Moving Obstacle Avoidance for Mobile Robot Moving on Designated Path

Taichi Yamada[1], Yeow Li Sa[1] and Akihisa Ohya[1]
[1]Graduate School of Systems and Information Engineering, University of Tsukuba,
1-1-1, Tennoudai, Tsukuba City, Ibaraki Pref., 305-8573, Japan
(Tel : +81-29-853-5155; E-mail: tyamada,lsyeow,ohya@roboken.esys.tsukuba.ac.jp)

Abstract - In reality, it is necessary to define the accessible area of the robot in an environment which may consist of door entrance, descending stairways and others. In this paper, a method to closely follow the global designated path by setting manually. In this method, the robot moves in a range of velocities and thus being able to avoid obstacle staying closely to the designated path. The results of the experiments verify that this method is able to let the robot move closely to the designated path while avoiding moving obstacle.

Keywords – Mobile Robot, Moving Obstacle Avoidance, Path Planning, Laser Scanner.

1. Introduction

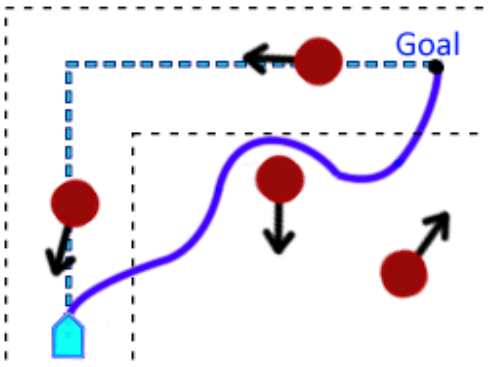
1.1 Robot Navigation

For mobile robot navigation, it is a necessity to travel to a destination while avoiding collision amidst obstacles [1]. Generally, a robot needs path planning and obstacle avoidance to move from start point to goal point safely. There are many well researched methods developed for static obstacle avoidance, namely potential fields [2], grid-based methods [3] and dynamic windows approach [4]. In recent years, these static obstacle avoidance methods are then adapted to achieve moving obstacle avoidance as well [5,6,7,8].

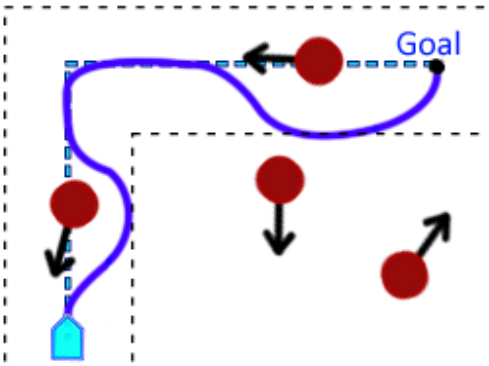
Besides, planning methods proposed by Tsubouchi et al.[9] and Sakahara et al.[10] consider the time dimension and actively avoid moving obstacles by predicting their movement depending on current velocities. The former utilized geometric approach while the latter applied random searching to generate the shortest collision free path towards goal.

Yet, these approaches do not follow a designated path as they generate detours for obstacle avoidance as seen in Figure 1a. It actually crucial for robot to move within its movable space considering that the environment may have prohibited or dangerous zone for robots such as door way (robot may run into opening door) and descending stairway. It may be better for robot to stay close to its designated path to avoid complicated recalculation.

Besides, obstacle that are coming from behind the robot with a higher velocity are not put into consideration, and the robot moves in a constant speed during all time. It is common having collision from sideways, which is the blind spot of the sensor which detects only a limited range of area focusing on the front of robot. In order to keep the robot in original path, it is natural to travel with variable velocities on the designated path, just like a human slows down to let a person cross in front of him and then resumes on walking, and to walk faster when someone is following closely behind to avoid collision.



(a) path generated does not follow a designated path



(b) path generated following a designated path

Fig 1. Collision Avoidance

1.2 Problem Statement and Purpose of Research

In this research, the robot is to follow a designated path, and to avoid moving obstacle when found while staying within the path boundary. The robot is required to make a new avoidance path to prevent collision with the obstacle and return back to the designated path.

Thus, in this research, the robot follows the global designated path closely by heading to subgoals on the path. If there is an obstacle, the robot is to make a decision to avoid the obstacle while staying within the path boundary of the designated path (as seen in Figure.1b) as much as possible. It is proposed that obstacle avoidance can be achieve with alteration of Tsubouchi et al.[9]'s method to let the robot generate avoidance path which can be traveled with various velocities. This method is preferred to Sakahara et al.[10]'s method as it comparatively generate path with less randomness.

In order to avoid obstacles, a robot requires a sensor to perceive its surrounding environment, Laser scanner is chosen in this research as it has high accuracy and wide detection angle. Yet, due to the currently commercialized laser range sensors having less than 360 degree of scanning range, a method is proposed to detect obstacle using 2 laser scanners. With this method, obstacles not only from front , but from behind and sideways of the robot can be detected as well.

As the research setting, a non-holonomic robot mounted with 2 laser scanners having 240 degrees scanning area (on in the front and on at the back of the robot so as to provide 360 degree view of the environment)

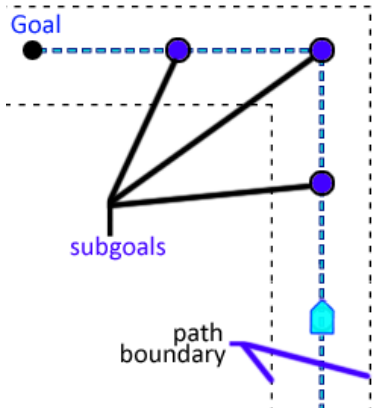


Fig 2. Subgoals and Path of a Designated Path

is to detect obstacles coming from front back and sideways.

In the next section, the method how the robot follows the designated path is described. Next, the enhancement on local path generation algorithm (for moving obstacle avoidance) is elaborated. The results of designated path following method verified with simulation and the path implementation of the avoidance algorithm is verified with experiment will be presented in the subsequent section. Lastly, future works is discussed and than the paper ends with the conclusion of the research.

2. Following the Global Designated Path

In this section, the method of making the robot follow the designated path is elaborated. In Figure.2, a robot is to move to current subgoal along the designated path (blue dotted line path) and when reaches the subgoal, robot move to a next subgoal. The subgoals that connects the line path and path boundary that limits the reachable area of the robot are set manually so the robot can travel closely to the goal along the designated path.

First, the robot is to move along the path by heading to its subgoals on the path. When there is no obstacle detected, the robot moves towards the current subgoal, and proceed to the next when the current subgoal is within a certain distance less than the threshold value.

If there is an obstacle detected, then the robot is to keep itself within the path boundary as much as possible to avoid the obstacle. In the case when the next subgoal corrects to a line path which requires change of robot orientation, it is desirable for the robot to move straight till the end of the junction then only to turn its orientation towards the goal point.

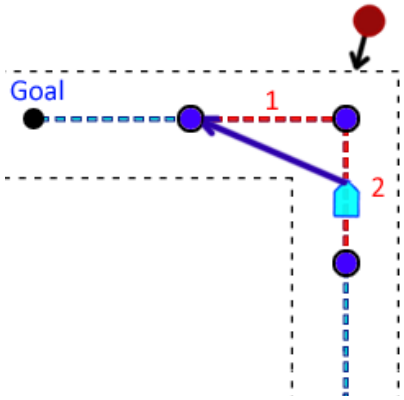


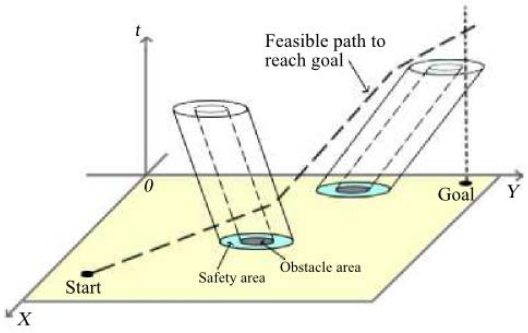
Fig 3. Obstacle Avoidance at Subgoal

When an obstacle is too close to the subgoal shown at Figure.3, the robot may not reach at the current subgoal. Hence the robot will give up making obstacle avoidance path to reach the current subgoal, when the robot is close to subgoal than the threshold value. Instead, the robot bypass it and move towards the next subgoal.

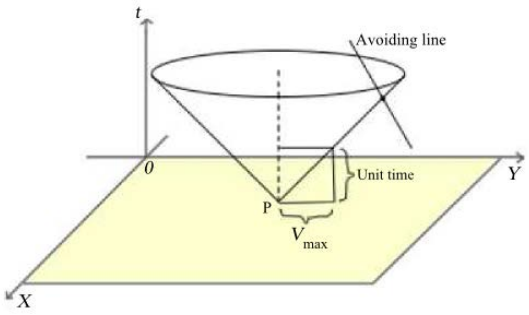
3. Local Path Planning with Variable Velocity

In the local path generation that is described in [9]. The configuration space of the forecast algorithm is shown in Figure 4a, whereby *x* and *y* coordinate denotes the position of robot and obstacles in the planar space, while the vertical axis t denotes the time coordinate. Therefore, a moving obstacle can be represented by oblique cylinder, and the obliqueness denotes their moving velocity. With this, assuming the obstacle moves in constant velocity, a robot can avoid moving obstacles by knowing the forecast of their location based on their current velocity. This path planning is a graph search to find an optimum path connecting the start point and goal point without intersection of obstacle.

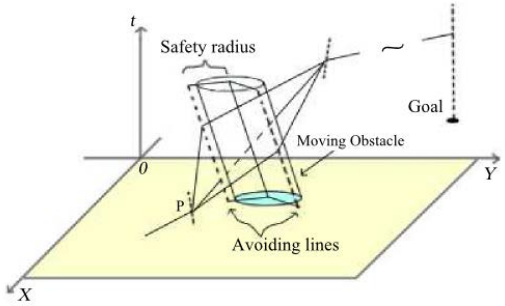
In the algorithm, reachable cone (the area a robot can travel within its maximum velocity as in Figure 4b) and avoiding line (a line of position points which the robot can travel to while avoiding the obstacle represented as a cylinder in Figure 4c, and resume back to its original path to reach its goal point) are used to plan the optimum local path. The avoiding line is a parallel line touching the surface of cylinder looking from the robot position point *p*.



(a)Time-space trajectory planning



(b) Intersection point between reachable cone and avoiding line



(c) Moving obstacle and avoiding line

Fig 4. Local Path Planning

In this scheme, the intersection of reachable cone and avoiding line is the via point to detour around the obstacle.

However This may not be the optimum path for the robot that moving on designated path and it may run out of the boundary too. It is proposed that the robot would avoid the obstacle by changing its velocity to follow the designated path more closely. There might be a chance that the path generated may stay closer to the designated path as the robot can avoid obstacle by speeding up or slowing down. With this belief, we modify the path generation algorithm to generate path for various robot velocities with procedure as such:

1. If the robot path is not crossed by any obstacle in the configuration space, the straight line is the optimum path.
2. If it crosses a cylinder of obstacle, place the apex of the reachable cone at point *p*, and find the closest intersection point of the reachable cone and avoiding line. This will be the via point to avoid the obstacle using robot's current velocity.
3. Then repeat the local path generation algorithm using velocities 0.1m/s, 0.2m/s, 0.3m/s, 0.4m/s, 0.5m/s respectively and
4. The path that have a via point out boundary is removed. The most optimum path is chosen from among those remaining based on the following criteria referring to Figure.5:
   a. Total distance of the path (*dist1* + *dist2*)
   b. Time to travel the total distance
   c. Difference in velocity form normal velocity
   d. Angle of robot heading and robot via point direction, θ
5. These steps are repeated recursively as the robot moves until it reaches the goal.

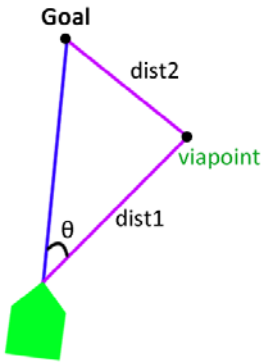


Fig 5. Angle Between Robot Heading and Robot Via-Point Direction

$$E = k_1 \times Dist + k_2 \times Time + k_3 \times vel + k_4 \times \theta \quad (1)$$

The parameter values k1 to k4 of best path evaluation equation *E* are determined by a design of navigation. For example, k2 is given a larger value to arrive at the goal early, or k4 is given a larger value to move close to the designated path.

The path with the least value will be chosen as the best avoidance path. With this method integrated with the global path following, it can avoid moving obstacle while being within the path boundary.

4. Experiment and Result

In this section experiment to verify if the robot can avoid collision while following the designated path closely and stay within the path boundary. First the robot hardware configuration is described. Then the experiment and result will be shown next.

4.1 Hardware System Configuration

In this research, in order to realize obstacle avoidance, the self-contained differential drive robot (Figure.6) is used as the mobile robot. It has DC motor driver circuits with built-in encoders and the interface circuits Hitachi

SH-2 boards called "B-LOCO". This B-LOCO board provides robot locomotion and odometry-based

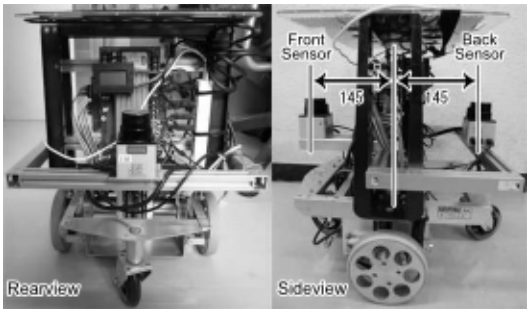


Fig 6. Mobile Robot Platform

self-position estimation functions. Thus this allows precise robot mobility control by defining arc and line path to follow at the user's end.

As for the external sensor, a laser range sensor called URG-04LX or known as classic URG from Hokuyo Automatic Co., Ltd. is being used. It has an optimum point distance detection of 4 meters with scanning range of 240 degrees. For this research, in order to get the surrounding environment of robot in all directions, an additional sensor is mounted on the back of the robot. Both of the sensors are 0.145 meter away from the central robot axis as shown in Figure.6, providing 360 degrees view of the surrounding environment.

A HP Pavilion dm3 notebook computer located on top of the robot platform is connected to the B-LOCO board via serial communication cable, and to the URG sensor via USB cable. The PC, serving as the principle controller, holds the user program, which processes the environment data from URG sensor and sends motion commands to the B-LOCO board. The hardware overview is as shown in Figure.7.

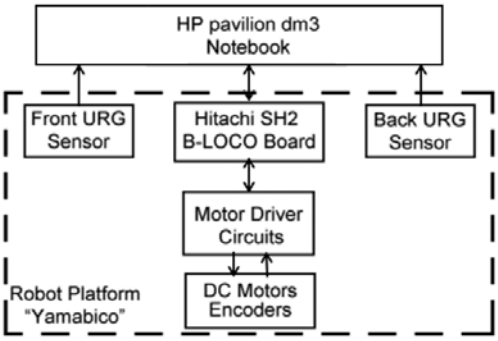


Fig 7. Hardware overview

4.2 Experiment

An experiment is performed to test the effectiveness of the calculation of local path for obstacle avoidance. The parameter values k1 to k4 of best path evaluation equation *E* is being decided with k1 = 0.1, k2 = 0.001, k3 = 0.08, k4 = 0.3 based on trial and error. The robot is set in a wide space environment with one mobile robot as the moving obstacle, and there are no boundaries or other static obstacles. The robot is to move forward with a constant velocity *v* = 0.2m/s towards the goal five meters ahead. The obstacle is set with a constant velocity v = 0.3m/s crossing the robot path from front, back, and side of the robot. The setting can be seen in Figure.8, where the polygon indicates the robot position, and the circle as the obstacle, and the arrow as the obstacle's moving direction. In this experiment, the robot moves towards the goal at coordinate (5.0, 0) in a straight line if there is no collision with obstacle detected. The calculation of local path is triggered as soon as a moving obstacle which has a moving line path crossing the robot path is detected. The robot moves within the path boundary (set at 1m left and right of the designated path line) towards the via point of local path with appropriate velocity and then to the goal point. This is repeatedly executed throughout the course reaching the goal.

4.3 Result

The robot is able to avoid moving obstacle with constant velocity from the back, front and side without collision. The robot position, velocity and the obstacle position are plotted in Figure.9, in which the 'x' marks

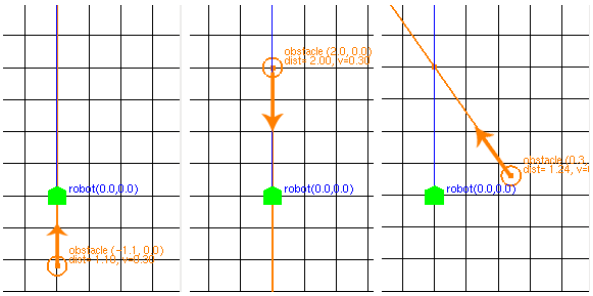


Fig 8. Setting of moving obstacle (left): 1m behind the robot (center): 2m in front of robot (right): 1m right side of robot

the via points of avoiding path created, '*' marks the robot moving towards the via point, 'square' marks the robot moving to goal and 'circle' marks the obstacle position. The increment of the size of the robot path markings is proportion to the change of velocity of the robot.

When the robot detects an obstacle coming from its behind, the locomotion of the robot is plotted in Figure.9a, and it path when the obstacle is behind the robot. When obstacle comes from front, plot.12b shows that the robot accelerated while turning left to avoid the obstacle coming from ahead. As for obstacle from side of robot, the plot in Figure.9c shows the robot slowed down and let the obstacle from the side pass before continuing on the designation path with original velocity. With this result, it is verified that the local path can avoid moving obstacle well and it could be applied to let the robot reach its subgoals efficiently.

5. Future Works and Conclusion

5.1 Conclusion

In this paper, we have presented method for which a robot can avoid moving obstacle while staying in a designated path. First, we have presented how a robot could follow a designated path closely by setting subgoals. As the robot is moving on the designated path, it reacts to moving obstacle with consideration to the designated path.

Next, the obstacle avoidance algorithm which is based on Tsubouchi et al.[9]'s method is described. It is modified to make motion planning considering obstacle from all directions, and robot moves with several choices of velocities. The path with the best evaluation (based on the distance, time of travel, velocity used etc) is chosen. This way, the robot can move with less deviation from the designated path.

The experiment results of the local path generation verifies that the robot can avoid obstacle moving from front, back and sideways of the robot while staying close to the designated path and being within the path boundary.

5.2 Future Works

The combination of the global path following and the local path for moving obstacle avoidance should be executed in the mobile robot. The robot's motion limitation is not considered in this local path generation, so it should be implemented to generate trajectory path which is smoother. And the determination of parameter k1 to k4 of best path evaluation equation *E* should be discussed. These parameter is important factor, because the robot motion is directly depend on these parameter.

References

[1] J.C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, Boston, 1991

[2] O. Khatib, "Real Time Obstacle Avoidance for Manipulators and Mobile Robot", International Journal of Robotics Research, Vol.5 No1, pp 90-98, 1986

[3] A. Elfes, "Using occupancy grids for mobile robot perception and navigation", IEEE Computer, 6, pp 4657, 1989.

[4] D. Fox, W. Burgardm S. Thrun, "The dynamic window approach to collision avoidance", IEEE Robotics and Automation Magazine, Vol4-1, pp. 23-33, 1997.

[5] S.S. Ge, Y.J. Cui, "Dynamic Motion Planning for Mobile Robots Using Potential Field Method", IEEE International Conference on Robotics and Automation, Vol.13 No.3, pp. 207-222, 2002.

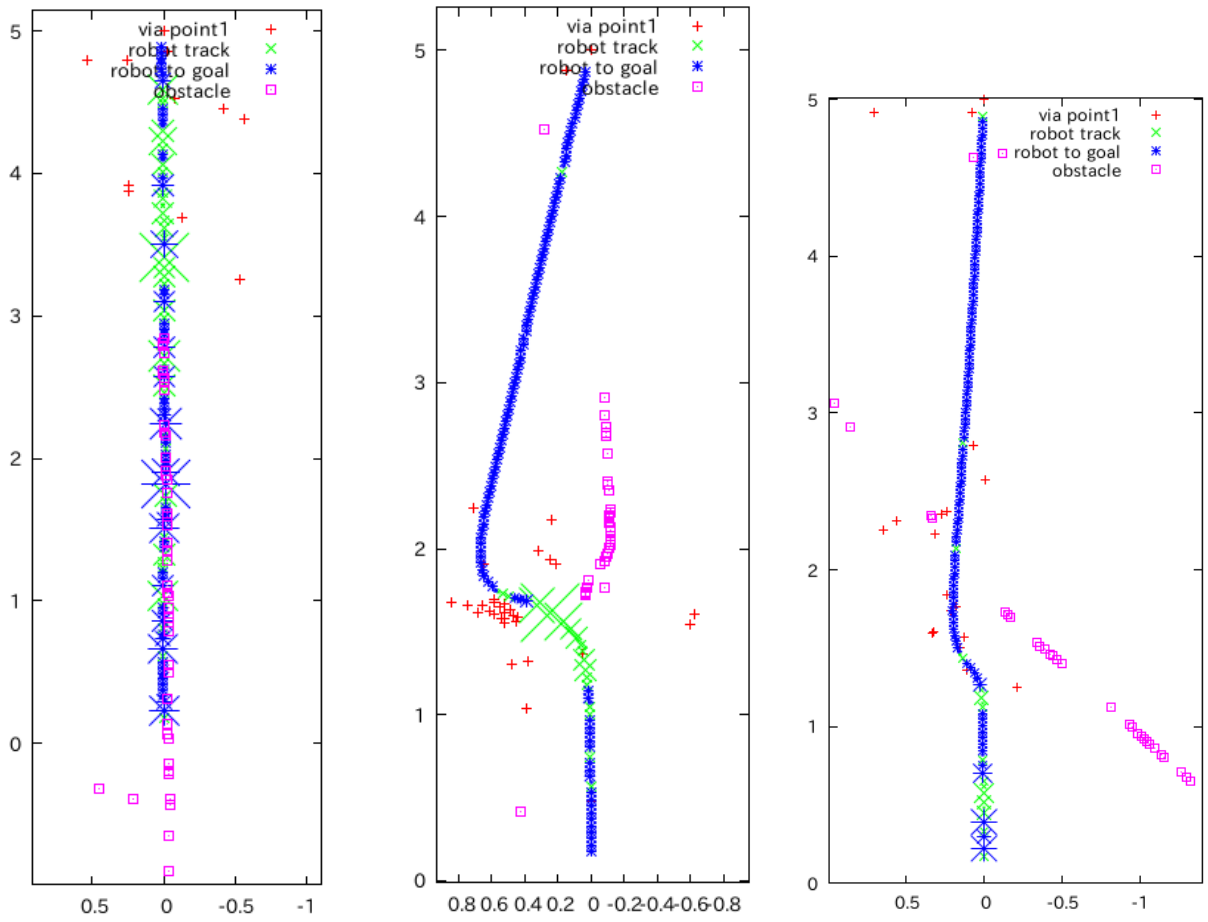
[6] R. Bis, H. Peng, G. Ulsoy, "Velocity Occupancy Space: Robot Navigation and Moving Obstacle Avoidance with Sensor Uncertainty", Dynamic Systems and Controls Conference, 2009.

[7] Konolige, K., 2000, "A gradient method for realtime robot control," Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1, pp. 639-646. 2000.

[8] M Seder, I Petrovic, "Dynamic Window Based Approach to Mobile Robot Motion Control in the Presence of Moving Obstacles", IEEE International Conference on Robotics and Automation, pp. 1986-1991, 2007.

[9] T. Tsubouchi, T. Naniwa, S. Arimoto:"Planning and Navigation by a Mobile Robot in the Presence of Multiple Moving Obstacles and Their Velocities", Journal of Robotics and Mechatronics, Vol.8 No.5, pp. 58-66, 1996.

[10] H. Sakahara, Y. Masutani, F. Miyazaki, "Real Time Motion Planning in Dynamic Environment Containing Moving Obstacles Using Spatiotemporal RRT", Transactions of the Society of Instrument and Control Engineers, Vol.43 No.4, pp. 277-284, 2007.



(a) Obstacle from behind robot (b) Obstacle from front (c) Obstacle from side of robot

Figure 9. Avoidance path in each case

a) The robot speeds up on the designated path to run ahead of obstacle from behind
b) The robot turns left to avoid the obstacle from ahead
c) The robot slows down and let the obstacle from the side pass before continuing on the path