Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

# Perception of Dynamic Environments in Autonomous Robots

**Meng Joo Er*, Zi Jing Bay* and Yi Zhou+**

**School of Electrical and Electronic Engineering*
*Nanyang Technological University*
*Block S1, 50 Nanyang Avenue,Singapore 639798*
*(e-mail: { emjer, BAYZ0001@ntu.edu.sg }@ntu.edu.sg)*

*+Department of Electrical and Electronic Engineering*
*Singapore Polytechnic, 500, Dover Road, Singapore 139651*
*(e-mail : zhouyi@sp.edu.sg)*

**Abstract:** Perception of dynamic environments is the first and most critical step in mobile robots. Without a good perception (e.g. mapping) that is close to the true environment of the robot, no accurate navigation or effective obstacle avoidance can be accomplished. This paper addresses the occlusion problem that occurs frequently in perception of dynamic environments. The use of the Bayesian Occupancy Filter (BOF) to address these issues is proposed in this paper. The BOF using a range sensor is implemented and problems encountered during the implementation of the BOF are discussed. Simulation results demonstrate the effectiveness of the proposed approach.

## 1. INTRODUCTION

Research on perception of environment has begun since the first Adaptive Cruise Control (ACC) system was introduced in the market in 1999. However, few results address the dynamic characteristics of the environment. The dynamic environment, which has obstacles like the human (which is the most common moving object that shares the same environment as the autonomous vehicle), will affect the performance of the localization algorithm, for example, the Simultaneous Localization and Mapping (SLAM) [1], or trigger ineffective action to avoid obstacles.

In this paper, the focus is on the implementation of currently available algorithms, such as occupancy grid mapping [2] to store environment information, probability perception [3] to include uncertainties in measurements, Bayes' theorem [2] to fuse sensor information over time and Bayesian Occupancy Filter [4][8] that makes use of the classical Bayes Filter to consider the dynamic characteristic of obstacles to solve the occlusion problem.

In autonomous navigation, dynamic obstacles, such as humans, sharing the same environment have been causing problems in accurate localization and obstacle avoidance for autonomous robots.

Localization requires features or objects of known location in order to localize or correct its position. Thus, moving objects, if taken as a reference, will cause in localization errors. Therefore, a preferred perception will be one that can differentiate between static and dynamic obstacles.

Obstacle avoidance is tedious if obstacles are dynamic. Dynamic obstacles can temporarily disappear and reappear due to occlusion, thus causing ineffective avoidance movement if an obstacle suddenly reappears in front of the robot. If the dynamic characteristic of the obstacle is not taken into consideration when determining the avoiding action, ineffective avoidance may occur.

Tracking [13][14] is commonly used in order to solve the aforementioned problems, which in turn leads to data association problems such as the list of objects to track or objects to delete from the list. In a normal traffic road situation, increasing the number of pedestrians causes an increase in computational time in tracking, therefore more robust and real-time perception is required.

In robotic perception, there are three ways to consider the dynamic objects in the environment:

The first way is to consider the dynamic object as a static object. This method revolves around taking the measurements of all obstacles regardless of whether they are moving or stationary and it is the easiest way to implement. In this case, SLAM cannot be used and obstacle that is moving towards the robot cannot be avoided safely.

The second way is to ignore the dynamic object [3]. If an obstacle appears nearer to the robot than expected, the readings are discarded. This is done to allow the robot to localize itself in a dynamic environment as only static objects are considered. The knowledge of the map is an a priori precondition for this.

The common way is to track the dynamic objects using multi-target tracking. This method of considering dynamic objects is tedious since as dynamic objects increase, the list of objects to track increases as well. Thus, it consumes significant amount of computational time and is not feasible in environments such as a road with other vehicles.

The perception algorithm that has been chosen to be implemented in this project is the Bayesian Occupancy Filter, which is derived from the classical Bayes filter [5]. The advantages for using the BOF are that it includes uncertainties in sensor measurements, stores information such as velocities and positions of obstacles and avoids problems in data association and modeling, thus increasing the robustness of the system. This algorithm makes use of prediction and estimation steps as illustrated in Figure 1 [4]. It predicts the probability of occupying a cell given a prior information (position, velocity) of a cell and control parameters of the robot. The estimation step will then fuse the predicted output with the new sensor measurement to give input to the next prediction step and the cycle repeats. The output of the prediction can be a probability of a cell or a map that can be fed into a path planner to avoid obstacles.

The occupancy map, besides storing the occupancy information of the cell, also stores the relative velocity of the cell with respect to the robot, hence allowing the robot to choose either a static map for localization or a map that includes the dynamic obstacles for effective obstacle avoidance.

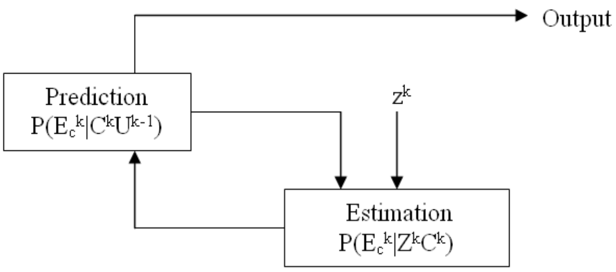


Figure 1. BOF as a recursive loop

The symbols are defined as follows:

C : cell information such as position and relative velocity;

$E_c$: occupancy of a cell, C;

U: control parameters of the robot;

K: time instance;

Z: sensor measurement;

$P(E_c^k|C^kU^{k-1})$: predicted probability of the cell state (occupied or empty) given the cell information and the control parameters,

$P(E_c^k|Z^kC^k)$: estimated probability of the cell state (occupied or empty) given the cell information and sensor measurements.

This paper is organized according to the implementation of BOF. Section II reviews the sensor model used for the range finder in this project. Section III describes the implementation of the occupancy grid map which stores information such as the positions and velocities of obstacles in the field of view. Section IV discusses the velocity estimation algorithm that is required to implement the BOF. In Section V, the implementation of the BOF is explained. Experimental results are presented at the end of every section.

## 2. SENSOR MODEL

Ranger finders are commonly used sensors in robotics research. In order to include uncertainties in range measurements, probability perception which gives the belief of each measurement, can be used. This can be done by designing a sensor model that the fits the sensor system in this project.

The sensor used in this project is the SICK LMS291 laser range finder. From the technical specifications of SICK LMS291 laser, the measurement error is typically +/- 35mm at a range of 1 to 20m. This error is negligible in comparison to the cell dimension that will be used to store the sensor measurements. Therefore, the maximum error in estimating the cell position, which is halve of the cell dimension of around 5cm, is used.

The sensor measurement is modeled using Gaussian estimation as shown in (1).

$$P(Z \mid Ec = OCC) = \frac{1}{\sqrt{2\Pi\sigma^2}} \exp(\frac{-(Z - Zmean)^2}{2\sigma^2}) \qquad (1)$$

where $P(E_c = OCC|Z)$ is the probability of occupancy, OCC, of a cell given Z, the distance of the cell from the sensor, $\sigma^2$ is the variance of the system and the $Z_{mean}$ is the measured distance.

## 3. OCCUPANCY GRID MAPPING

Occupancy grid map is made up of cells of equal dimension. The dimension of a cell is usually 5cm to 20cm for indoor robot and 15cm to 50cm for outdoor robot, and it depends on the information the robot needs to capture. One example is the environment which consists of the legs of a chair and tables. In this case, if these legs are considered as obstacles to the robot, the diameter of the legs of the chair can be considered as the cell dimension.

The information stored in the cell, C, is the probability of the state, $E_c$, of the cell which is either occupied, OCC, or empty, EMP. In this paper, the probability of velocity, $V_c$ of the cell is also stored in the cell to implement the BOF. The cell probability of occupancy is calculated using the Bayes' theorem as shown in (2),

$$P(E_c = OCC \mid Z_{t+1}) = \frac{P(Z_{t+1} \mid OCC)P(E_c = OCC \mid Z_t)}{\sum_{E_c} P(Z_{t+1} \mid E_c)P(E_c \mid Z_t)} \qquad (2)$$

where $P(E_c = OCC|Z_{t+1})$ is the probability of occupancy, OCC, of a cell given the sensor measurement, Z, at time t+1 and $P(Z_{t+1}| E_c = OCC)$ is the belief of the measurement Z at time t+1 given that the cell is occupied.

The Bayes theorem accumulates the probability of occupancy of a cell using the previous measurement, thus increasing its belief if they coincide and vice versa.

In practice, the cells are indexed [x, y] which enables easy updating of sensor measurement after a simple trigonometry conversion as shown in (3) and (4).

$$x = x^* + z\cos(\theta_1 + \theta_2) \qquad (3)$$

$$y = y^* + z\sin(\theta_1 + \theta_2) \qquad (4)$$

where $[x^*, y^*]$ is the position of the sensor, $\theta_1$ is the direction of the sensor system facing with respect to north, $\theta_2$ is the angle of the laser beam with respect to the sensor system and $z$ is the distance measured by the sensor.

a)

b

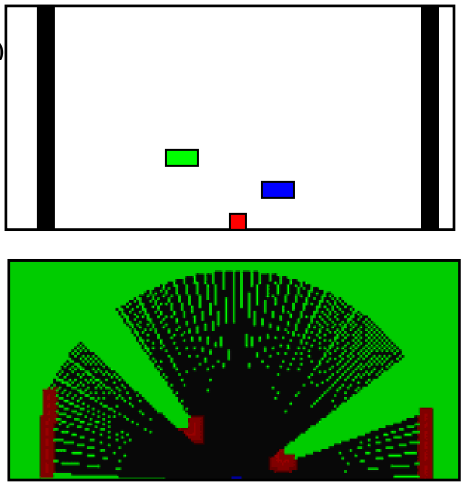

Figure 2. a) An environment with a robot (red) and two obstacles (green and blue), bounded by two black walls at the sides. b) The occupancy grid map of the environment, with unknown area (green), obstacles (red) and empty area (black).

Figure 2a shows the environment set up for mapping. The red box is the sensor location and the environment is occupied by two obstacles (green and blue rectangle) and bounded by two black walls at the sides of the sensor. The sensor used here is LMS 291 laser with a configuration of 8m range, 0.5° resolution and 180° field of view. The red area in Figure 2b is the obstacle with probability of occupancy shown by different tones of red. The higher the probability of occupancy, the brighter the red will be. The area in green is the unknown area and the area in black denotes grids that are empty. The belief of each sensor measurement, $P(Z_{t+1}| E_c = OCC)$, is given by the sensor model in Section II. The area behind the obstacles is unknown thus causing occlusion if dynamic obstacles are to travel behind these static obstacles. In this case, there will be appearance and disappearance of those dynamic obstacles, hence tracking will be difficult.

## 4. VELOCITY ESTIMATION

In order to consider dynamic obstacles, the dynamic characteristics or the velocities of the obstacles must be estimated. This can be done by tracking objects of the same feature at different times. The change in the position of the same obstacle over differences in time between measurements will give the velocity of each obstacle. In grid mapping, the feature modeled using range sensors will vary if the obstacle moves, therefore it is difficult to track the same object at different times. In tracking, data association problems such as observation to track and management (insert or delete) of objects from the list of objects to track must be taken care.

To address this problem, it is assumed that the dynamic grid will only move to its surrounding grid and will not change speed before it is captured again by the next scan. By virtue of these assumptions, the velocity of object can be estimated by comparing two maps sensed at different times. The mismatched areas of the map will then be used to estimate the velocity. In this way, the problem of modeling features and data association problems in tracking can be ignored.

The problem with velocity estimation by using the mismatch area is that only parts of a moving obstacle are detected. This is illustrated as shown below.

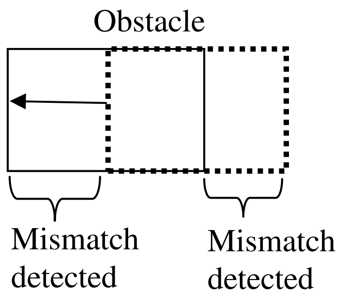


Figure 3. Only parts of an obstacle that is moving are detected mismatch.

Therefore, a preprocessing step such as grouping of occupied grids that belong to the same obstacle is required and these can be done by using the Breath First Search algorithm [6]. The cells that are occupied are given identity values and neighboring cells are given the same identity value. If a velocity is given to a cell, the same velocity is spread to the cells of the same identity value. Using this method, the partial mismatch problem is alleviated or even removed.

In practice, there will be false or missed estimation of velocities due to localization errors of the sensors and the obstacles. This problem can be reduced by increasing the belief of the velocity sensed only after a few similar scans by using the Bayes' theorem. The belief of the velocity estimated is modeled using Gaussian distribution of variance depending on the velocity of the obstacle with respect to the sensor. The higher the velocity estimated, the higher the variance is used. This is true as the errors in estimation will be higher for fast moving obstacles.

The environment set up in Figure 4a is similar to Figure 2a. The only difference is that the two obstacles in blue and green are dynamic. This experiment is set up to estimate the velocity of the moving obstacles. The occupancy grid map in Figure 4b shows the grids that are dynamic. The velocities of these grids estimated vary around $1 ms^{-1}$ which is close to the actual velocities of the obstacles. Although false and missed estimation of velocity occur, recursive measurement of the positions and velocities of the obstacles reduces the errors.

There are some limitations to the velocity estimation technique used here. The limitations are due to the use of grid mapping. The velocities estimated here can only be discrete values and slow moving obstacles, of less than 1 grid change per millisecond, are sensed only after they have moved for at least one grid length. These limitations can be adjusted by changing the grid dimension or the time difference between measurements. In this project, these limitations can be ignored since slow moving obstacles can be considered as static obstacles since they pose less danger of obstructing the path planned.

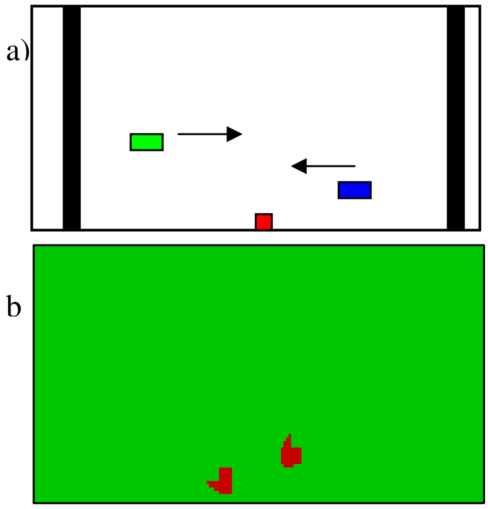


Figure 4. a) An environment with a robot, in red, and two obstacles, in green and blue, moving at 1ms$^{-1}$ and -1ms$^{-1}$ along the x-axis respectively. b) The occupancy grid map that shows only grids that are dynamic (velocity not equals to zero).

## 5. Bayesian Occupancy Filter

After occupancy grid mapping and velocity estimation are done, the BOF can be implemented. The BOF is derived from the Bayes Filter with a prediction stage and an estimation stage (as shown in Figure 1). The BOF makes use of the occupancy grid mapping to store the probability of occupancy and estimated velocities of the cells. This method of storing the map information allows a user to choose either a static map or a dynamic map (map with dynamic characteristic such as velocities of the obstacles in the surrounding) for processing. The other advantages are that it includes uncertainties in measurements and avoids problems in data association and modeling thus increasing the robustness of the system.

The prediction stage is based on the previous scanned result and control input to predict the next output measurement as shown in the following equation:

$$x_k = f(x_{k-1}, u_{k-1}, w) \cdots\cdots\cdots (5)$$

where $x_k$ is the obstacle position at time k, $x_{k-1}$ is the obstacle position at time k-1, $u_{k-1}$ is the control input parameters at time k-1 and $w$ is the noise of the control input.

The function, $f$, is the state transition equation. It depends on the system used and the control input of the system. If the control input parameters, $u_k$ are the changes in positions, the function, $f$ will be an addition to the prior position $x_{k-1}$with noise, $w$. In this paper, the control input is the change in positions with Gaussian noise of zero mean and variance depending on the velocity of the cell.

The predicted measurements are then fused with the current measurements using the Bayes' theorem to give the estimated measurements.

In practice, in order to keep the prior map information for fusion, a transformation equation is required. It is used to transform the prior map to the same reference coordinate system as the current map due to movement of sensors. The errors (Figure 5) in rounding up during transformation and localization will cause false estimation of velocity of stationary object.

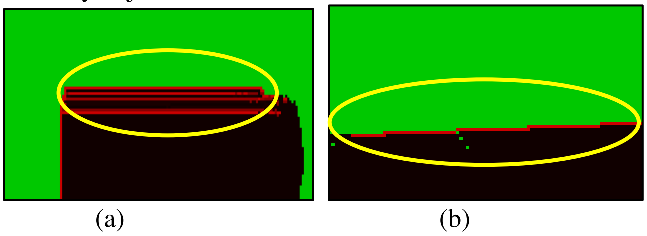


Figure 5. Pictures a) and b) shows segment of a map with errors. Map with localization error in a) and rounding up error in b) due to transformation of previous map coordinates to new map coordinates.

Figure 5 a) shows a segment of a map with localization error which causes multiple updates of the same obstacle at different positions. There are a number of ways to reduce these errors. One of the ways is to improve the localization of the robot using the map built during run time to predict the robot's next position. Classical kalman filter [7][15] can be used to fuse the predicted and measured position to give a more accurate position. In Figure 5b), the rounding up error causes an originally straight obstacle to be broken into several parts. These rounding up errors in transformation can be reduced by using the occupancy grid map with smaller dimension per cell. Another way is to maintain a global map to remove the need for transformation which is used in this paper.

The environment in Figure 6 is set up such that temporary occlusion will occur. The mobile robot (blue) is moving from left to right at 1ms$^{-1}$ in front of the sensor (red). The velocity of the mobile robot is estimated during the period while it is still visible (from screenshot a1 to b1) to the sensing robot. From screenshots c1 to d1, the mobile robot is blocked by the wall in front of the sensing robot but the estimated map from c2 to d2 is able to predict its movement behind the wall due to the prediction stage of the BOF. The mobile robot is predicted to be behind the wall in the prediction stage and the output (predicted map) is then fused with the new measurements to give an estimated map as shown in a2 to f2.

After the mobile robot reappears from e1 to f1, the measured position of the robot is fused with the predicted position, thus correcting the robot's position shown in the estimated maps e2 to f2. These results verify that the BOF algorithm does solve the occlusion problem and provide a more informative estimated map for obstacle avoidance. In practice, missed and false velocity estimation occurs, and it is corrected by the new measurement sensed. These errors are caused by slow moving obstacles and sensor localization errors.

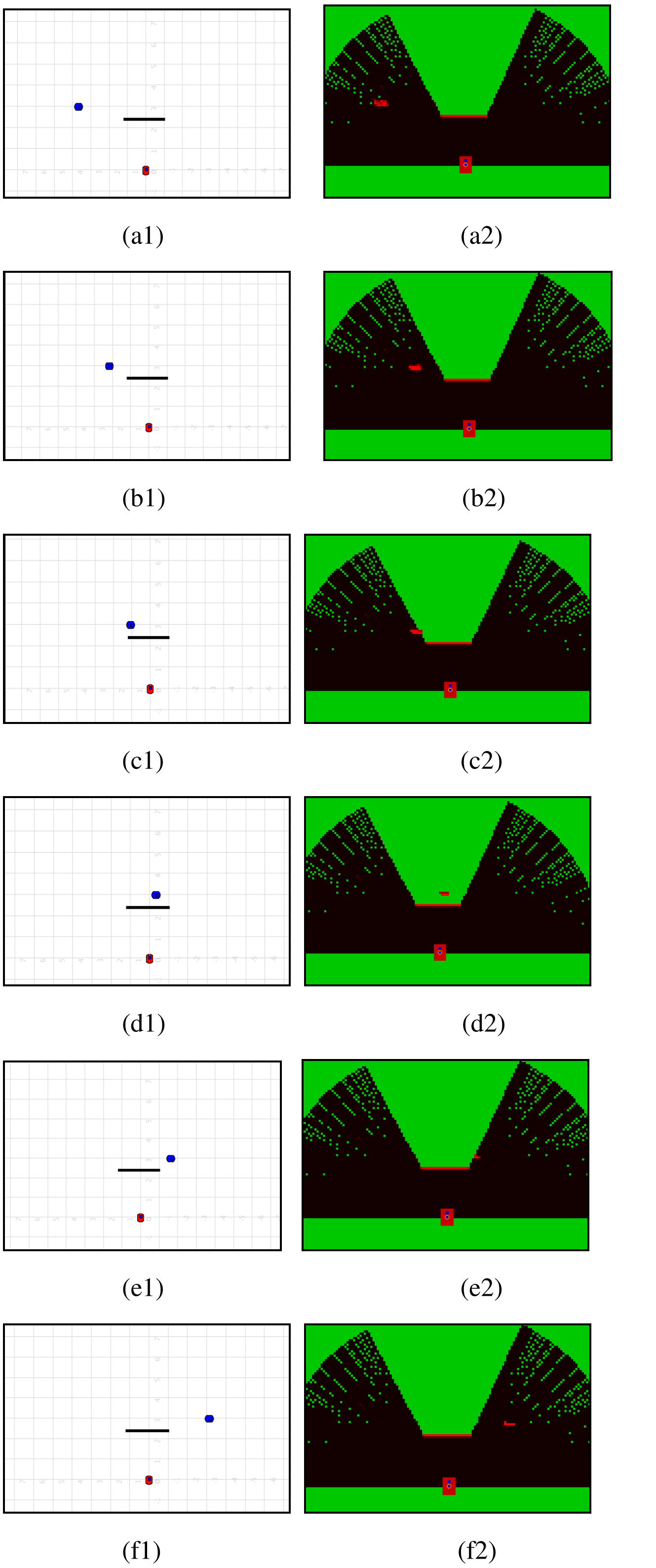


Figure 6) Screenshots of environment (a1 to f1) set up to test BOF and the results of the estimated maps are from a2 to f2. Screenshots a1 to a2 show a mobile robot (blue) moving from left to right in front of the stationary sensor (red). Temporary occlusion (blue robot blocked by the black wall) occurs from screenshots c1 to d1. The mobile robot (blue) reappears from screenshots e1 to f1.

## 6. Obstacles Avoidance

The output of the BOF can be for path planning such as D* algorithm [10] and motion planning [9], obstacle avoidance with arc path [11] and vector field histogram [12], localization [1] or even information capture. In this paper, the BOF is used to output a map for obstacle avoidance. The obstacle avoidance algorithm used here is a simple straight line speed selection to avoid obstacles in front of the robot. The objective of this section is to verify the use of BOF output can improve on obstacle avoidance for temporary occluded obstacle.

The speed profile used here is a trapezoidal curve as shown in Figure 7, with a maximum speed of $4ms^{-1}$. The obstacle avoidance algorithm selects the maximum speed input to the robot if the sensor's front view has a clearance of greater than or equal to 4m. The speed selected decreases linearly with respect to sensor's front clearance as shown in (6).

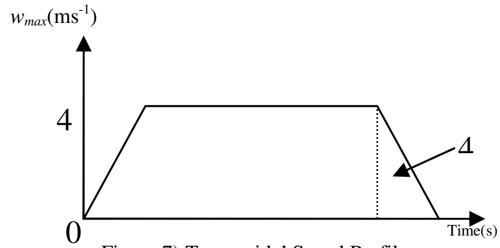


Figure 7) Trapezoidal Speed Profile

$$Speed = \begin{cases} c & 0 \le c \le 4 \\ w_{\max} & c \ge 4 \end{cases} \qquad \cdots\cdots(6)$$

where $c$ is the clearance in front of the sensor and $w_{\max}$ is the maximum speed allowed.

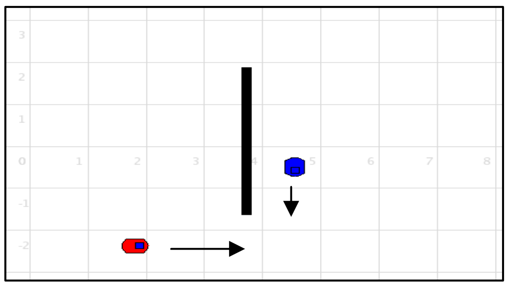

Figure 8) Environment that is set up for straight line obstacle avoidance.

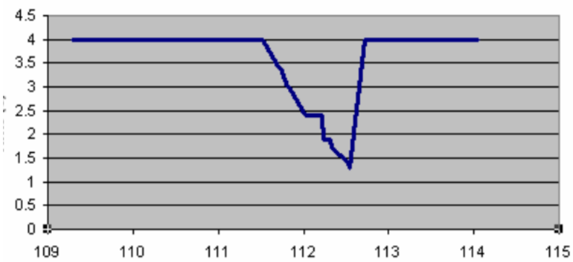


Figure 9) Plot of speed ($ms^{-1}$) of robot (red) against time (ms)

By using this obstacle avoidance approach, the robot is able to move in a straight line avoiding dynamic obstacles blocking its way. The experiment setup is shown in Figure 8. The robot (red) is able to avoid the dynamic obstacle (blue) even when temporary occlusion occurs. The speed of the robot measured is plotted in Figure 9. The sudden change in speed at time 111.5ms is due to avoidance of the dynamic obstacles that will be blocking the robot before it even appears from behind the wall. This is the advantage of using Bayesian Occupancy Filter (BOF) in maintaining the occupancy grid map. The dynamic obstacles were constantly tracked on the grid map which in turn provides a more informative map for the obstacle avoidance algorithm to avoid obstacles.

It is shown that the BOF does help to increase the chances of avoiding obstacle temporary occluded in a dynamic environment. The algorithm is kept simple, thus increasing its robustness for situations which have multiple dynamic obstacles appearing and disappearing in its field of view. This situation will be common in future robotic field where their operation environments are usually filled dynamic obstacles such as people.

## 7. CONCLUSIONS

In this paper, the problems caused by dynamic obstacles sharing the same workspace as the robot are addressed. The perception algorithm carried out to solve these problems exploits the use of the BOF. The steps involved, such as occupancy grid mapping and velocity estimation, are discussed. Simulation results using a laser range finder mounted on a robot demonstrate the effectiveness of the proposed algorithm. After the implementation of the BOF, simple straight line obstacle avoidance test is carried out and it is proven to increase the robot's chances of avoiding temporarily occluded obstacles. Given its simplicity, the robustness of the perception algorithm is achieved even in highly dynamic environments.

## REFERENCES

[1] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in Proceedings International Conference. Robotics and Automation, Septemeber, 2003 Page(s):842 - 849

[2] Elfes, A, "Using occupancy grids for mobile robot perception and navigation," IEEE computer Volume 22, Issue 6, June 1989 Page(s):46 - 57

[3] S. Thrun, W. Burgard, and D. Fox. Probability Robotics. MIT Press, 2005.

[4] Christophe Coué , Cédric Pradalier, Christian Laugier, Thierry Fraichard, Pierre Bessière, 2006, "Bayesian Occupancy Filtering For Multi-target Tracking," The International Journal of Robotics Research Vol. 25, No. 1, January 2006, Page(s): 19-30

[5] A. H. Jazwinsky. Stochastic Processes and Filtering Theory. New York: Academic Press: 1970.

[6] Richard Johnsonbaugh, Marcus Schaefer. 2004. Algorithms. Pearson Prentice Hall. Page(s): 181-187.

[7] P.J. Hargrave. 1989. "A tutorial introduction to Kalman filtering," Kalman Filters: Introduction, Applications and Future Developments, IEE Colloquium on 21 Feb 1989 Page(s):1/1 - 1/6.

[8] C. Chen, C.Tay, C. Laugier and K. Mekhnacha. "Dynamic Environment Modeling with Gridmap: A multiple-Object Tracking Application," Control, Automation, Robotics and Vision, 2006. ICARCV ' 06. 9th International Conference on December, 2006 Page(s): 1-6

[9] Nancy Amato, "A Brief Introduction: The Motion Planner Problem, Configuration Space, Basic Path Planning Methods", University of Padova,2004.

[10] Antony Stentz, "The D* Algorithm for Real-time Planning of Optimal Traverses", The Robotics Institute, Carnegie Mellon University; Pittsburgh, September 1994.

[11] Ankur Naik , "Arc Path Collision Avoidance Algorithm for Autonomous Ground Vehicles", Blacksburg Virgina, Virginia Polytechnic Institute and State University, December 15, 2005

[12] J. Borenstein, Y. Koren, "Real-time Obstacle Avoidance for Fast Mobile Robots", Third International Symposium on Intelligent Control, Arlington, Virginia, August 24-26, 1988

[13] Dirk Schulz, Wolfram Burgard, Dieter Fox, Armin B. Cremers, "Tracking Multiple Objects with a Mobile Robot", University of Bonn, University of Freiburg, University of Washington, 2001.

[14] C. Cou´e, Th. Fraichard, P. Bessi`ere,E. Mazer, "Using Bayesian Programming for Multi-Sensor Multi-Target Tracking in Automotive Applications", International Conference on Robotics and Automation, May 12-17, 2003. Taipei.

[15] Greg Welch, Gary Bishop, "An Introduction to Kalman Filter", Chapel Hill, Univeristy of North Carolina, Department of Computer Science, April 2004.