Global Journal of Computer Sciences: Theory and Research

Volume 7, Issue 3, (2017) 123-128

www.gjcs.eu

A dynamic path planning method for wheeled mobile robots (Dyna-bug)

Suat Karakaya*, Mechatronics Engineering Department, Kocaeli University–Umuttepe, Kocaeli 41380, Turkey.
Gurkan Kucukyildiz, Mechatronics Engineering Department, Kocaeli University–Umuttepe, Kocaeli 41380, Turkey.
Hasan Ocak, Mechatronics Engineering Department, Kocaeli University–Umuttepe, Kocaeli 41380, Turkey.

Suggested Citation:

Karakaya, S., Kucukyildiz, G. & Ocak, H. (2017). A dynamic path planning method for wheeled mobile robots (Dyna-bug). *Global Journal of Computer Sciences: Theory and Research. 7*(3), 123-128.

Received July 20, 2017; revised October 11, 2017; accepted December 4, 2017.
Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, North Cyprus.

Abstract

In this study, a hybrid path-planning scheme is presented. The main contribution of this paper is merging the static grid costs of the global map and the immediate environmental structure of the local map. The stationary condition of the map and the instant local goal is weighted by certain coefficients in order to determine the next move of the wheeled mobile robot (WMR). Thus, the cost function is defined in terms of the grid costs and the dynamic parameters. The main assumption is that the WMR on which this scheme is executed must be equipped with a field scanning sensor. The sensor readings in each processing cycle are pre-processed before plugging in the cost function. The passages in the local map are extracted from the sensor data, then the optimal collision-free point lying on the passages is obtained via the cost function.

Keywords: Path planning, collision avoidance, mobile robot.

* ADDRESS FOR CORRESPONDENCE: **Suat Karakaya,** Mechatronics Engineering Department, Kocaeli University–Umuttepe, Kocaeli 41380, Turkey. *E-mail address*: suatronik@gmail.com / Tel: +90 (262) 303 10 00

1. Introduction

The majority of path-planning studies are focused on a fully known map before navigating the wheeled mobile robot (WMR). This approach may lead to the lack of ability to avoid collisions when the algorithm is implemented on a dynamically changing environment. On the other hand, barely sensor-based local path-planning schemes do not guarantee to reach the global target, as such an approach makes the next move regarding only the immediate window of the WMR's sight; therefore, the WMR does not have any prior knowledge about the workspace [1]. Grid-based global planners determine the static trajectory from the cell with a higher cost to the lower grid region [2]. This strategy is benefitted in various schemes such as distance transform (DT), hill-climbing and potential-fields [3, 4]. The cost depends on the Euclidian, city-block and neighbourhood accessibility parameters in DT. Although these approaches are complete and conventional, various modifications will improve the performance of the planner.

The path planners can be categorized according to the map type, searching characteristics and completeness. The properties can be sub-categorized as heuristic or complete, global or local and dynamic or static. The heuristic approaches do not guarantee the best solution in a solution space, where the complete methods find the best solution according to a certain performance criterion. Heuristic models offer optimal solutions close to the best with less complexity and computational cost. Complete approaches mostly need more capacity for computation in order to achieve the global optimum [5]. The global path planners benefit from complete information about the global map, whereas the local path planners need immediate sensor readings. Most global path planners need prior information about the whole map. Local path planners read the status of the environment and decide the next move according to the local frame. The static path-planning methods refer to the environment that is populated with only stationary obstacles and non-moving objects. The dynamic path planners encompass moving obstacles and changing environments. In [6], three main issues of the mobile robot motion planning process are proposed, namely mapping, path planning and path tracking. Although these classifications provide a systematic way to develop a path-planning algorithm, a fixed sub-tasking in path planning may not be always feasible. For example, bug-0 algorithm [7] does not need to extract a model of the environment or traverse an already planned trajectory. On the contrary, simultaneous localization and mapping (SLAM)-based navigation methods perform mapping, path planning and path traversal simultaneously [8]. The mobile robot determines the current position with respect to custom landmarks. SLAM addresses the problem of non-existence of any external positioning system and it considers the mobile robot to have the ability to discover its environment while building a map and moving on an optimal trajectory.

This paper presents a new bug-type algorithm, namely, dyna-bug, which exploits both global and local information about the mobile robot's environment. The main contribution of this paper is a novel switching condition between the path traversal and motion-to-goal movement. The static environment is mapped by constructing the LIDAR data at certain intervals. Similar to tangent bug, the proposed algorithm searches the local environment to detect an immediate target; it also reconstructs a static trajectory based on the current map when the robot traps into local minima. In this way, repeatedly traversing a region is avoided. The algorithm also provides a choice to make use of static map information. The DT algorithm is initially implemented based on the initially given static obstacle configuration; then dyna-bug, the real-time navigation scheme, is implemented to the pre-processed configuration space. The distance costs of the grid map are evaluated while making local target estimation. The initial stationary map data are updated while the robot moves around the environment for further improvement in accuracy of static path-tracking phase. The data are used for further improvement of the global trajectory. A global map is created as the robot moves, and the mobile robot exploits this map to determine the static global trajectory. The robot tracks the trajectory until certain events occur. This hybrid approach avoids trapping of local minima. Significant improvement in the path-planning performance is achieved.

1.1. Path-Planning Phase

The initial step is the path planning phase, which generates a stationary trajectory based on the given obstacle configuration. If the environment is totally unknown, then a direct path towards the goal will be generated as if there was no obstacle. On the other hand, Euclidian DT (EDT) algorithm [9] is implemented to the partially or fully known map in order to achieve initial steering costs of the grids on the map. The static trajectory is then calculated based on the current EDT costs. Starting from the given initial position, the adjacent grid cell with the minimum cost is determined continuously until reaching the goal point. In this way, a static trajectory is generated in a greedy-descent manner. Then it is checked for whether there is a direct or obstacle-free access towards the goal within the range sensor. The robot is moved directly to the goal without determining any local targets or passable gaps, whenever the goal is visible by the robot. In other cases, the algorithm switches path-tracking or motion-to-goal modes to converge the goal point. Given the start and goal points on a sample grid map, the static trajectory generated by EDT is shown in Figure 1, where the filled grids correspond to the occupied region and the remaining cells are the free space. Assuming that points a and b are picked as the start and the goal locations, the EDT algorithm calculates a reference path based on the current stationary conditions.

1.2. Path-Planning Phase

Path tracking is basically a local target determination method like the motion-to-goal scheme. Following the initial path-planning phase, the algorithm receives a dataset from the range sensor (LIDAR). LIDAR data provide the distances of the obstacles within a finite range: view angle and angular resolution. The algorithm initially searches the goal point in the scanning field of LIDAR. If the goal point is not visible by the robot within the scanning field, the robot is led to keep track of the initially planned trajectory. Since the robot cannot ‘see’ the global target (goal), it is to be converged to the goal by calculating the local targets at each cycle of the algorithm. The local target detected in path-tracking mode must be on the planned path; therefore, the static path points inside a particular region are examined. A virtual circle with a certain radius (rL), centred at the robot’s position, namely, look-ahead circle (LAC), is defined to sense the tendency of the static path in a local area. LAC is actually a semi-circle as it is defined only in the half-plane through which the robot is currently heading. The intersection point(s) of the static trajectory and LAC is detected and the point that has the minimum DT cost is assumed to be the candidate local target. This point is accepted as the local target if it is visible by the robot. Otherwise, path-tracking mode is interrupted by the other mode, motion-to-goal.

1.3. Motion-to-Goal Phase

The motion-to-goal mode encompasses multiple tasks such as passable gap detection, local target determination and local minima supervision. This scheme initially checks whether there is at least one passable gap within the LIDAR scanning range. If it exists, the gap is also checked if it is physically large enough to steer the robot. The passable gaps are then reorganized, depending on the passage priorities. The points on the line segments between the corners of the final gaps are given to a cost function to obtain the optimal target. The current local target is assumed to be the point that minimizes the cost function. The motion-to-goal mode depends on the existence of passable gaps in the local frame. Therefore, a local target determination is not possible in the absence of any gaps. The path-tracking mode also cannot offer a local target at this stage, as the intersection point is already invisible. Such a situation, where the intersection point is invisible and there is no gap available in the local frame, means that the robot has tapped into a local minimum. The distance vector is saturated with the maximum measurement range (R) of the LIDAR. For a particular $\mathbf{x}_r$ and measurement angle (α), sensor data model (m($\mathbf{x}_r$, α)) is expressed as

$$m(\mathbf{x_r},\alpha)=\min_{\gamma\in[0,R]}\left(\mathbf{x_r},\mathbf{x_r}+\gamma\begin{bmatrix}cos(\alpha)\\ sin(\alpha)\end{bmatrix}\right)\Leftrightarrow\left(\mathbf{x_r},\mathbf{x_r}+\gamma\begin{bmatrix}cos(\alpha)\\ sin(\alpha)\end{bmatrix}\right)\in\bigcup_k O_k \qquad (1)$$

The elements of the measurement array, which do not correspond an occupied region, are saturated with the maximum range of the sensor (R)

$$m(\mathbf{x_r},\alpha)=\begin{cases}m(\mathbf{x_r},\alpha), & if\ m(\mathbf{x_r},\alpha)<R\\ R, & else\end{cases} \qquad (2)$$

2. Results

An illustrative demonstration of path-tracking mode is given in Figures 2 and 3. The scenario proves that the algorithm can benefit from prior knowledge about the map. As the environment is not completely known, the static trajectory crosses the unknown obstacles. The intersection points of the static path and LAC are sorted based on the EDT scores. The intersection grid that has the minimum EDT score is assigned as the candidate local target on the static trajectory. Only the intersection points, which are in the half-circular plane through the robot's heading, are taken into account while determining the current local target point. It can be seen that both the intersection points are inside the scanning range and the minimum EDT scored one is determined as the candidate local target. Then the point is checked to see whether it is visible. If the minimum EDT scored intersection point is visible, the robot is steered to the target's location. The robot keeps tracking the static path via the local targets until approximating to an unknown obstacle closer than the radius of LAC. The scenario shown in Figure 3 represents that the intersection point lies on an unknown obstacle, which means path-tracking mode is to be interrupted by motion-to-goal scheme.

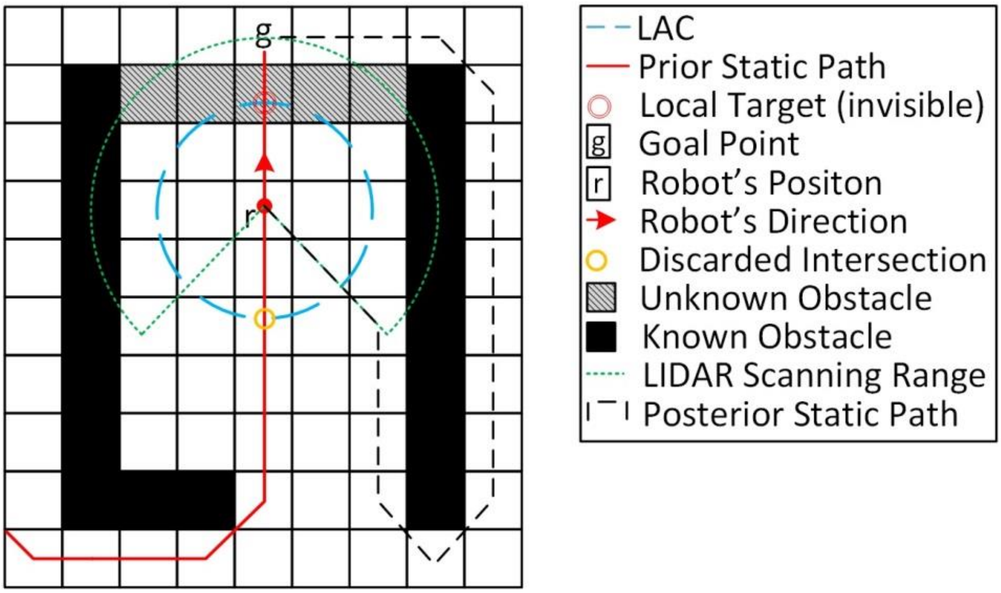

Figure 1. Sample scenario for trapping into local minimum

In Figure 1, it can be seen that the intersection point on the direction of the robot is invisible and there is no passable gap within the LIDAR range. In order to overcome this problem, prior knowledge about the indoor map is updated with the recent LIDAR data. Then the path-planning algorithm is applied to the most recent static map, and the path-tracking mode is restarted in the new trajectory (reference headings are in bold but have no numbers).

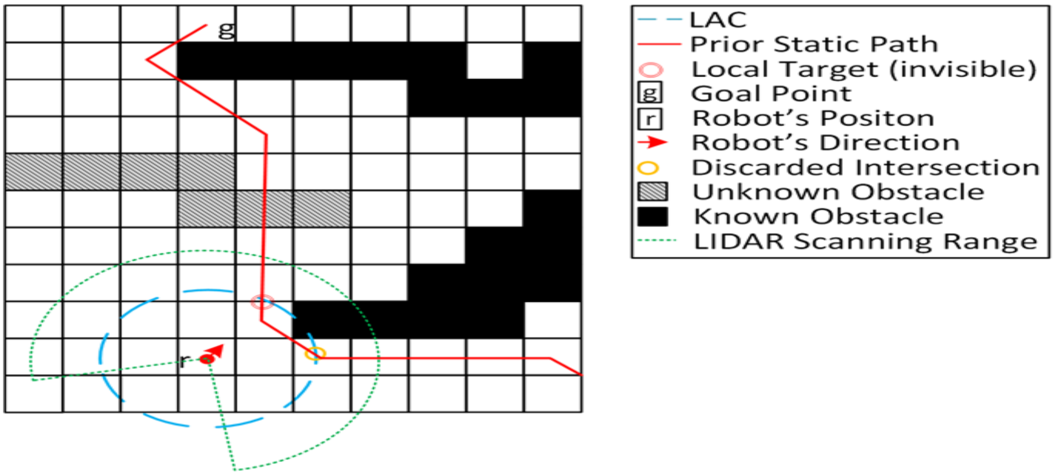

Figure 2. Path-tracking demonstration on a sample grid map

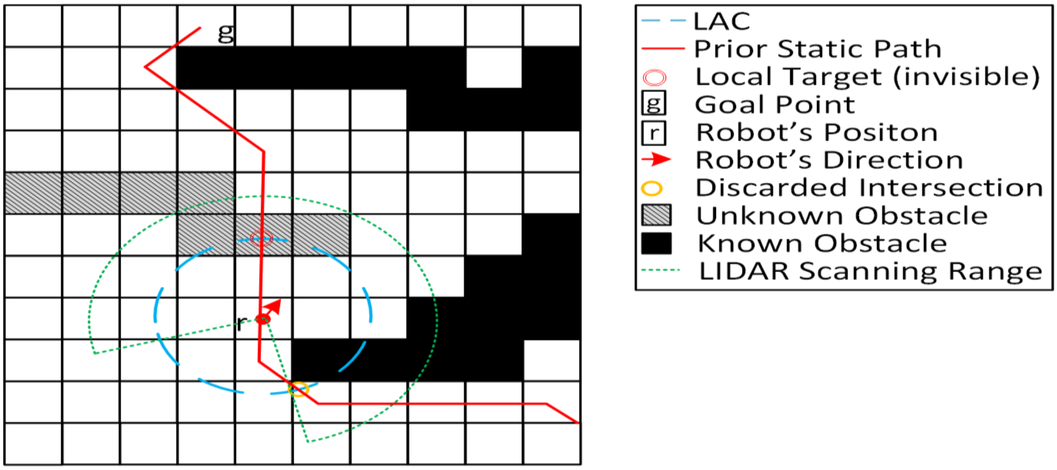

Figure 3. Sample scenario for switching motion-to-goal mode

References

[1] Behnke, S. (2006). Local multiresolution path planning. In *7th RoboCup International Symposium*, (pp. 101-110).

[2] Buniyamin, N., Wan Ngah, W. A. J., Sariff, N., & Mohamad, Z. (2011). A simple local path planning algorithm for autonomous mobile robots. *International Journal of Systems Applications, Engineering & Development, 5*(2), 151-159.

[3] Daokui, Q., & Du Zhenjun, X. D. (2008). Study on path planning method of mobile robot. *Robotics, 30,* 97-102.

[4] Elizondo-Leal, J. C., Parra-Gonzalez, E. F., & Ramirez-Torres, J. G. (2013). The exact Euclidean distance transform: a new algorithm for universal path planning. *International Journal of Advanced Robotic Systems, 10*(6), 266.

[5] Huang, S., & Dissanayake, G. (2006). Convergence analysis for extended Kalman filter based SLAM. In *Robotics and Automation ICRA Proceedings 2006 IEEE International Conference* (pp. 412-417). IEEE.

[6] Jarvis, R. A. (1985). Collision-free trajectory planning using distance transforms: Transactions of the Institution of Engineers, Australia. *Mechanical Engineering, 10*(3), 187-191.

[7] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research, 5*(1), 90-98.

[8] Latombe, J. X. (1991). *Robot motion planning*. Norwell: Kluwer Academic.

[9] Lumelsky, V., & Stepanov, A. (1986). Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Transactions on Automatic control, 31*(11), 1058-1063.