A Hybrid Strategy for Robot Navigation in Semi-structured Environments

Guilherme C. R. de Oliveira, Kevin B. de Carvalho
Departamento de Informática
Universidade Federal de Viçosa
Viçosa MG, Brasil
Email: guilherme@ufv.br, kevin.carvalho@ufv.br

Alexandre S. Brandão
Núcleo de Especialização em Robótica
Departamento de Engenharia Elétrica
Universidade Federal de Viçosa
Viçosa MG, Brasil
Email: alexandre.brandao@ufv.br

***Abstract*—Autonomous mobile robots should be able to navigate on different types of environments. Some strategies deals with the navigation problem by taking into consideration only the obstacle avoidance task, adopting a reactive behavior. Others, at cost of increased hardware processing, look for optimal paths that must be replanned once an obstacle appears in the planned path. The proposal of this paper is to integrate the A* search and obstacle avoidance tangential escape algorithms, in order to overcome the A* limitations in non-static environments. Instead of calculating a new route to avoid collisions with unexpected obstacles, as in most popular solutions in the literature, the robot perform deviations while it tries to keep on the planned route. The strategy also uses the robots sensor data for mapping the environment in an Occupancy Grid, in order to navigate deliberatively. Numerical simulations validate the proposal.**

***Index Terms*—path planning, local planning, global planning, robot navigation**

I. Introduction

Planning is a topic within the mobile robotics which involves autonomous platforms that are able to perceive and act in an environment making intelligent decisions. When developing planning algorithms, we need to take into consideration the amount of information regarding the capacities of sensing, degrees of freedom of the robots movement as well as the environment where they move [1]. For the situations where there are obstacles and the environment is unknown, the robot should count only with information received by its sensors in order to move. In this scenario, strictly reactive strategies, where actions are usually taken in real time, are carried out in order to prevent collisions. On the other hand, in situations where there is some kind of world representation, whether it is total or partial, a deliberative strategy may be used considering information about free zones and obstacles in order to plan paths. In this second case, we should take into account the problems with mapping and path planning [2], [3], as well as replanning strategies for unexpected situations, so that the robot is able to move in an autonomous way.

The aim of this article is to develop a navigation strategy with planning for terrestrial robots, in environments that can be mapped a priori or not. The algorithm also takes into consideration the limited capacity of the robots sensors and processing, typical characteristics of low cost platforms. The mapping algorithm Occupancy Grid [4] is used to model the environment. The A* search planning algorithm [5] and the Tangential Escape obstacle avoidance strategy [6] are combined for navigation, so that whenever possible, the robot can follow an optimal path and avoid unexpected obstacles. The strategy considers any type of obstacle, known a priori or not, so that the deviations are carried out during the navigation, in a dynamics way. Besides the obstacles, the avoidance strategy is also used to prevent collisions that may be caused by small noises in sensors, odometry or mapping.

The general idea for the algorithm is to accumulate information from the environment so that the navigation is improved as time passes by. When a destination point is defined, a path within the mapped area, if it exists, is traced. Otherwise, the robot adopts a reactive behavior. In both cases, the robot has mechanisms to avoid obstacles, whether they are known or not. The strategy is designed for indoor and structured environments, where it can be related to household chores, as with robots that do cleaning tasks, move heavy things or take care of the elderly [7].

The manuscript is splitted in the following way: in Section II we talk about related works, in Section III, we describe the techniques that are used. In section IV we discuss in details of the hybrid navigation proposed in this paper. In section V the tests are shown and the results are discussed, and then in section VI we conclude the article, taking the chance to mention possible future works.

II. Related works

The work done on this paper connects itself directly with two problems of mobile robotics: mapping and planning. The first one is solved using the Occupancy Grid algorithm, that may be related to other mapping algorithms which also use grids or assume that the robots' poses are known. For an example, in [8] [3], the proposed algorithm uses a map differentiation technique to identify the environment objects, defining objects models and their positions in the real world. This type of approach has shown to be difficult to integrate in the proposed strategy of this work, taking into consideration that the map may not be modeled a priori. Also, there might be an increase in the processing cost to identify the objects as well.

978-1-5090-5949-2/18/$31.00 ©2018 IEEE

The choice of mapping algorithm and the way of modeling the environment impose restrictions in the definition of the path planning strategy. Sampling-based techniques in the literature, such as the Probabilistic Roadmaps Method (PRM) [9] finds a set of obstacles-free paths for the navigation; however, the algorithm aims modeling in visibility graphs and Voronoi diagrams. The RRT algorithm [10] creates a tree of paths in a random way, from the robot to the destination point. Asymptotic optimal solution for RRT, also called RRT* is presented in [11], while PRM* proposed in [12] is presented as a computationally efficient solution for PRM.

Several metaheuristics have also been proposed for solving the path planing problem. The ant colony Optimization [13] [14] uses the idea of pheromones for keeping an path that avoid obstacles and optimize it on time. Particle swarm optimization and genetic algorithm approaches have also been implemented [15] [16]. Even though behaviour-based methods succeed in find a path quickly in the real world, they struggle to reach the optimal convergence to the solution.

The A* search algorithm [5] is able to find, efficiently, an optimum path in a grid, but it fails for not offering a dynamic strategy for obstacle avoidance. This problem is solved by some implementations that look for optimum paths at non-static environments and uses replanning tools to avoid unknown obstacles [17] [18]. These strategies, despite efficiency, require a higher computational cost. Thus, as proposed in this work, we integrate the A* search with a local planning approach for avoiding obstacles in real time.

In the context of local planning, one can cite famous approaches such as the Potential Field method, explained and with its limitations detailed by Boreinstein in [19]. Improvements on the Potential field method can be found at [20]. In [21] [22] it is added different techniques to PFM, such as Evolutionary algorithm. The authors in [23] present an improvement of the Virtual Force Field: the Vector Field Histogram. Such strategy work on the creation of a probabilistic histogram through the information of the robots sensors and change the robots navigation direction analyzing the proximity and the value of each histogram cell.

The algorithm of tangential escape [6] aims to avoid obstacles creating a virtual point so that the robots trajectory in a straight line is tangent to the obstacle. Its implementation, as well as the Pontential Field and other local strategies, suffer with local minima and may lead to inefficient paths until it reaches the destination point. Therefore, a global planning strategy is integrated to the local planning proposed in this work, in order to increase the locomotion efficiency, avoid obstacles in real time and decrease the risk of local minima.

III. Planning

A. Mapping

The Occupancy Grid mapping algorithm [4] discretizes the environment in cells, which represent a local area and have binary values that indicate whether that area is occupied or free. A map m modeled as Occupancy Grid is denoted by the formula

$$m = \sum_{i=1} C_i,$$

where C_i is the grid cell with index i, having a binary value that indicates whether it is occupied or free. Given the imperfect nature of situations in the real world, where we need to deal with noises in sensors measurement, the occupation grids is approached in a probabilistic way, where $p(C_i)$ represents the probability of the i cell to be occupied. Thus, a probability density function to calculate the probabilities and an optimum estimator to find deterministic map must be used. Detailed explanations regarding the Occupancy grid as well as examples of application may be found in [3].

For this work, the probabilistic occupation grid may be given or built from zero, being updated during the navigation. Noises from odometry and controllers are ignored since the developed algorithm is focused on indoor environments with limited area. Furthermore, there is no need to worry about obtaining a perfect representation of the environment since a technique for the obstacle avoidance is used to prevent collisions. These restrictions make the modeling in occupancy grid carried out with a good precision regarding the environment and without a great computational cost.

B. A* search algorithm

The A* search algorithm originally proposed as a method for finding a minimum cost path in graphs [5] became very popular in the mobile robotics field due the way environments can be represented. When the map is modeled as grids or visibility graphs, it's data structure can be directly used for finding optimal paths. Also, a heuristic function is adopted to guide the search and reduce computational time.

The search begins at one given origin node and expands by its adjacent nodes, that minimizes the function

$$f(n) = g(n) + h(n),$$

where $g(n)$ represents the total cost to of the path from the origin to n and $h(n)$ represents the heuristic function that estimates the cost to go from n to the goal node. Two lists are also kept, the open list and closed list. The first one stores the nodes that have not be computed, and the second one deals with the nodes that have been computed. When analyzing one node, we put its adjacent nodes on the open list. If one of them is already there and the current value is less than what was calculated previously, we update the total cost of that node. The node is then added to the closed list and the search goes on the next node in the open list with the minimum value for $f(n)$. The search will keep going until the open list is empty or a path to the goal node is found. In the second case, the remaining nodes that have a value for $f(n)$ less then the one found in the path will shall be expanded. Figure 1 shows an example of the A* search in an simple grid.

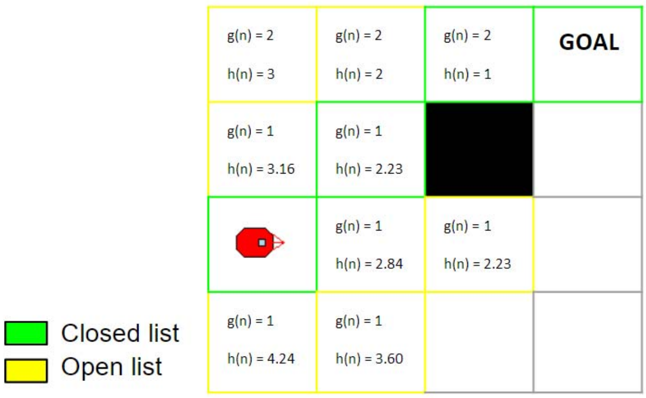


Fig. 1. A* star search.

C. Tangential escape

The method for obstacle avoidance used in this paper is the Tangential escape [6].Given that the focus of the hybrid algorithm is to navigate on structured non mapped environments, this strategy fits to this need. Moreover it is also simple to implement, efficient and computationally low-cost. The method proposes to avoid obstacles by creating a new desired virtual point that will make robot avoid the obstacle by following a path tangential to it. To achieve this, we rotate the real goal by an angle γ, obtained by

$$\gamma = \begin{cases} -\frac{\pi}{2} + \beta - \alpha, & \text{if } \beta \geq 0 \\ +\frac{\pi}{2} + \beta - \alpha, & \text{otherwise} \end{cases} \tag{1}$$

where β is the angle formed by the distance d_{min} to the nearest point of the obstacle and the linear velocity vector. The angle α is the formed between the linear velocity vector and the error vector ρ. Figure 2 illustrates the tangential escape strategy.

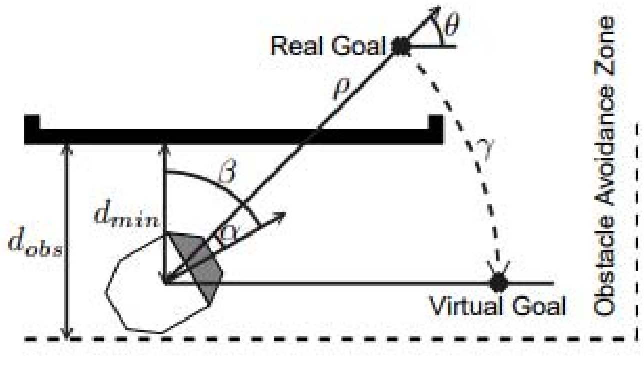


Fig. 2. Tangential Escape.

IV. Hybrid navigation

The basic performance of the proposed algorithm is the following: initially, an Occupancy Grid, either empty or filled, is passed as parameter. The same goes for the variables Xr and Xg which indicate the robots pose and the destination point, respectively. If the supplied grid is empty, we define the robots initial position as origin in the Cartesian plan *Xr* = (0.0) and its orientation is 0. After that, the A* algorithm tries to find a path to Xg. In this step, the cells marked as occupied (if they exist) are dilated in order to create a security zone and accommodate the robots dimensions. Two constants d_{obs_A} and d_{obs_T} are used as distance reference for the tangential escape.

In case of a route is found, a geometric path is generated so that the controller may supply the control signals that take the robot until its destination. While the robot navigate, the sensors feed the Occupancy Grid, after detecting obstacles and free zones. Besides the mapping, a condition is created for when the sensors capture obstacles at a distance shorter than an arbitrarily defined constant d_{obs}. When that happens, the tangential escape algorithm calculates a virtual point *Xv* and the controller starts to have as a target the point Xv. Once the robot either reaches the virtual target, or it is at a distance d bigger than d_{obs}, the robot recoveries the destination point Xd.

After reaching Xd, the algorithm waits to be supplied with a new destination position. From this point on, the more information is obtained in navigations and the better the representation of the environment is done, the better paths are obtained. When A* does not find a path that takes it to the destination (non-mapped areas are defined as obstacles), a reactive strategy is adopted and the controller will simply try to take the robot to the destination point. In this situation, d_{obs} is increased so that the tangential deviation may take place and the robot is able to deviate from obstacles.

Algorithm 1 Hybrid navigation algorithm.

Input: Occupancy Grid OG, position Xr, position Xg
Output: Occupancy Grid OG, position Xr

$route \leftarrow AStar\_search(OG, Xr, Xg)$
if $route\ is\ found$ **then**
  $path \leftarrow Generate\_path(OG, Xr, Xg)$
  $Controller \leftarrow Path\_following\_controller(path)$
  $Dobs \leftarrow Dobs\_A$
else
  $Controller \leftarrow Simple\_error\_based\_controller$
  $Dobs \leftarrow Dobs\_T$
end if
while $Xr \neq Xg$ **do**
  $Xr, M \leftarrow Get\_robot\_data()$
  $Update\_map(OG, M, Xr)$
  if $min(M.sensor\_range) \leq Dobs$ **then**
    $Xv \leftarrow Tangential\_escape$
    $U \leftarrow Controller(Xr, Xv)$
  else
    $U \leftarrow controller(Xr)$
  end if
  $send\_control\_signals$
end while
return OG, Xr

V. Results and Discussion

In the context of this work, where an optimum route is generated for the robot to follow, it is possible that this route may not be represented by a function due to the possibility of generating a non-injective curve. Ensure the the path differentiability is importante in robots navigation, so *Piecewise Hermite Cubic Interpolation*(*PHCIP*) is applied to guarantee such condition.

Other interpolation methods take into consideration only the points' coordinates to generate curves. In contrast, (*PHCIP*) uses the points' coordinates to determine a sequence of third degree polynomials connecting them, taking into account the smoothness of two consecutive curves. Such approach avoids non-injective curves since it fragments the path in differentiable curves and forces the continuity of the derivate in the polynomials exchange.

After having the optimum path points generated and several third degree polynomials produced by *PHCIP*, it is necessary some algorithm that makes the robot navigate following the path. For that, a simple pursuit method is used. It creates a desired virtual point at a constant distance and as the distance between the robot and the this virtual point decreases, the point is updated to the next one in the path.

In order to validate the proposed strategy, the simulation are performed in *Matlab* and *MobileSim* environment. They can provide numerical simulations in non-mapped structured environments that approach to real situations. The algorithm is tested on the environment illustrated in Figure 3. The robot model used during the tests is the Pioneer-3DX, equipped with 8 sonar-range sensors.

The results of generated paths, the covered distance and the time spent to go through them are presented and discussed in the following sections.

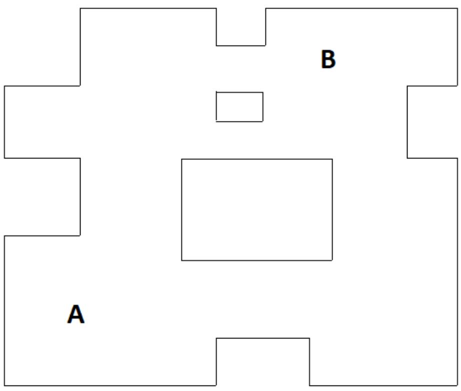


Fig. 3. The simulation environment

.

A. Simulations

The simulation that was carried out had as a purpose to make the robot move from point A (defined as origin in the Cartesian plane) until point B. Then, the robot should return to point A, this time using the information that were obtained during the departure. Finally, the robot moves again to point B, so that it is possible to compare the improvement in relation to the first departure.

In order to make the first movement, the robot did not have any information regarding the environment. Thus, it adopted a strictly reactive posture, avoiding obstacles and walls. The path covered by the robot and the obtained map is shown in Figures 4(a), 4(b) and 4(c).

When the return from point B to point A was performed, the robot had enough information to find a path in algorithm A*. Thus, a path was traced so that the robot could move. The observable area to make the tangential deviation is decreased to avoid that the robot makes unnecessary deviations, as showed in Figures 4(d), 4(e) and 4(f).

Finally, the robot had as a purpose to return to point B, this time with the information accumulated during the two first navigations. A new path is traced, optimizing the navigation. The data are shown in Figures 4(g), 4(h) and 4(i).

In addition to the paths and curves, data related to the distance and time spent in each route are presented in Table I.

TABLE I
NAVIGATION DATA

Goal	**Grid cells**	**Distance (meters)**	**Time (s)**
A to B	42	17.44	84.68
B to A	37	15.61	71.69
second A to B	32	15.56	69.83

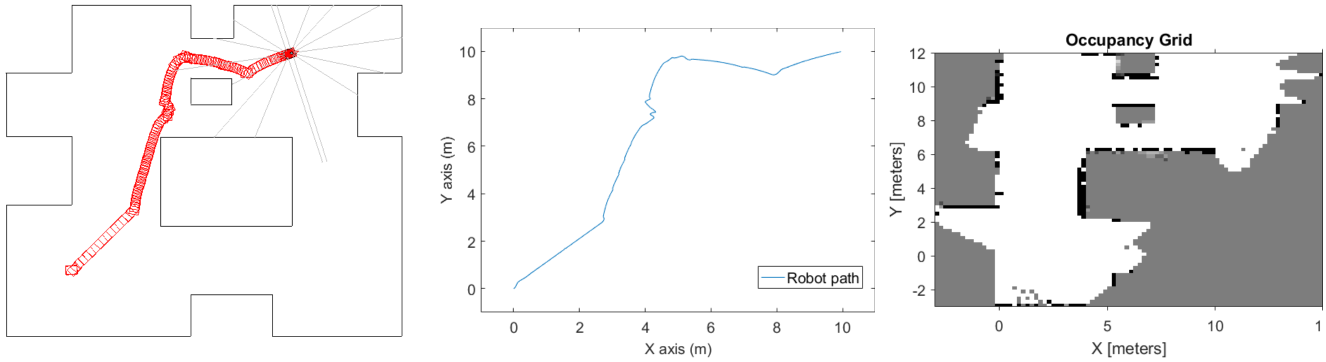
B. Discussion

The results show that the robot can improve its navigation significantly when there is enough information to plan a path. We can perceive that the robot has difficulties to perform the tangential escape due to the reduced capacity of perception (there are only 8 distance sensors), so that the deviations are performed near the obstacles and walls. In the first route from A to B where there are no information from the map a priori, the robot goes in a path "above" (in the Cartesian plane) obstacle 1, since it does not follow a defined path. In addition, the observable limit d_{obs} is enlarged in this situation.

On the second and third trajectories, the robot follows the traced path, but it is forced to perform a small deviation near obstacle 1. After overcoming the obstacles, the robot returns to the planned path. In both cases, we observed that the robot slightly leaves the planned path until it returns. This is caused by the use of the follow-up algorithm of the used path, which supplies curve points as target at a constant distance from the robot.

VI. Concluding Remarks

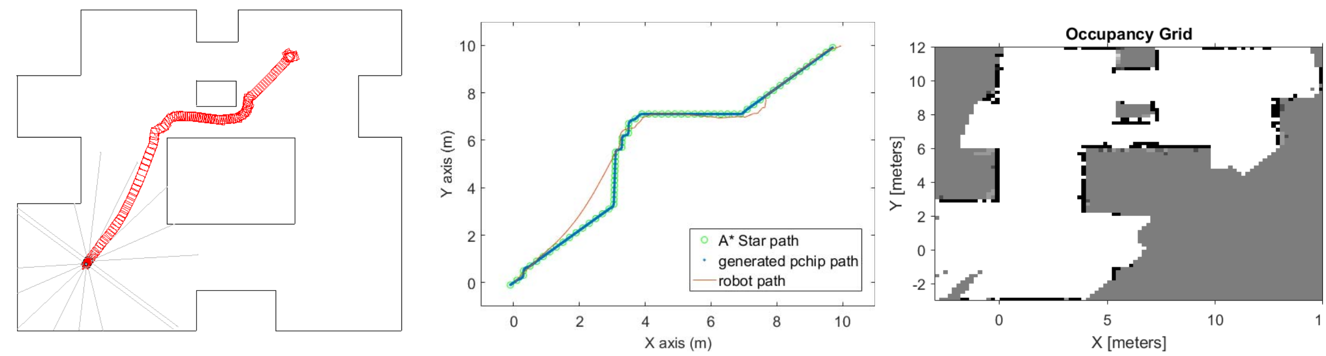
This paper present a strategy to navigate in indoor environments, mapped a priori or not. Even if the base algorithms have already been described in the literature, our method showed a new way of combining the advantages of both local and global planning for finding optimal routes and avoid obstacles in real



(a) The robot's movement trail from A to B.

(b) The robot's position graph.

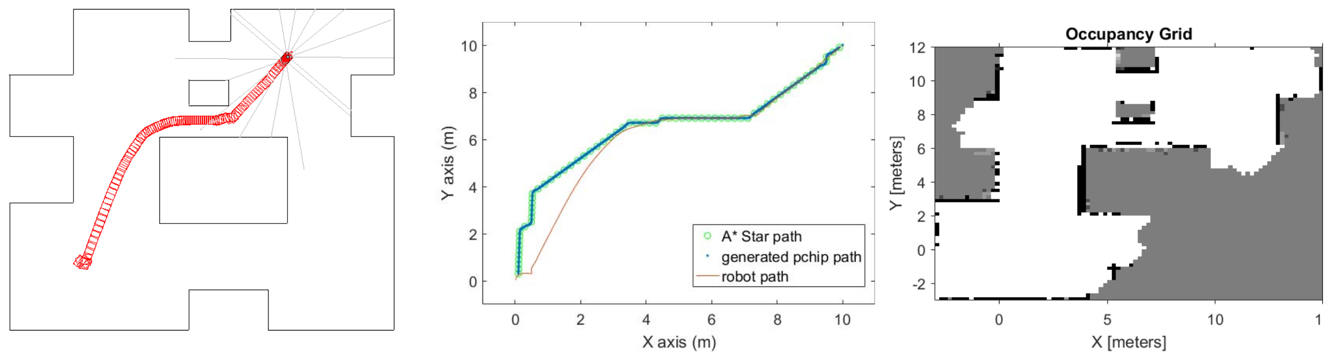
(c) The Occupancy Grid obtained at the moment that the goal is reached.



(d) The robot's movement trail from B to A.

(e) The route found, the path calculated by pchip and the robot's position graph.

(f) The Occupancy Grid obtained at the moment that the goal is reached.



(g) The robot's movement trail from A to B.

(h) The route found, the path calculated by pchip and the robot's position graph.

(i) The Occupancy Grid obtained at the moment that the goal is reached.

Fig. 4. Numerical simulations.

time, overcoming their original limitations. The contribution takes place through the way the techniques were integrated into a new method. The tests that were carried out show that it is possible to navigate in an efficient way and avoid unknown obstacles, even with limited sensing resources. Future works may be done performing experimental tests and comparing the hybrid algorithm with popular global planning approaches and also analyzing different controllers and environments with mobile obstacles.

Acknowledgments

The authors thank Federal University of Viçosa and Department of Electrical Engineering and Department of Informatics for all the support and infrastructure provided for the execution of this project. They also thank CNPq and FAPEMIG for their financial support. Mr. Carvalho also thanks CAPES for the scholarship.

REFERENCES

[1] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, p. 463497, 2015.

[2] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[3] S. Thrun, "Exploring artificial intelligence in the new millennium," G. Lakemeyer and B. Nebel, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, ch. Robotic Mapping: A Survey, pp. 1–35.

[4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[5] P. E. Hart, N. J. Nilsson, and B. Raphael, ""a formal basis for the heuristic determination of minimum cost paths"," *SIGART Bull.*, no. 37, pp. 28–29, Dec. 1972.

[6] A. S. Brandao, M. Sarcinelli-Filho, and R. Carelli, "An analytical approach to avoid obstacles in mobile robot navigation," *International Journal of Advanced Robotic Systems*, vol. 10, no. 6, p. 278, 2013.

[7] C. Ray, F. Mondada, and R. Siegwart, "What do people expect from robots?" in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 3816–3821.

[8] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 1014–1019 vol.1.

[9] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, Apr 1996, pp. 113–120 vol.1.

[10] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[12] ——, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5041–5047.

[13] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, "Ant colony optimization algorithm for robot path planning," in *2010 International Conference On Computer Design and Applications*, vol. 3, June 2010, pp. V3–436–V3–440.

[14] L. Juntao, D. Tingting, L. Yuanyuan, and H. Yan, "Study on robot path collision avoidance planning based on the improved ant colony algorithm," in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 02, Aug 2016, pp. 540–544.

[15] S. Alnasser and H. Bennaceur, "An efficient genetic algorithm for the global robot path planning problem," in *2016 Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, July 2016, pp. 97–102.

[16] P. K. Das, B. M. Sahoo, H. S. Behera, and S. Vashisht, "An improved particle swarm optimization for multi-robot path planning," in *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, Feb 2016, pp. 97–106.

[17] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 3310–3317 vol.4.

[18] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," *Artif. Intell.*, vol. 172, no. 14, pp. 1613–1643, Sep. 2008.

[19] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 1398–1404 vol.2.

[20] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," in *2012 IEEE International Conference on Mechatronics and Automation*, Aug 2012, pp. 1227–1232.

[21] O. Montiel, R. Sepúlveda, and U. Orozco-Rosas, "Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field," *Journal of Intelligent & Robotic Systems*, vol. 79, no. 2, p. 237, 2015.

[22] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5177–5191, 2015.

[23] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun 1991.