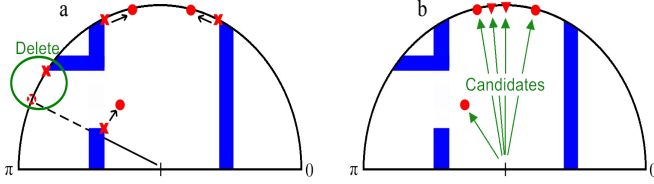


Figure 3. Candidate waypoints. Points “X” in Fig. 3a are located at the variances of the environment. “●” points, which are shifted a δ distance from points “X”. Shifted points that are in collision with obstacles are deleted. “▼” in Fig. 3b are randomly deployed points in open areas.

- (b). $E(\theta_{\text{obs}})=0$, i.e. the robot does not move forward when
- (i) there is not any obstacle in the way to target, or
- (ii) the forward direction is close to any blocking obstacle or a dead end and for safety. In this situation, the priority is to find a visible edge or corner of obstacles within sensing range to make a change of forward direction to visible openings that are likely to be closer to the target.
- In summary, given the set of candidate waypoints of robot current locations, the criterion used to select the next waypoint is defined as the one with least deviation from the calculated desirable direction of (6)

$$W_{i+1} = \min_{X \in C_i} |(\text{angle of } X) - \theta_d| \quad (7)$$

within the sensing range.

IV. REPULSIVE AREA

A. Limit cycle situations

Either the sensing range is limited and may be blocked to detect the extent of obstacles [28] or the errors in waypoints [18], it is likely that the robot is stuck in a dead end or a limit cycle situation. Although wall-following is a method to escape from limit cycle, it is not efficient due to the overly long path to the goal. Xu [26] presented a virtual target approach to overcome limit cycle problem in fuzzy behavior-based navigation. The direction of target attraction is changed at a critical point that abrupt change of target attraction occurs. Zhu and Yang [29] proposed a state memory scheme to memorize the relative directions (both left or both right or the other situations) of target and obstacles at previously visited locations, so that the robot could be aware of limit-cycle path.

Pin and Bender [13] presented a virtual obstacle concept to remedy limit cycle problems. By comparing current position of the robot and those stored in the memory, the robot will create a rectangular area as a virtual obstacle to prohibit the robot from entering regions that have generated navigation limit cycle if it recognizes the place as visited before, and at the same time a virtual subgoal is set in the opposite direction to the real goal. However, this method could cause a problem. For example, assume the robot starts circling in a region. Although previously visited positions are at hand, all new waypoints fail to be set at the vicinity of sparsely old waypoints, making the robot hard to

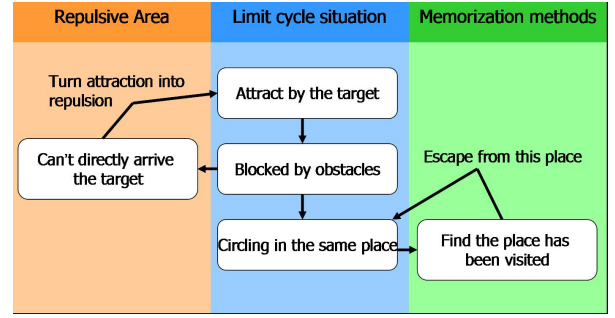


Figure 4. Methods handling limit cycle situation

recognize that its current position has been previously visited. Wang and Liu [4] proposed that each square grid stores two types of information: Obstacle Memory Dot and Trajectory Memory Dot. These values will be increased or decreased every time the grid is sensed and updated. The robot tends to go to the region with minimum sum of grid values, so that the robot can avoid enroute obstacles and revisiting visited regions. Compared with Pin and Bender's method, this method guarantees the iteration situation can be detected by calculating regional values. However, since it needs to store all the data in the whole work area, and the size of the environment is unknown, overflow could occur. To tackle with navigation limit cycle, we propose Repulsive Area defined as the location where there is no way for the robot to arrive the target by a simple target-oriented navigation method. An area is marked as Repulsive Area only when the robot passes across this area attempting to go to the target but is blocked by any obstacle. A typical example is dead end. Once encountering this situation, our system exploits opportunistic navigation by switching from goal-directed navigation to goal-repulsion in Repulsive Area, which is conceptually similar to the strategy of virtual target. Thus, the robot won't get stuck in Repulsive Area, and can seek other probable routes freely in an opportunistic manner. A notable distinction from virtual wall approach [24] is that a region marked as Repulsive Area is not prohibited for traversing, but the preferred motion direction is goal-repulsion. Fig.4 shows the scheme of our method in comparison with trajectory memorization methods.

B. Updating Repulsive Area

Let $W_i = (r_{W_i}, \theta_i)$ denote polar coordinate of the current waypoint in the global map, where the radius r_{W_i} is equal to the Euclidean distance of current waypoint to the target. The update rule of Repulsive Area every time the robot reaches a waypoint is as follows.

$$\begin{aligned} &\text{if (mode = target oriented) and } r_{W_{i-1}} < r_{W_i}, \\ &\text{then Repulsive}([\theta_j / 0.5] + 1) = R(\theta_j), j = i - 1, i \end{aligned} \quad (8)$$

where $R(\theta_j)$ is a set of radius values from W_i to W_{i-1} . In (8), $r_{W_{i-1}} < r_{W_i}$ means there is an obstacle between the robot and the target. In this case, the system will calculate the straight line from W_i to W_{i-1} , then retrieve the radius values of points on this straight line every 0.5° .

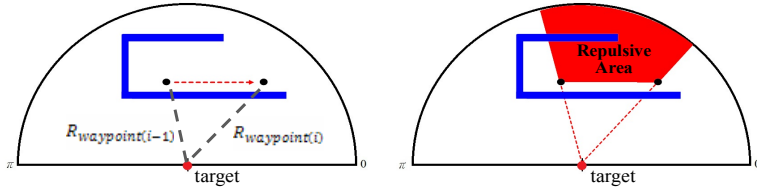


Figure 5. Creation of Repulsive Area as the robot moves from W_i to W_{i-1}

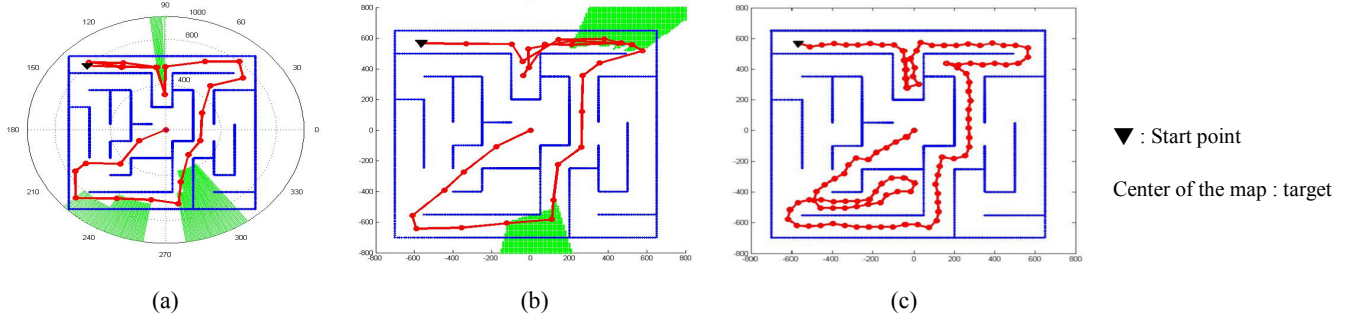


Figure 6. The size of the map is 14x14 m, and the width of most corridors is 1.5 m. (a) path in polar coordinate, length 62.75 m (b) path in Cartesian coordinate, length 56.58m (c) path obtained by [4], length 82.47 m. The green areas represent the Repulsive Area.

These radius values are stored in Repulsive Area matrix with index $[\theta_j / 0.5]$. According to the stored data, an example of Repulsive Area is then created as Fig. 5 shows.

C. From goal attraction to repulsion

According to the definition, when the robot reaches a waypoint $W_i = (r_{w_i}, \theta_i)$ at which (8) achieves the optimum for W_α, W_β , it is stored and (9) is used to construct a repulsive area A at which limit cycle may very likely occur. A path is planned out from the area. Repulsive area is thus recognized as a landmark for possible occurrence of limit cycle and the rule of preferred direction is changed from heading-to-goal into (9),

$$\begin{aligned} \text{If } \text{Repulsive}([2 * \theta_i]) \leq r_{w_i} \\ \theta_{\text{target}} = \theta, \theta \notin [\angle(\overrightarrow{W_i O}) - \theta_s, \angle(\overrightarrow{W_i O}) + \theta_s] \\ \text{Else } \theta_{\text{target}} = \angle(\overrightarrow{W_i O}) \end{aligned} \quad (9)$$

where θ_s is a suitable side angle that the mobile robot is allowed to turn. (9) extends the direction toward the target into a range of directions define by circular sector with angle $\angle(\overrightarrow{W_i O}) - \theta_s, \angle(\overrightarrow{W_i O}) + \theta_s$ when the robot is inside the Repulsive Area.

V. SIMULATION RESULTS

The navigation method is demonstrated by the simulations of an orthogonal maze-like environment shown in Fig. 6 with comparison with [4]. The mobile robot is a 50x50 cm rectangle, and its sensing range is 3 m. There are several dead ends, corners, traps and multiple routes, which is a conventional challenge of the navigation efficiency [4], [8] and there may be no clear routes to the goal. In Fig. 6a, the target is located at the

origin of the polar coordinate system, and the green areas represent the Repulsive Area with respect to the traveling path. There is little difference between two paths in polar and Cartesian coordinates because they utilized the same method, and the slight difference is caused by the opportunistic direction selection. Fig. 6c is the path obtained by Wang and Liu's method [4] which stored the trajectory as "Trajectory Memory Dot" during exploration. However, reaching the target by the navigation method of [4] requires a longer moving distance, thus higher computational cost in escaping from dead ends and fitting the configuration of the maze. Table 1 indicates that less memory is needed using polar coordinate. This is not only because Cartesian coordinate needs additional coordinate transformation, but also has to process more data from Repulsive Area. Since the sensor range is limited, we perform simulations to study the effect of different sensing range on navigation behavior. In general, the shorter the sensing range, usually the more waypoints deployed in a map and the closer the deployed waypoints, while the longer the sensing range, the more edges are detected and thus more helpful for avoiding or escaping from the traps using waypoint navigation [28].

TABLE I. MEMORY SPACES REQUIREMENT IN MATLAB

	Polar	Cartesian
Memory spaces (kb)	5.6	315.6

TABLE II. COMPUTATION TIME PER STEP (S) FOR DIFFERENT SENSING RANGE (AVERAGED OVER 100 RUNS)

scenario I					
Range (cm)	300	350	400	450	500
Polar	0.0550	0.0554	0.0545	0.0539	0.0545
Cartesian	0.0961	0.0874	0.0951	0.0939	0.0906
Time saved	42.7%	36.6%	42.7%	42.6%	39.8%

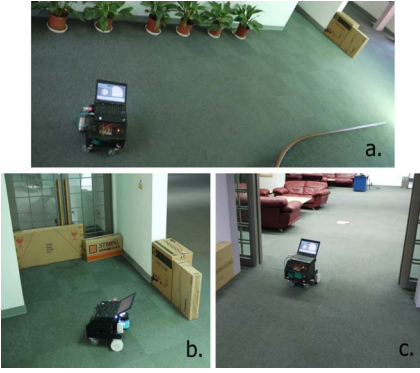


Figure 7. Experimental snapshots of the proposed waypoint navigation system in action.

TABLE III. PATHS INFORMATION FOR DIFFERENT SENSING RANGE(AVERAGED OVER 100 RUNS)

Range (cm)		300	350	400	450	500
Polar	travel distance	73.04	67.78	65.58	62.49	59.78
	# step	38.82	37.33	34.73	33.36	33.28
Cartesian	travel distance	68.28	70.54	70.11	64.52	59.30
	# step	36.29	38.74	37.14	35.24	33.31

Table 2 shows that using polar coordinate the average savings in path computing time is are about 40 for different sensing range. Thus, use of polar coordinate could account for the practical memory and computing time requirements for real-time navigation tasks, especially for long tome or long path. Table 3 provides additional information of paths in simulations. In general, sufficiently larger sensing radius is beneficial in shortening the path length. In the two scenarios, similar results are obtained for different sensing range since the same navigation method is employed in small maps, despite which coordinate system is used. It is shown that running the navigation algorithm in Cartesian coordinate system inherently needs more time, not because of traveling longer distances. In addition, waypoint navigation seems more constrained by the topology of environment and it can work well with weak range sensor in small maps.

VI. EXPERIMENTS

Implementation and testing of the waypoint navigation system is conducted on a mobile robot of size 30 cm* 50 cm* 40 cm. It is equipped with a SICK laser range finder in the front to perform a scan of 240 degrees of FOV with maximum sensing range 5m to obtain nearby environment information, and two integrated servo motors to drive left and right wheels. A 2.4 GHz notebook is attached on the top of the robot to execute calculation and navigation algorithm. An indoor experiment is performed at the lobby in the first floor of office building. The target is set at the center of the lobby, slightly close to a couch, and the initial position of the robot is outside of the lobby and is heading to goal (Fig. 7a). This is a situation where the target is located on the other side of a wall that may cause limit cycle

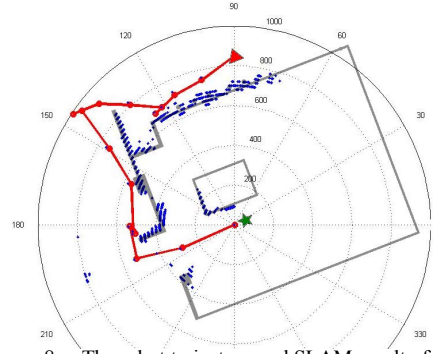


Figure 8. The robot trajectory and SLAM result of the boundary of experiment site in a safe navigation, where the red triangle and green star indicate the start and the target, respectively. The navigation system explored and entered and steered away a trap in the mid-way, and selects a path through the opening toward the target using a laser range finder.

using reactive goal-directed navigation. Given the start and target, the robot employs the locally available range data to find the way toward the doorway into the lobby without being trapped (Fig.7b), and finally find the way into the lobby (Fig.7c) using our navigation system. This experiment can be viewed online at <http://youtu.be/jGuq4SsEN9Q>.

In this experiment, the robot stops and turns before moving forward. Motor encoders are used to localize the robot. It is seen that localization errors in robot and waypoints accumulated over time [18] give rises to the error in target position. Fig.8 indicates the trajectory encapsulating the robot navigation behavior and the indoor environment information obtained by SLAM technique. The robot starts from the red triangle, moves to its next waypoint (red dots) along the red line one after another, and finally reach the target (green star). The massive blue dots in this figure which draw the outlines of objects are obtained from the laser range finder, and these data are exactly the input of our navigation system. A complete series of action executed by the mobile robot is as follows:

(a) Stop at the current waypoint for a very short period of time to scan the environment, process the data and based on the information in front of the robot to calculate candidate next waypoints.

(b) Turn to the direction of the best candidate waypoint closest to a desirable direction according to the rule (6).

(c) Move a step along straight line to the best candidate waypoint in constant speed using point-to-point navigation, so that a path is a polyline joining waypoints for the mobile robot to follow. After reaching the desired waypoint which tends to bring the robot closer or more ready to get closer to the target, the waypoint navigation system updates the current waypoint

(d) Reach the target or go to Step (a)

The massive blue dots in Fig. 8 which draw the outlines of objects, are obtained from the laser range finder based on measurements of actual execution, and these data are exactly the sensory input of environment information (i.e. the world map available) to our navigation system. Note that uncertainties in the map data could influence the risk assessment of waypoint

navigation and thus a safety distance is incorporated into the waypoint selection process. To simplify the display of this figure, the resolution has been down to 20 cm, though in the navigation system it is originally 1 cm for calculation purpose. Besides, the gray lines refer to the scales from practical measurement.

VII. CONCLUSION

We proposed a waypoint navigation method based on polar coordinate embedded with a memory component of repulsive area for navigation of a mobile robot in an unknown environment. Strictly intended for avoiding limit-cycle situation that may be encountered, repulsive area is created and memorized for switching the goal-oriented behavior based on prior navigation experience of limit cycle situations to resolve the limit cycle situations. The candidate waypoints for robot motion are near corners or edges of obstacles with clearance to avoid the deployment of waypoints into the obstacle interiors or randomly deployed waypoints at areas free of any obstacles. Next, the waypoint navigation system is efficient by combining reactive navigation with limit cycle prediction based on Repulsive Area for waypoint selection. By using polar coordinate system as working coordinate of robot motion, in addition to the explicit description of range and direction of obstacles to current robot position and goal, we demonstrate that the path computing time is lowered and the required memory space is fixed by the angular resolution used to discretize the map, no matter the size of explored area or the travel distance of the robot. The promising efficacy of our waypoint navigation method is demonstrated by both simulations and indoor experimental testing. More complex case is ongoing.

REFERENCES

- [1] R. R. Murphy, "Activities of the Rescue Robots at the World Trade Center from 11–21 September 2001," *IEEE Robotics & Automation Magazine*, September 2004.
- [2] A. Stentz, "The focused D* algorithm for real-time replanning," *International Joint Conference on Artificial Intelligence*, August 1995.
- [3] X. Sun, S. Koenig, and W. Yeoh, "Generalized adaptive a*," *7th International Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 469–476.
- [4] M. Wang and J.N.K. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends," *Robotics and Autonomous Systems*, vol. 56, pp. 625–643, 2008.
- [5] X. Yang, M. Moallem and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 6, pp. 1214–1224, 2005.
- [6] T. Yamashiro and K. Ito, "Comparative study of affordance-based navigation and model-based navigation," *IEEE, International Conference on Robotics and Biomimetics*, December 2011.
- [7] V. J. Lumelsky and A. A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Transactions on Automatic Control*, vol. AC-31, no. 11, pp.1058–1063, 1986.
- [8] J. Ng and T. Bräunl, "Performance Comparison of Bug Navigation Algorithms," *Journal of Intelligent and Robotic Systems*, 2007.
- [9] S.S. Ge and Y.J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, pp.207–222, 2002.
- [10] C. Ye, "Navigating a mobile robot by a traversability field histogram," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 2, APRIL 2007.
- [11] H. Kobayashi, K. Kikuchi, K. Ochi and Y. Onogi, "Navigation strategies referring to insect homing in flying robots," *IEEE International Conference on Robotics and Automation*, May 2001.
- [12] K.M. Krishna and P.K. Kalra, "Perception and remembrance of the environment during real-time navigation of a mobile robot", *Robotics and Autonomous Systems*, vol. 37, pp.25–51, 2001.
- [13] F.G. Pin and S.R. Bender, "Adding memory processing behavior to the fuzzy behaviorist approach: Resolving limit cycle problems in mobile robot navigation", *Intelligent Automation and Soft Computing*, vol. 5, no.1 ,pp.31–41, 1999.
- [14] K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Anytime dynamic path-planning with flexible probabilistic roadmaps," *IEEE International Conference on Robotics and Automation*, 2006.
- [15] M. S. Branicky, S. M. LaValle, K. Olson and L. Yang. "Quasi-randomized path planning," *IEEE International Conference on Robotics and Automation*, 2001.
- [16] Y. S. Chou and J. S. Liu, "An efficient path planner combining random sampling and a fuzzy system," *IEEE International Conference on Methods and Models in Automation and Robotics*, August 2009.
- [17] Y. Wang, A. Mulvaney, I. Sillitoe and E. Swere, "Robot navigation by waypoints," *Journal of Intelligent and Robotic Systems*, vol.52, pp.175–208, 2008.
- [18] K. Demura and Y. Komoriya, "A navigation method using the mutual feedback of waypoints and self-positions," *Advanced Robotics*, Vol. 26, Issue 14, pp.1677–1691, 2012.
- [19] H. Teimoori and A.V. Savkin, "Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements," *Robotics and Autonomous Systems*, vol. 58, pp.203–215, 2010.
- [20] M. Seder, K. Macek and I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," *31st Annual Conference of IEEE Industrial Electronics Society*, 2005.
- [21] A. Al-Jumaily and C. Leung, "Wavefront propagation and fuzzy based autonomous navigation," *International Journal of Advanced Robotic Systems*, vol. 2, no. 2, pp. 93–102, 2005.
- [22] K.M. Krishna and P.K. Kalra, "Solving the local minima problem for a mobile robot by classification of spatio-temporal sensory sequences," *Journal of Robotic Systems*, vol. 17, no. 10, pp.549–564, 2000.
- [23] O. R. Esmaili Motlagh, T. S. Hong and N. Ismail, "Development of a new minimum avoidance system for a behavior-based mobile robot," *Fuzzy Sets and Systems*, vol.160, pp.1929–1946, 2009.
- [24] C. Ordonez, E.G. Collins Jr., M.F. Selekwia and D.D. Dunlap, "The virtual wall approach to limit cycle avoidance for unmanned ground vehicles," *Robotics and Autonomous Systems*, vol.56, pp.645–657, 2008.
- [25] L. Adouane, "Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation," *9th Conference on Autonomous Systems and Competitions*, 2009.
- [26] W.L. Xu, "A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot," *Robotics and Autonomous Systems*, vol. 30, pp. 315–324, 2000.
- [27] Y. Zhu, T. Zhang, J. Song and X. Li, "A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge," *Knowledge-Based Systems*, vol. 27, pp.302–313, 2012.
- [28] C. Ye and P. Webb, "A sub goal seeking approach for reactive navigation in complex unknown environments," *Robotics and Autonomous Systems*, vol.57, pp.877–888, 2009.
- [29] A. Zhu and S.X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 4, pp. 610–621, 2007.
- [30] C.-H. Hsieh and J.-S. Liu, "Nonlinear model predictive control for wheeled mobile robot in dynamic environment," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp.363–368, 2012.
- [31] F. Belkhouche, B. Belkhouche and P. Rastgoufard, "Autonomous navigation and obstacle avoidance using navigation laws with time-varying deviation functions," *Advanced Robotics*, vol. 21, no.5–6, pp.555–581, 2007.