

Speed and direction of an obstacle using virtual window

Hj. M.A. Hj. Mansor

Faculty of Electrical Engineering, MARA University of Technology 40450 Shah Alam, Malaysia

Article Info

Article history:

Received Aug 24, 2018

Revised Oct 15, 2018

Accepted Oct 28, 2018

Keywords:

Relative path equation

Relative speed determination

Uniformly moving obstacle

Virtual window

ABSTRACT

A virtual window is used to determine the direction and speed of a uniformly moving obstacle. Two intersections with the virtual window at different location are used to calculate the relative path and speed of the obstacle. Two simulations were performed using Excel 2010. The first simulation simulates a practical running of a uniformly moving obstacle, while the second has the obstacle moving at a very high speed. The result shows that the system was able to determine the relative speed and path of the uniformly moving obstacle accurately.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Hj. M.A. Hj. Mansor,
Faculty of Electrical Engineering,
MARA University of Technology,
40450 Shah Alam, Malaysia.
Email: pakngah60@gmail.com

1. INTRODUCTION

In recent years many research works are directed to the solution of obstacle avoidance in path planning. This is due to the interest in autonomous vehicles, such as DARPA Urban Challengers [1] and by automotive companies such as Mercedes Benz, Audi, Volvo and Google [2-4].

An autonomous car [5] is a vehicle that do not need human control and it basically drives itself. It is also called driverless car or self-driving car. Autonomous cars utilise various kind of techniques to sense their environment, like radar, laser light, GPS, odometer, and computer vision. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage [6-7].

This work concerns mobile robot, but the technique applied can be used in autonomous cars as well. When a car or a mobile robot moves in an unknown and changing environment, the main aim is to get to its goal without any collision. If the path taken happens to be optimal then that is a bonus. Obstacle avoidance is the basic requirement of autonomous robot navigation. The robot needs to acquire information about its surroundings along with any stationary and moving obstacles present so that a navigation system with path planning algorithm is able to guide the robot to the target without colliding with obstacles within the environment.

There are many path planning algorithms with obstacle avoidance that has been proposed; such as potential field [8-10], visibility graphs [11-12], grid methods [13], and virtual window [14]. Obstacle avoidance research has traditionally been handled by two techniques [15]: the deliberative approach, usually consisting of a motion planner, and the reactive approach, based on the instantaneous sensed information.

The research work that is reported here is based on reactive approach utilising virtual window. This is an extension and improvement of a previous work described in [16]. Unlike [17-18], the technique implemented can be used to determine the path and speed of a moving obstacle. In this work, the environment is assumed to have only a single obstacle moving in a straight line with constant speed.

The technique is used to track the movement of the obstacle by calculating its path and speed from two crossings with a virtual window. This virtual window is generated from a device that is located either on a stationary or moving mobile robot. When the direction and speed of the obstacle is known, appropriate action can be taken to avoid collision. Speed and path of the obstacle are taken as being relative to the mobile robot.

2. BRIEF DISCUSSION OF THE VIRTUAL WINDOW

A virtual window basically acts as a sensor and is a rectangular plane that is projected ahead of a mobile robot for the purpose of detecting obstacle. Figure 1 shows a mobile robot having a virtual window 1 m ahead with a square shape (1 m by 1 m) having a pixel resolution of 10×10. Depending on the requirements of the work, the shape can be rectangular, and its width and height, and pixel resolution of the virtual window can be set to other values other than that shown in Figure 1.

In a vision system, an image of the view forward of the mobile robot is captured. Intensive and complex image processing will have to be applied in order to retrieve the relevant information. This is due to the fact that the captured information consist of the image of the plane of interest and also image before and beyond that plane, albeit out of focus (3D view to 2D image). On the other hand, the virtual window will return only the image of the plane of interest.

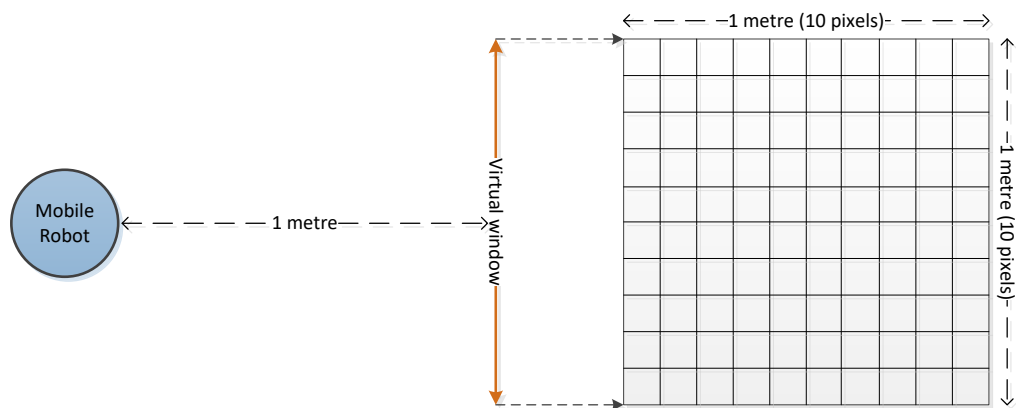


Figure 1. A diagram showing a mobile robot with its corresponding generated virtual window

This virtual window has a similar characteristic as a camera in that it has a display resolution. This display resolution can be set to any value depending on the needs of the work by changing the amount of data that is read from the plane of interest. The size of the virtual window, i.e. the length and width, can also be set to any value that is required. Ideally, the size of the virtual window is usually rendered a little bigger than the size of the robot. Obviously this allows for a bigger forward image to be monitored and thus a bigger safety net. A full and complete discussion of the possible implementation of the virtual window can be read from [13].

3. SIMULATION

As was described in [13], a laser range finder can be used to generate the virtual window. In this arrangement, the set-up for the virtual window is as shown in Figure 2, with the laser range finder being on the mobile robot. The range of the virtual window is the range where any intersection or collision with the obstacle is considered. Outside of this range, any occurring collisions are ignored. D is the minimum distance of the virtual window from the mobile robot, d is the width of the range, t_1 and t_2 are the times that light takes to travel from the mobile robot to a point on the vw_1 and vw_2 , respectively, and R is the pixel resolution of the virtual window.

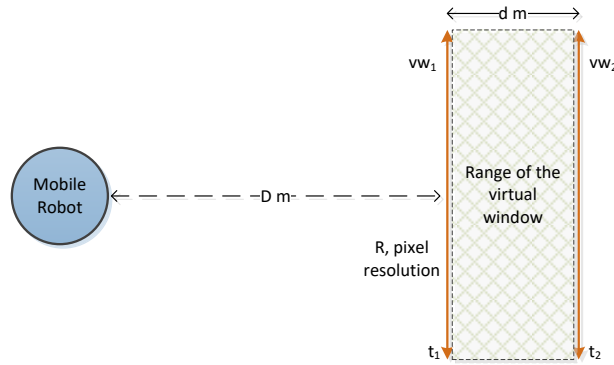


Figure 2. A diagram showing a mobile robot with its virtual window range and general parameters

For this work, it is assumed that the width of the range, or d , is 1 cm. The minimum distance to the virtual window, D , is set at 1 m. These values are set arbitrarily, i.e. the values can be changed to other values subject to the ability of the laser range finder to accurately measure the time differences.

Assuming the speed of light in air to be 2×10^8 ms⁻¹, then the time taken for a single laser beam to travel to a point on the plane of interest, in this instance 1 m from the mobile robot, and back is about 10 nsecs. The time taken to travel a further 1 cm (or a total of 2.02 m) is 10.1 nsecs.

For a total pixel count of 100, i.e. the resolution of the virtual window being 10×10 pixels, the total time taken will be between 1 μsecs and 1.01 μsecs. For comparison, for a pixel count of 10,000 (100×100), the total time is 0.1 msecs and 0.101 msecs, respectively. If the virtual window is considered to be a square of 1 m by 1 m as in [13], a pixel resolution of 10×10 will still detect an intersection or collision with an obstacle as small as 100 mm (or 10 cm) in size, either width or height. Current practical mobile robots are usually much bigger than 100 mm in sizes [19-21]. Much smaller mobile robots can be considered, however this will have some impact on the time needed to process the information.

If there happens to be an obstacle at the plane of interest within the range, the beam that is reflected within the specified range of time will be considered. Outside of this range of time period, it is assumed that no collision occurs with the obstacle.

Since there is a range of time that the obstacle will be considered as intersecting the virtual window, there is an associated maximum speed that the obstacle can travel that will still allow the system to have the two intersections with the virtual windows. Otherwise, only one intersection will be detected, and that single information do not allow for the system to calculate the path and speed of the obstacle. This maximum relative speed depends on the parameters shown in Figure 2 and is shown in the (1) below:

$$v = \frac{cd}{2R(2D+d)} \tag{1}$$

where c is the speed of light in air. Taking the values stated above, $v = 4,975.124$ m/s. This is equivalent to 17,910.45km/hr. As a comparison, if R is increased to 10,000 (100×100 pixels), the speed is 179.1km/hr. These times are inclusive of processing time.

Table 1 shows the relationship between the relative speed of the obstacle and its distance travelled for:

$$t_T = t_1 + t_2 \tag{2}$$

where t_T is the total scanning time of the virtual window, with t_1 , at first intersection and the second intersection at t_2 . In this situation, only the speed of the obstacle is varied, while the rest of the parameters remain constant.

Table 1. Relationships between Relative Speed of Obstacle and Distance between the two Intersections

	(second)	Speed _{relative} (m/s)	Distance _{travel} (cm)
t_1	0.00000101	4000.000	0.80400
t_2	0.000001	4975.124	1.00000
		5000.000	1.00500
		5200.000	1.04520

As can be seen, the limit of the relative speed is 4975.124m/s, as calculated, where the distance travelled between the two intersections is 1 cm. Higher relative speed will not allow the system enough time to process the information since the range of the virtual window was set at 1 cm apart Figure 2. Of course, this can be rectified by increasing the range.

Simulations are performed using Excel 2010 to calculate the distance travelled by a uniformly moving obstacle in the general direction of the mobile robot. The speed and path is taken as relative to the moving mobile robot. The time period of the calculation is taken to be the total time that the system needs to scan the whole virtual window at the two intersections Figure 3. This is a valid assumption since the two intersections take roughly the sum of the two scanning time. It is stated as being roughly because, the exact time depends at which pixel the intersection occurs.

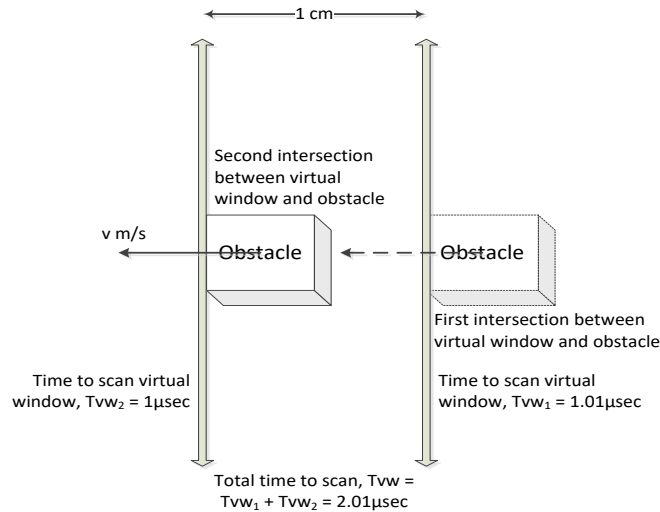


Figure 3. A figure showing the parameters considered in the simulation

4. RESULTS AND DISCUSSION

Figure 4 shows a model demonstrating the simulation that was performed for this research. The obstacle is moving towards the general direction of the mobile robot. The relative speed of the moving obstacle is v m/s, and the first intersection with the virtual window was assumed to be at coordinate (98, 101). Time period for the calculation of the simulation starts from that moment on.

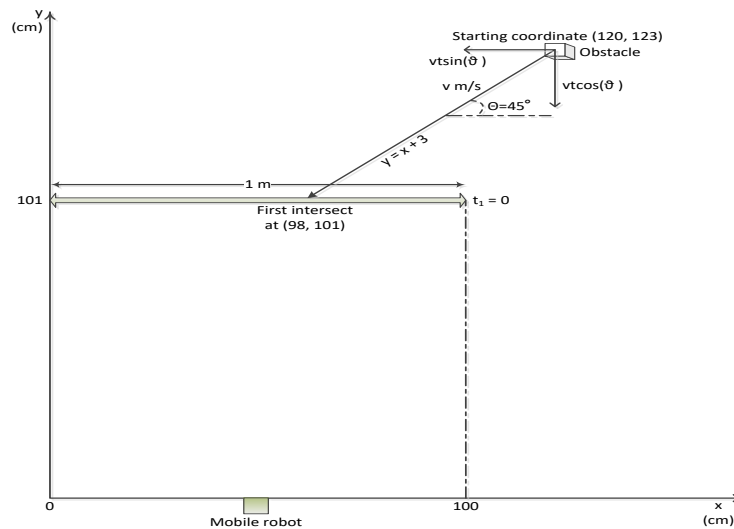


Figure 4. A model representing the simulation setting that was considered

For every time period, the time it takes for the system to sweep or scan the virtual window, its location in the x and y direction was calculated. Shown and discussed in this report are two results obtained from the simulation. Table 2a dan 2b shows two samples of the result of the simulation.

Table 2a. A Table Showing the Results from the Simulation with Time Period at 2.01 μsecs

No	t (second)	T _{vw} (second)	x (cm)	y (cm)
1	0.00000000	0.000001010	98.00000	101.00000
2	0.00000201	0.000001010	97.96447	100.96447
3	0.00000402	0.000001009	97.92894	100.92894

Table 2b. A Table Showing the Results from the Simulation with Time Period at 0.9 μsecs

No	t (second)	T _{vw} (second)	x (cm)	y (cm)
1	0.00000000	0.000001010	98.00000	101.00000
2	0.00000090	0.000001010	97.98409	100.98409
3	0.00000180	0.000001010	97.96818	100.96818

The last two columns of Table 2 give the x- and y-distance of the obstacle from the origin, respectively. The relative speed is assumed to be 250 m/s, angle θ at 45°, speed of light at 2×10^8 m/s, and pixel resolution of 10×10 . The size of the virtual window is set at 1×1 m.

Row 1 and column T_{vw} of Table 2a shows that the first scan time of the virtual window takes 0.000001010 secs. That is also the first intersection of the obstacle with the virtual window, thus t=0. After 0.00000201 seconds later, the second intersection with the virtual window occurs. As can be seen, the choice of the time period is logical as there is enough time for the first scan to finish before starting with the second scan. If the time period was less, say 0.0000009 second as in Table 2b, it can be seen that the scan of the virtual window will not have completed when the second time period calculation is started. Even though the calculation can be done in simulation, in practice the first scan is not yet completed and thus will produce an illogical result. The suggested time period is the minimum time period.

From Table 2a and depicted in Figure 5, the relative path and speed of the obstacle can then be calculated. Taking the two coordinates, (98, 101) and (97.96447, 100.96447), the gradient is found to be 1 and the y-intersect is 3. This gives the path equation as $y = x + 3$. For relative speed, the time it takes for the obstacle to travel from point (98) to point (97.96447) in the x direction, and from point (101) to point (100.96447) in the y direction is 0.00000201 sec each. Both will give a speed of 176.7662 m/s. The resultant of the two components will give a relative speed of 249.985 m/s at an angle of 45°. This result compares very well with the parameters used in the simulation.

Table 3 and Figure 6 shows a second simulation performed but with a relative speed of 5000 m/s towards the direction of the mobile robot. Discussed earlier that the maximum speed acceptable in order to still be able to detect the two intersections is 4975.124 m/s, this simulation will fail in its attempt to get a valid relative path and speed of the obstacle. Even though the time period is more than the time taken to scan the virtual window, there is only one valid result. The second result cannot be applied as it exceeds the range that was set earlier, 1 cm.

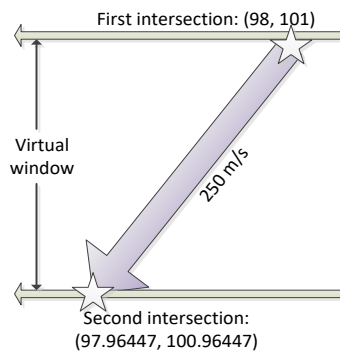


Figure 5. The two intersections with the virtual window in the first simulation

Table 3. A Table Showing the Results from Second Simulation using Parameters Shown

No	t (second)	T_{vw} (second)	x (cm)	y (cm)
1	0.000000000	0.000001010	98.000	101.000
2	0.000002010	0.000001000	98.000	99.995
3	0.000004020	0.000000990	98.000	98.990
Relative speed, v (m/s):		5000.00	Speed of light, c (m/s):	2×10^8
Angle, θ (degrees):		90.00	Pixel resolution, R:	100

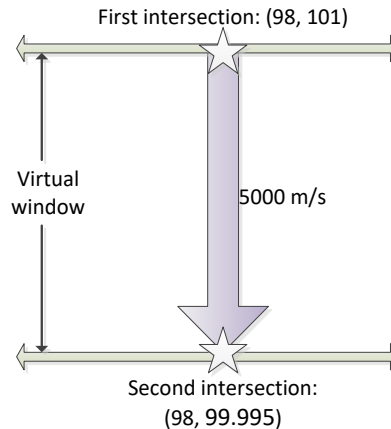


Figure 6. The two intersections with the virtual window in the second simulation

5. CONCLUSION

It was shown that the system was able to determine the relative speed and path of the uniformly moving obstacle when there are two intersections with the virtual window. With the simulation, the choice of the time period has to be a carefully selected value. If the value chosen is valid, that is, it takes into consideration the time needed for the two scan of the virtual window, then the result of the simulation is acceptable. Otherwise, even though the result is present but in practice it cannot be realised as explained earlier.

ACKNOWLEDGMENT

The authors would like to acknowledge that this research project is funded from a grant awarded by the Ministry of Higher Education: FRGS/1/2017/ICT04/UITM/02/5.

REFERENCES

- [1] Buehler M *et al.* "Special Issue on the 2007 DARPA Urban Challenge", Part I – III. *Journal of Field Robotics*. 2008; 25(8-10): 423-860.
- [2] Volvo News. "Autonomous Driving according to Volvo Car Group: benefits for society and consumers alike". <http://www.volvocars.com/us/top/about/news-awards/pages/default.aspx?itemid=68>. December 2, 2013.
- [3] Hagon T. "Audi says cars that accelerate, steer and brake in traffic jams could be here sooner than most think". <http://www.drive.com.au/motor-news/driverless-cars-possible-by-2015-20130313-2fz96.html>. March 13, 2013.
- [4] English A. "Autonomous cars – is this the end of driving?" <https://www.telegraph.co.uk/motoring/road-safety/10570935/Autonomous-cars-is-this-the-end-of-driving.html>. January 16, 2014.
- [5] Thrun, Sebastian. "Toward Robotic Cars". *Communications of the ACM*. 2010; 53(4): 99–106.
- [6] Lassa, Todd. "The Beginning of the End of Driving". http://www.motortrend.com/features/auto_news/2012/1301_the_beginning_of_the_end_of_driving/. *Motor Trend*. January 2013 (Retrieved 1 September 2014).
- [7] European Technology Platform on Smart Systems Integration (EPoSS). "European Roadmap Smart Systems for Automated Driving". <https://www.smart-systems-integration.org/public/news-events/news/eposs-roadmap-smart-systems-for-automated-driving-now-published>. 2015.
- [8] Ge SS & Cui YJ. "Dynamic Motion Planning for Mobile Robots using Potential Field Method". *Proc. IEEE Int. Conf. Autonomous Robots*. 2002; 13(3): 207-222.

- [9] Sugiyama S *et al.* "Path Planning of a Mobile Robot for Avoiding Moving Obstacles with Improved Velocity Control by Using the Hydrodynamic Potential". *Int. Conf. on Intelligent Robots and Systems (IROS2010)*. Taipei, Taiwan. 2010: 18-22.
- [10] Yin L *et al.* "A New Potential Field Method for Mobile Robot Path Planning in the Dynamic Environments". *Asian Journal of Control*. 2009: 11(2): 214-225.
- [11] Raja P & Pugazhenti S. "Path Planning for a Mobile Robot in Dynamic Environments". *Int. Journal of Physical Sciences*. 2011: 6(20): 4721-4731.
- [12] Li L *et al.* "Present State and Future Development of Mobile Robot Technology Research". *Robot*. 2002: 24(5): 475-480.
- [13] Boschian V & Pruski A. "Grid Modelling of Robot Cells: A Memory-Efficient Approach". *Journal of Intelligent and Robotic Systems*. 1993: 8(2): 201-203.
- [14] Mansor MA & Morris AS. "Path Planning in Unknown Environment with Obstacles Using Virtual Window". *Journal of Intelligent and Robotic Systems*. 1999: 24(3): 235-251.
- [15] Minguez J *et al.* "Motion Planning and Obstacle Avoidance". *Springer Handbook of Robotics*, Springer. 2008: 827-852.
- [16] Mansor MA *et al.* "Application of Virtual Windows to Determine the Path of a Uniformly Moving Obstacle". *IEEE International Conference on Control System, Computing and Engineering (ICCSCE2013)*. Penang, Malaysia. November 29–December 1, 2013.
- [17] Ren L *et al.* "A New Fuzzy Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot". *Int. Conf. on Mechatronics and Automation (ICMA2012)*. Chengdu, China. August 5–8, 2012.
- [18] Le XT *et al.* "Real-Time Obstacle Avoidance of Mobile Robots". *Int. Conf. on Control, Automation and Systems (ICCAS2007)*. Seoul, Korea. October 17-20, 2007.
- [19] <http://www.robotshop.com/4wd-omni-directional-mobile-robot-kit-2.html>
- [20] <http://www.dfrobot.com/>
- [21] <http://www.coreconagvs.com/products>