Paper: ASAT-16-121-AV

16th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 16 – May 26 - 28, 2015, E-Mail: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel : +(202) 24025292 – 24036138, Fax: +(202) 22621908

Simulation of a Laser Measurement System LMS Sensor Mounted on a Moving Vehicle Detecting Moving Obstacles

Gamal Abbass Zaghloul† and H. Arnaoot *
Port Said University

Abstract: this paper introduces a new algorithm to generate simulated 2D LMS point scan sensor mounted on a moving vehicle detecting moving obstacles,
The suggested algorithm could be used to asset the testing of autonomous vehicle LMS data processing system to avoiding unnecessary 3D long simulation time and complexity
The suggested algorithm was tested with a number of obstacles with different sizes, velocities and distances and compared with the input obstacle data, finally this paper introduces a formula to estimate detection possibility based on LMS angular resolution, obstacle shape and obstacle distance to aid in the choice of LMS angular resolution.

Keywords: Simulation, obstacle Detection, LMS, autonomous navigation, point scan.

1. Introduction

Along with the increasing use of LMS, comes the need for a stable and accurate simulation system [1]. It could be used to test LMS data processing systems. The suggested algorithm is meant to be used to predict the LMS output data in different situation. The suggested algorithm was tested using a VB program written for this purpose, see figure 1.

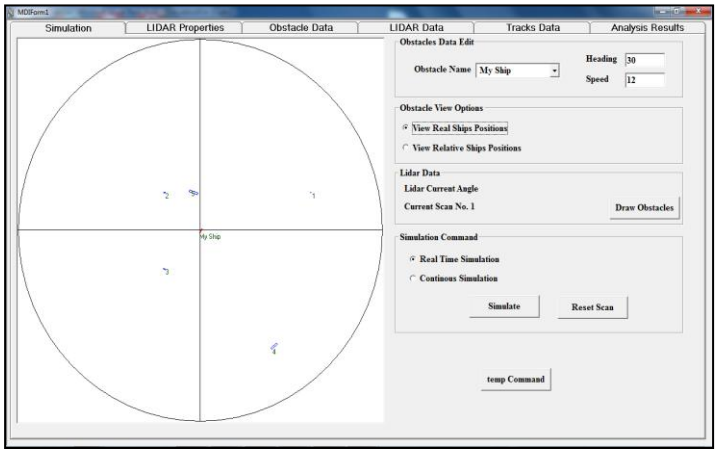

Fig. 1. Snap Shot of the Application Created to Simulate LMS Action

It should be mentioned that 3D calculation requires large computer resources and leads to delay [2], so it is preferred to avoid using 3D calculation whenever it is possible.

† Lecturer, Mechanical Power Engineering Dept., Faculty of Engineering, , Gazmousa@hotmail.com
* MSc. Student, Hanyarnaoot@yahoo.com

The suggested algorithm could be used to solve the famous problem of finding the intersection between a line segment and a polygon. This problem is somehow similar to intersecting ray or segment against box needed in collision detection, but the suggested algorithm solves for polygons with virtually unlimited number of sides not a rectangular with lines parallel to X,Y axis's and is not limited to four perpendicular corners, [3].
The suggested algorithm is meant to simulate point-scanning systems not line scanning systems, [4]
There has been a previous attempt to simulate LMS sensors based on using modern computer graphics hardware making heavy use of recent technologies like vertex and fragment shaders, [5].

2. Obstacle Points Co-Ordinates Calculation:

Obstacle characteristics include (position co-ordinates (X, Y), obstacle width, obstacle Length, Heading and speed. The obstacle is assumed as a rectangle, however the suggested algorithm works for obstacle with more than four corners (theoretically infinite number of corners) but for simplicity the rectangle shape (with four points) was chosen.

Based on the previous obstacle characteristics the obstacle four corners points co-ordinates could be calculated as shown in figure 2.

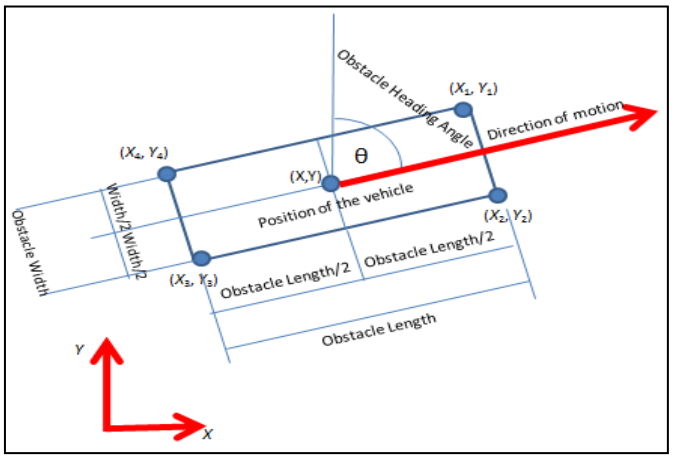

Fig. 2. Relation between Obstacle Position Co-Ordinates and Corner Co-Ordinates

3. The Simulation Algorithm

Obstacle Position Update:
During the simulation the new positions co-ordinates is calculated according to current heading and speed of all obstacles and LMS carrying vehicle using the following equations, see figure 3.

$$x_n = x_{n-1} + V.\sin(\theta).\Delta t \quad (3)$$

$$y_n = y_{n-1} + V.\cos(\theta).\Delta t \quad (4)$$

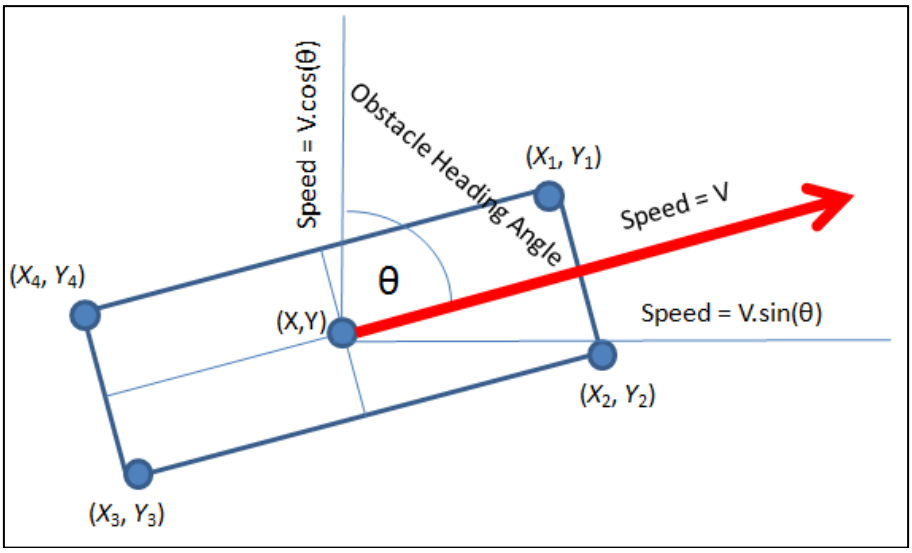

Fig. 3. Speed Component of a Moving Vehicle

Where:

θ	.....	Heading angle
x_n	.....	The **X** co-ordinates of the obstacle at n^{th} runs of the simulation
y_n	.....	The **Y** co-ordinates of the obstacle at n^{th} runs of the simulation
V	.....	Obstacle Speed
Δt	.....	Time intervals between two consecutive position update

LMS Simulated Value Calculation:

Before proceeding with LMS simulation two terms must be well defined the first term is the obstacle corners lines which are the group of lines forming the obstacle shape (polygon corners), for the case of rectangle there are four points and hence four corner lines.

The Second term is the scanning line which is the line beginning at current LMS carrying vehicle position and ends at LMS maximum range at current scan line angle).

Calculation steps

1- Obtain obstacle(s) data e.g. (position, heading, speed…)

2- Calculate obstacle(s) corners points co-ordinates(x,y)

3- Calculate LMS scan ray end point co-ordinates (the point at distance equal to LMS maximum range at current LMS scan angle)

4- Calculate intersection point P(x,y) co-ordinates between every obstacle corner line (in blue) and scan line (in red) using the equation 3 and 4 , [6] , as shown in figure 4, where :

P_x	.....	is the x axis co-ordinate of the intersection point
P_y	.....	is the y axis co-ordinate of the intersection point.

$$p_x = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & y_3 \\ x_3 & y_3 \end{vmatrix} & \begin{vmatrix} x_3 & 1 \\ x_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & 1 \\ x_3 & 1 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} \quad (3)$$

$$p_y = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & y_3 \\ x_3 & y_3 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \end{vmatrix} \\ \begin{vmatrix} x_3 & 1 \\ x_3 & 1 \end{vmatrix} & \begin{vmatrix} y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} \quad (4)$$

Or simply

$$P_x = \frac{(x_1y_1 - x_2y_1)(x_3 - x_4) - (x_3y_4 - x_4y_3)(x_1 - x_2))}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \quad (5)$$

$$P_y = \frac{(x_1y_1 - x_2y_1)(y_3 - y_4) - (x_3y_4 - x_4y_3)(x_1 - x_2))}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \quad (6)$$

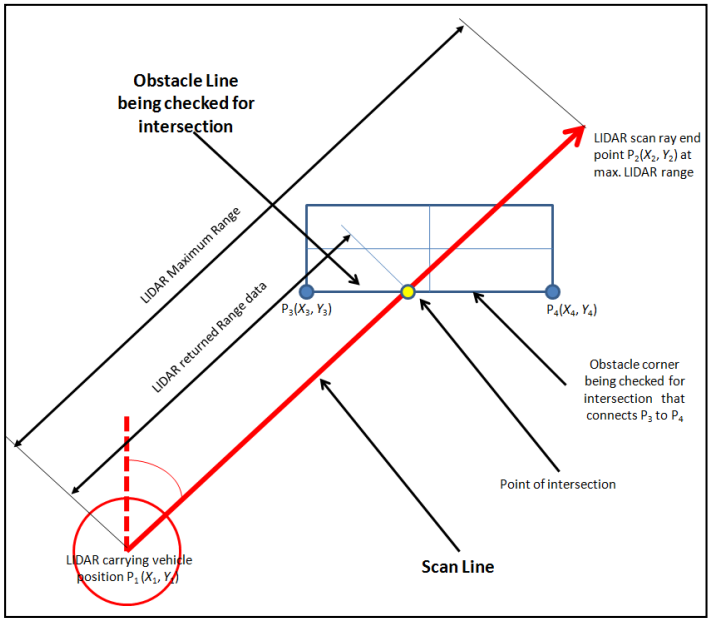

Fig. 4. Scan Line L_1 and Obstacle Corner Line Being Checked for Intersection L_2

5- Applying previous equations with $L_1(P_{1(X_1,Y_1)}, P_{2(X_2,Y_2)})$ as the scan line and $L_2(P_{3(X_3,Y_3)}, P_{4(X_4,Y_4)})$ (as obstacle corners line for each obstacle corner line yields a set of points.

6- In case $P_x = 0$ and $P_y = 0$ this means that scanning line does not intersect with that obstacle line(s) i.e. LMS should return max range value

7- If $P_x > 0$ and $P_y > 0$ then calculate the distance between intersection point and LMS carrying vehicle current position.

8- Check if scanning line intersecting with more than one corner line of the obstacle, see figure 5, if true reject the value.

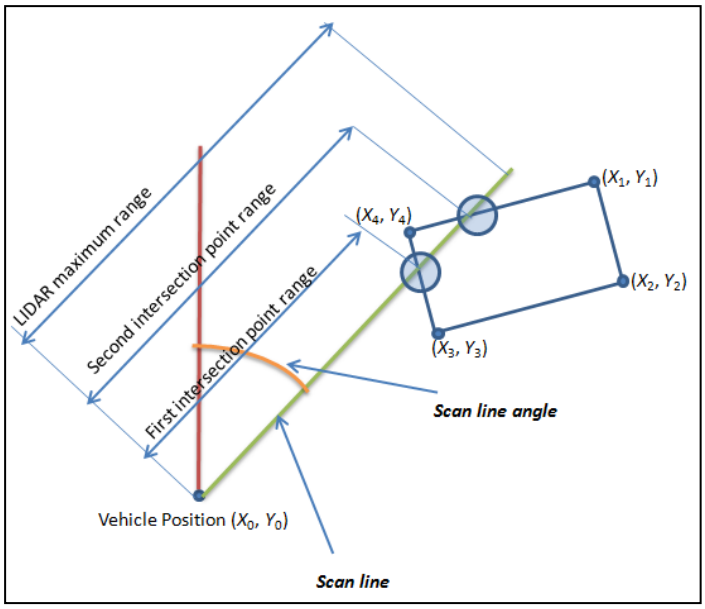

Fig. 5. Scanning Line Intersecting with More Than One Corner Line of the Obstacle

9- Check if the scan line extension on 180° intersect with an obstacle, see figure 6.

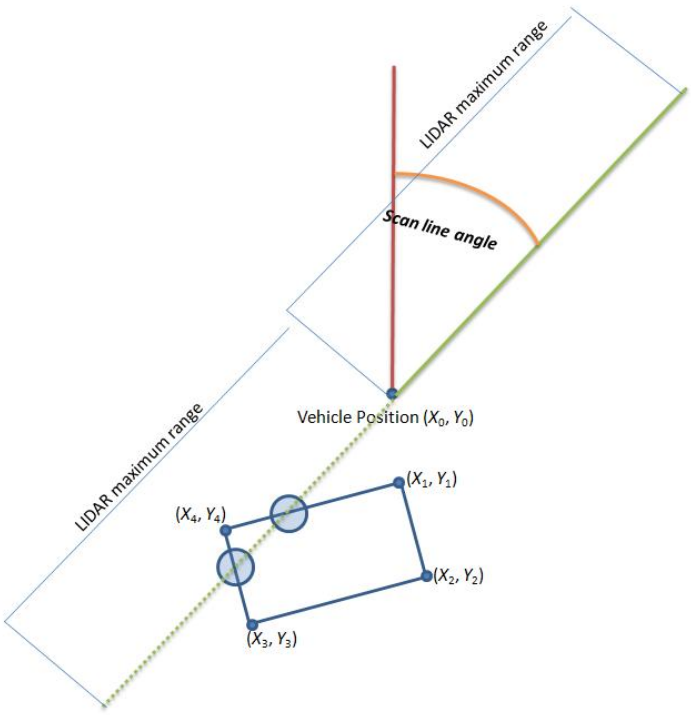

Fig. 6. Scan Line Extension on 180° Intersect With an Obstacle

10- check if the scan line intersects with the extension of the obstacle corner it may mistakenly show that there is an intersection point see figure 7.

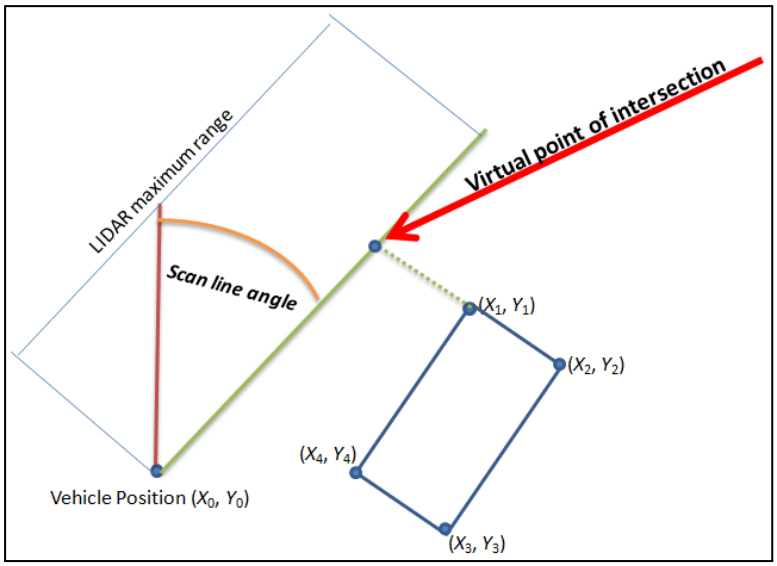

Fig. 7. Scanning Line Intersecting With the Extension of the Obstacle Line

11- If the value calculated has not been rejected in steps 8,9,10 add the value calculated in step **7** to the data matrix at the current scan angle.
12- Calculate detecting ship & obstacle(s) new position (x,y) based on current position, heading and speed.
13- Calculate new scan line current angle where: new scan line current angle= scan line current angle + LMS angular resolution
14- Repeat steps from 2 till simulation is complete.

In order to obtain a full scan matrix the LMS simulation algorithm should be repeated changing scan line angle from 0 to 360° with angle step β (Angle between two consecutive LMS shoots) and with each obstacle.

4. LMS Angular Resolution Effect:

The angular resolution is the angle between two consecutive LMS shoots during scanning; it is a key specification in the LMS since it governs the level of details it can detect.
This paper introduces a formula to calculate the possibility of a LMS to detect an obstacle based on obstacle geometrical shape, used LMS angular resolution and distance between LMS and obstacle geometrical center. It worth mentioning that during the verification trial when the obstacle faces the LMS with two points only the formula is valid by replacing the center of geometry by the middle point between the two subjected points.

From figure 8 the obstacle subjected part geometric center distance to LMS represents an average distance that considers all points of the subjected part.
The distance between LMS and subjected part geometric center $\boldsymbol{L}$ could be used to calculate the distance between two consecutive LMS rays at range equal to $\boldsymbol{L}$ using the following approximation since the Angular Resolution is comparatively a small angle[5].

$$\boldsymbol{W = L.\beta} \tag{7}$$

Possibility of detection is the result of division of $\boldsymbol{O}$ by $\boldsymbol{W}$ which yields the following equation:

Possibility of detection $\boldsymbol{D} = \frac{\boldsymbol{O}}{\mathbf{W}} = \frac{\boldsymbol{O}}{\mathbf{L.\beta}}$ (8)

Since $\boldsymbol{L}$ or $\boldsymbol{O}$ cannot be controlled by system designer, so to ensure detecting obstacle of minimum width $\boldsymbol{O}$ at an appropriate distance $\boldsymbol{L}$, for $\boldsymbol{D = 1}$.

$$\boldsymbol{\beta} = \frac{\boldsymbol{O}}{\boldsymbol{L}} \quad (9)$$

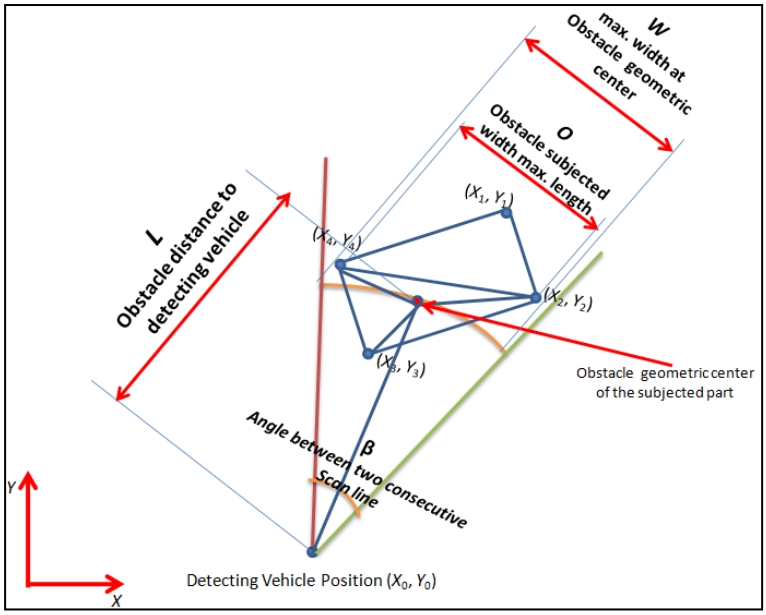

Fig. 8. Angular Resolution and Obstacle Size Relation to Possibility of Detection

5. RESULTS

In order to reduce the effect of movement of the vehicle carrying the LMS on the simulated data, the position of the vehicle is stored four times during scan i.e. when scanning beam is at 0º, 90º, 180º, 270º and each point detected is converted from local vehicle polar co-ordinates to universal Cartesian co-ordinates and then corrected based on interpolation of the scanning beam angle and positions stored, however the effect was found to be relatively small in the current simulated data this is due to the high LMS scanning speed and relatively low moving obstacle speed.

The simulation was carried out using a core I7 processor running windows 7, the variable used are all single precision, the simulation of three obstacle moving detected by a moving vehicle took on average 227 millisecond for a one complete 360° scan with angle step β=1° and Single Shot Time=10 millisecond.
For the same above case with only two obstacles and one obstacle the average san time was 207 and 182 millisecond respectively.

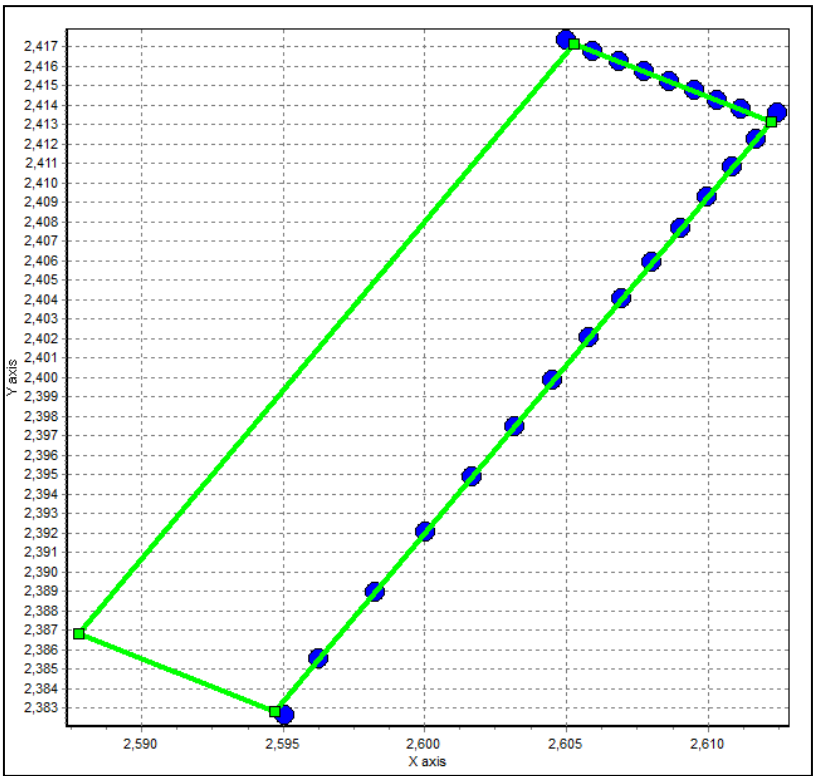

Fig. 9. Real Position in Red, Simulated detected Point(s) in Blue of a 35m Long, 8m Wide Stationary Obstacle, 62 m Away , 235° bearing detected by a stationary vehicle using a 1° LMS Angular Resolution

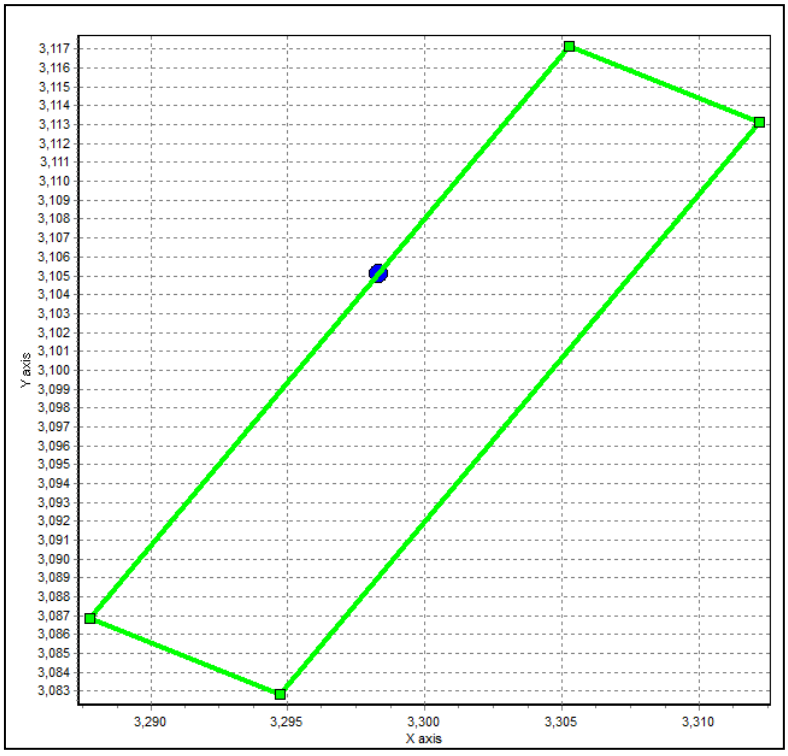

Fig. 10. Real Position in Red, Simulated detected Point(s) in Blue of a 35m Long, 8m Wide Stationary Obstacle, 1500 m Away 1° LMS Angular Resolution

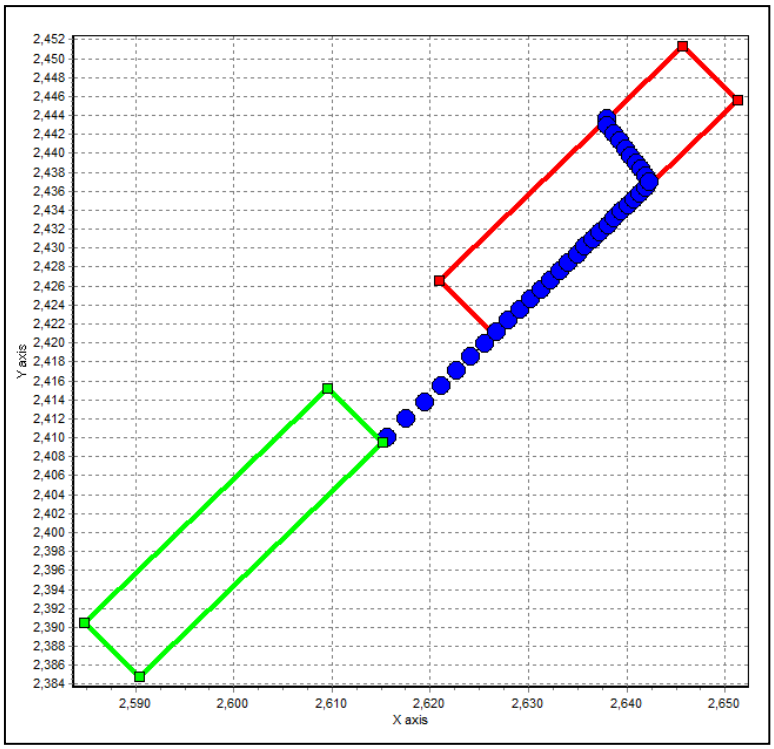

Fig. 11. Initial Position in Red ,Final Position in Green, Simulated detected Point(s) in Blue of a 35m Long, 8m Wide, 50 m away, 270 ° bearing, Moving at 30m/S , Heading 45º, Detected By LMS Mounted on Moving Vehicle Moving at 20m/S , Heading 90º ,1° LMS Angular Resolution , 10ms Single Shot Time.

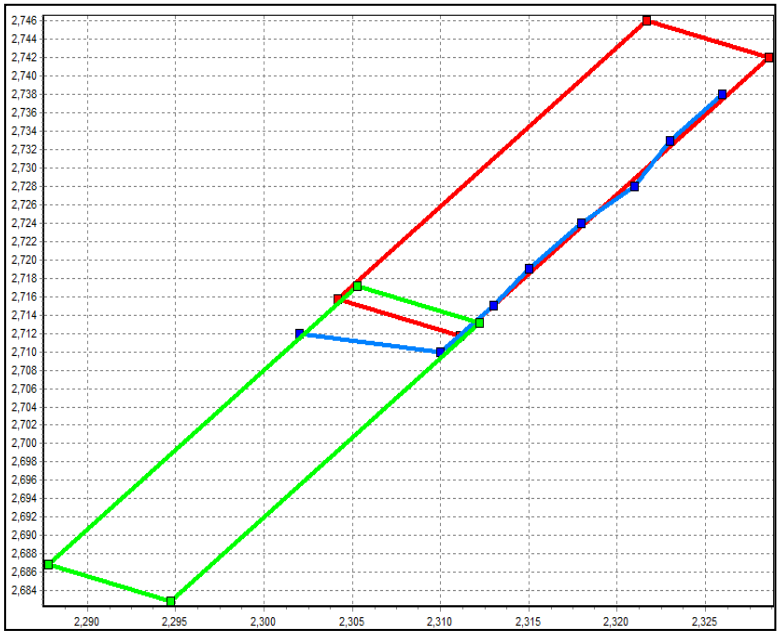

Fig. 12. Initial Position In Red, Final Position in Green, Simulated Points In Blue of a 35m Long, 8m Wide Obstacle Moving at 20m/S , Heading 30º, Detected by LMS Mounted On Stationary Vehicle , 1° LMS Angular Resolution , 10ms Single Shot Time.

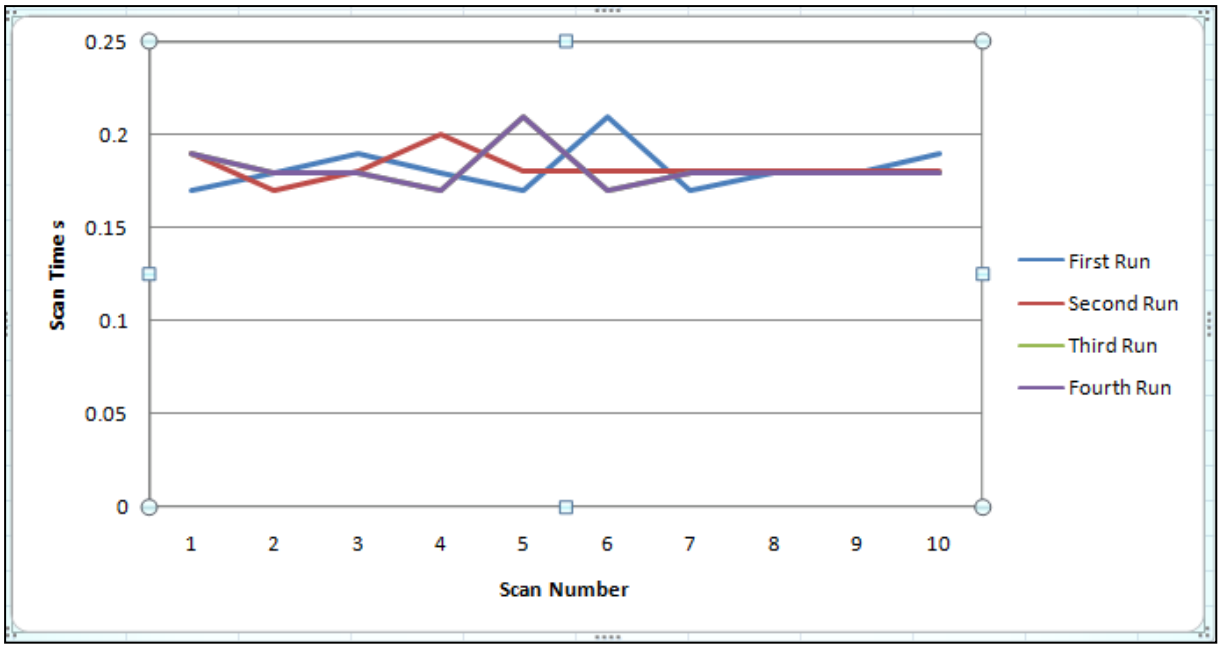

Fig.13. Simulation time of 10 complete 360° scans for one moving obstacle detected by a moving vehicle carried out using a pc with core I7 processor, the variable used are all single precision, with angle step β=1° and Single Shot Time=10 millisecond.

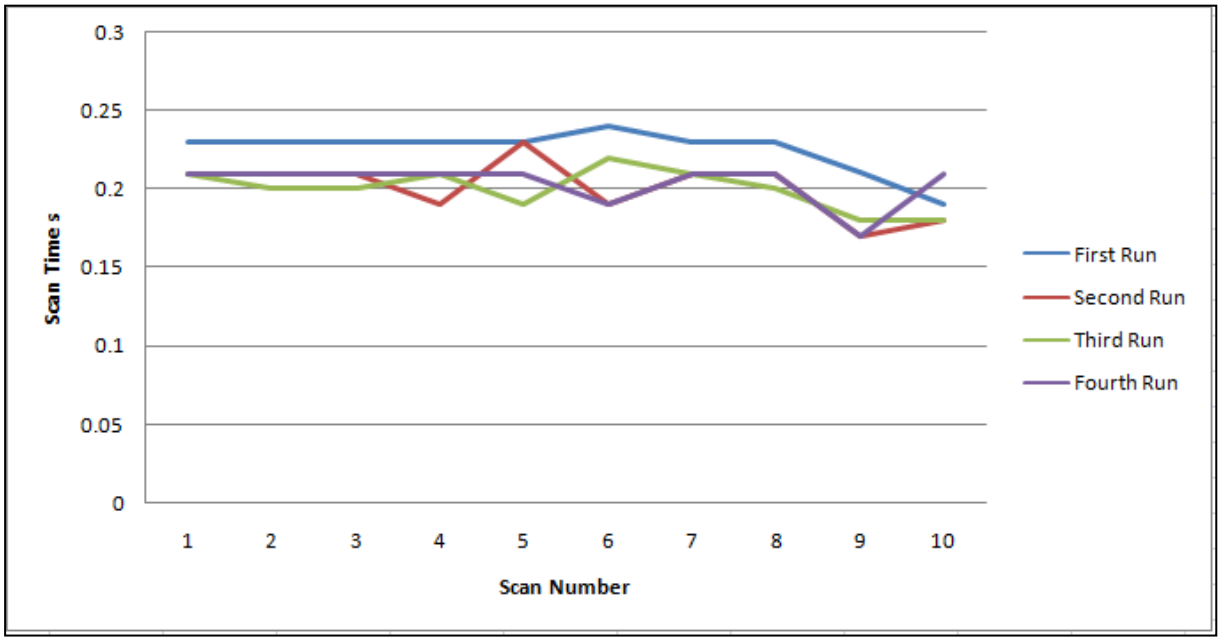

Fig.14. Simulation time of 10 complete 360° scans for two moving obstacle detected by a moving vehicle carried out using a pc with core I7 processor, the variable used are all single precision, with angle step β=1° and Single Shot Time=10 millisecond.

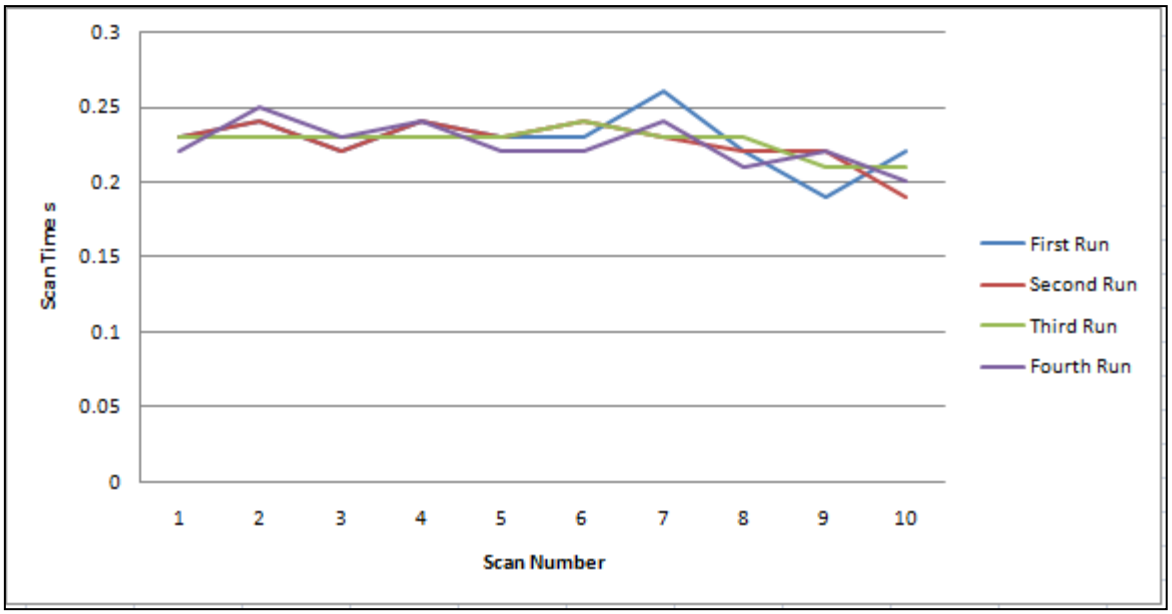

Fig.15. Simulation time of 10 complete 360° scans for three moving obstacle detected by a moving vehicle carried out using a pc with core I7 processor, the variable used are all single precision, with angle step β=1° and Single Shot Time=10 millisecond.

6. Conclusion

It is important to carefully choose Angular Resolution when integrating LMS into an obstacle sensing system to enable the system to detect obstacle at suitable distance.

10. Future Work

Integrating a mathematical model into the suggested algorithm to simulate the obstacle and detecting vehicle dynamic behavior (turning, acceleration, deceleration …)

5. References

[1] Deschaud, J.E., Prasser , D., Dias, M.F., Browning, B., Rander,P. , "Automatic Data Driven Vegetation Modeling for LIDAR Simulation," ", *U.S. Army Engineer Research and Development Center, (ERDC) under cooperative agreement "Fundamental Challenges in World and Sensor Modeling for UGV Simulation" ,(Number W912HZ-09-2-0023),2012*

[2] Gryaznov , N. and Lopota , A., "Computer Vision for Mobile On-Ground Robotics" *Proceedings of 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM , 2014*, pp. 1376-1380.

[3] Christer Ericson, *Real Time Collision Detection*, 1st. ed., Vol. 1, Elsevier, San Francisco, 2005, pp. 6-10.

[4] National center for geospatial intelligence standards, "Light Detection and Ranging LIDAR Sensor Model Supporting Precise Geopositioning", 1/8/2011, USA, 2013, pp. 25-29.

[5] Peinecke, N., Lueken, T., Korn, B.R., "LIDAR simulation using graphics hardware acceleration," ", *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th"*, Oct 26-30, 2008, , pp. 4.D.4-1 - 4.D.4-8,

[6] De Berg, M. (ed.), Computational Geometry, 3rd. ed., Springer, New York, 2000, pp. 19-29.