Virtual force field based obstacle avoidance and agent based intelligent mobile robot

Saurabh Sarkar[*a], Scott Reynolds[b], Ernest Hall[a]
[a]Dept of Mechinical Engineering, University of Cincinnati [b]Dept. of Computer Science, Univ. of Cincinnati
2600 Clifton Ave., Cincinnati, OH USA 45221-0072
Internet: http://www.robotics.uc.edu/

ABSTRACT

This paper presents a modified virtual force based obstacle avoidance approach suited for laser range finder. The modified method takes advantage of the polar coordinate based data sent by the laser sensor by mapping the environment in a polar coordinate system. The method also utilizes a Gaussian function based certainty values to detect obstacle. The method successfully navigates through complex obstacles and reaches target GPS waypoints.

Keywords: Virtual force field, laser ranger finder, agents.

1. INTRODUCTION

Real-time obstacle avoidance algorithms are important for mobile robots. Khatib demonstrated the real-time obstacle avoidance with his concept of artificial potential field in 1985[1]. Moravec and Elfes came out the widely popular concept of certainty grids. It is especially useful for data accumulation and sensor fusion [2].

Borenstein and Koren developed the Virtual Force Field by merging the last two concepts [3]. A certainty grid divides the robot's work area into small cells. Each cell then assigned a certainty value. The certainty value is the measure of confidence of an obstacle being in the cell. The greater the certainty values of the cell the more probability of an obstacle being in the cell.

The moving robot keeps updating the certainty values in the grid around itself. Each of the cells applies a repulsive force on the robot and pushes it away from cell. The force is proportional to the certainty value of the cell and inversely proportional to square of distance between the obstacle and the robot.

$$F(i,j) = \frac{F_{cr} C(i,j)}{d^2(i,j)} \left[\frac{x_t - x_0}{d(i,j)} \hat{x} + \frac{y_t - y_0}{d(i,j)} \hat{y} \right] \quad (1)$$

Where

F_{cr} = Force Constant
$d(i,j)$ = Distance between robot and cell (i,j)
$C(i,j)$ = Certainty Level of cell (i,j)
x_o, y_o = Robot's present coordinates
x_i, y_j = Coordinates of cell (i,j)

The resultant repulsive force is the vector sum of all the repulsive forces from the cells

* sarkarsh@email.uc.edu

$$F_r = \sum_{i,j} F_{i,j} \tag{2}$$

Simultaneously the robot is acted upon by an attractive force, which pulls the robot towards the target. The attractive force F_t originates from the target. The target may be waypoint whose coordinates are known to the robot. The magnitude of the force is given by

$$F_t = F_{ct}\left[\frac{x_t - x_0}{d_t}\hat{x} + \frac{y_y - y_0}{d_t}\hat{y}\right] \tag{3}$$

Where

F_{ct} = Force Constant (attraction to target)

d_t = Distance between target and the robot

x_t, y_t = Target coordinates

The final resultant force **R** is the vector sum of resultant and repulsive forces.

$$R = F_t + F_r \tag{4}$$

The direction of the resultant R, δ where $F_r = |F_r|(\cos(\delta) + i.\sin(\delta))$., is used to determine the steering rate command Ω of the robot.

$$\Omega = K_s[\min(\theta - \delta)] \tag{5}$$

Where

K_s = Proportional constant for steering (in sec^{-1})

θ = Current direction of travel

min(a-b) is the shortest rotational difference between two angles. The result is always between -180 and 180.

Some modifications have been made in the original algorithm to make it more suited for use with a laser scanner sensor. A novel feature of this approach is dividing the robot's work area into radial sectors instead of using a grid system. The sectors are 0.5 degrees wide and their origin is located at the center of the laser sensor. Another novel feature

is the equation used to determine the certainty value of each of these radial sectors. The details of each of the contributions will be described in the following sections.

2. SYSTEM OVERVIEW

Our robot system is designed as a distributed multi agent system. Multi agent systems have been identified by many including[4] as a promising approach to software engineering in a complex domain. Our system is designed to be open and always on. In these systems, software modules are modeled as autonomous agents. In our system, each sensor is represented as an agent that communicates with the central summation agent module. The individual agent acts as if it is the sole agent on the system and it is responsible for where the robot goes. For instance, the GPS sensor agent will always return to the summation module the angle correction the robot must take in order to drive down a straight line to the next way point. In this paper, we present a simple two agent system in which each sensor agent makes its decisions and demonstrate how these decisions are fused into one by the summation module.

Each sensor agent perceives and understands only a limited part of the environment in which the robot exists. This allows each agent to be solely focused on what it is trying to achieve. In this system, a sensor can fail and our robot will still be able to operate. This approach greatly simplifies the protocol in which the agents communicate and allows for a simple yet robust system.

Figure 1Bearcat Cub the robot using the algorithm

3. LASER RANGE SCANNER SYSTEM

3.1 Working principle

Sonar sensors were predominant in mobile robots before the laser range scanner became commercially viable. Laser scanners are more reliable and give much better resolution compared to the sonar. Laser range scanner works on a similar time of flight principle that is used by sonar or radar. It shoots out a highly focused laser beam into space. When the beams hits an obstacle it gets scattered and part of it returns to the scanner. The sensor detects both the changes in amplitude and phase and this is used to determine the distance the beam was reflected back calculating the time difference between release of the beam and arrival of the reflected beam.

The sensor pans its laser beam in a 180 degree span using a rotating mirror. As the laser sensors shoots out laser beams at different angles it calculates distance of obstacle laying in its field of view. Thus a 2-dimensional map of obstacles in-front of the laser sensor can be obtained.

The laser sensor used here is a SICK™ LMS 200. It uses as infrared laser beam of 835 nm wavelength. It has a span radius of 180 degrees. The resolution of scan be 0.25, 0.5, 1 degree. It can communicate with the computer using a RS 422 or 232 ports. The data transfer rate varies from 96 K to 500 K Baud.

Figure 2 SICK LMS 200

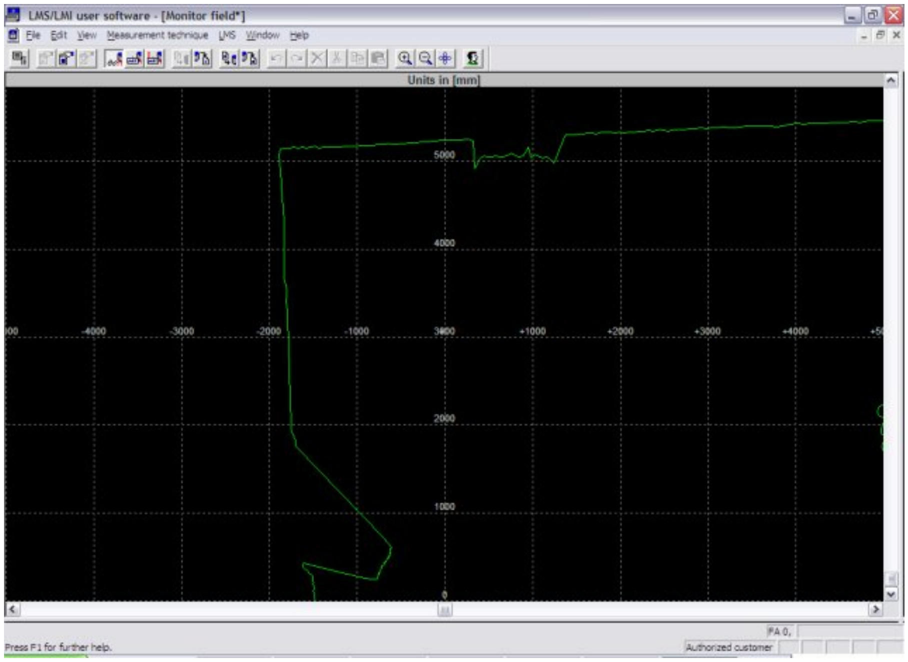

Figure 3 Laser measurement system output

3.2 Algorithm

The space in-front of the mobile robot is broken into radial sectors. Each sector is 0.5 degrees in width. Each of the sectors is given a certainty value. The certainty value is depends on a Gaussian function of distance at which the obstacle is from the robot. The reading from the laser range finder gives distances in each of the sector.

$$C(\theta) = C_t \frac{e^{-(x(\theta)-\mu(\theta))^2 / (2\sigma^2)}}{\sigma\sqrt{2\pi}} \quad \text{if } x(\theta) > \mu(\theta) \tag{6}$$

$$C(\theta) = C_t \quad \text{if } x(\theta) \le \mu(\theta)$$

Where

C_t = Constant

$C(\theta)$ = Certainty value of sector at θ

$x(\theta)$ = Distance of obstacle at sector θ

$\mu(\theta)$ = Mean value at sector θ

σ = Standard deviation

By using $\mu(\theta)$ the shape of the mean curve around the robot can be controlled. It can be either any 2-D shape like semi-circle or semi-ellipse. This will enable us to control how near to an obstacle a robot should be when passing it.

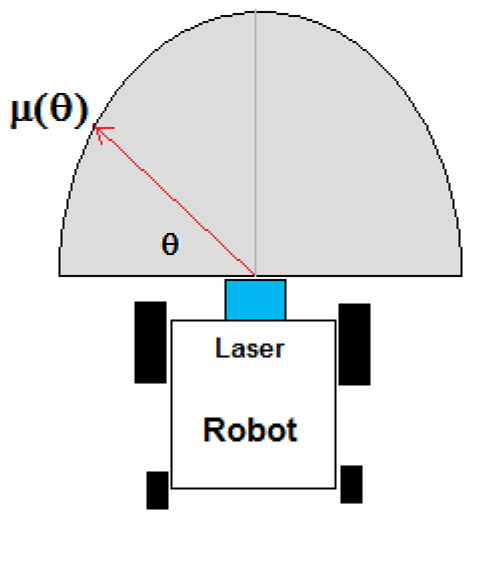

Figure 4 Mean Curve

As the robot keeps moving ahead it updates the certainty values of each sector. Each of the sectors applies a virtual repulsive force on the robot. The magnitude of the force depends on the certainty value of the sector.

$$F(\theta) = F_{ct} C(\theta) \tag{7}$$

Where

F_{ct} = Force constant

$C(\theta)$ = Certainty of sector θ

The resultant repulsive force is the vector sum of all the repulsive forces from the cells

$$F_r = \sum_{\theta} F(\theta) \tag{8}$$

The direction of the resultant steering angle is determined from the resultant force. The angle of the resultant force is given by δ where $F_r = |F_r|(\cos(\delta) + i.\sin(\delta))$.

$$\psi = \delta(+)\beta \tag{9}$$

Where

ψ = Steering angle of robot with respect to current heading

δ = Angle of resultant repulsive force

β = Constant with a value of $\pi/2$

$\alpha(+)\beta$ is a special function which equals $\alpha + \beta$ when α is less than $\pi/2$ and $\alpha - \beta$ when α more than equal to $\pi/2$

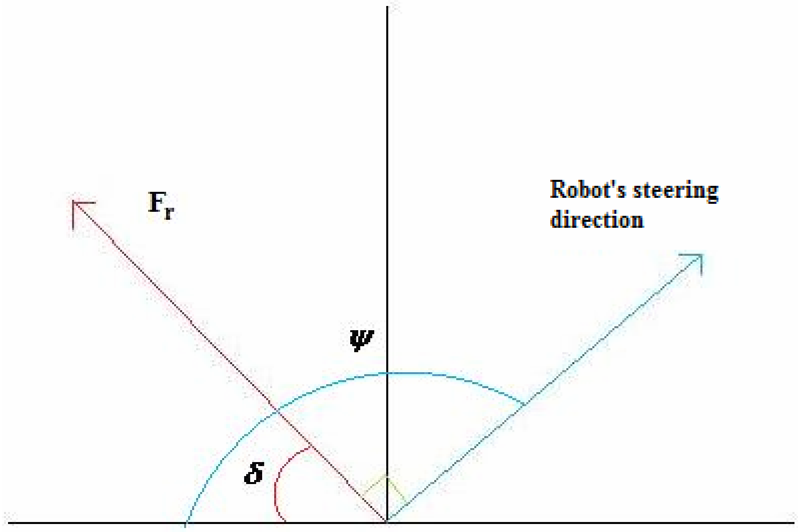

Figure 5 Determining steering angle when $\delta < \pi/2$

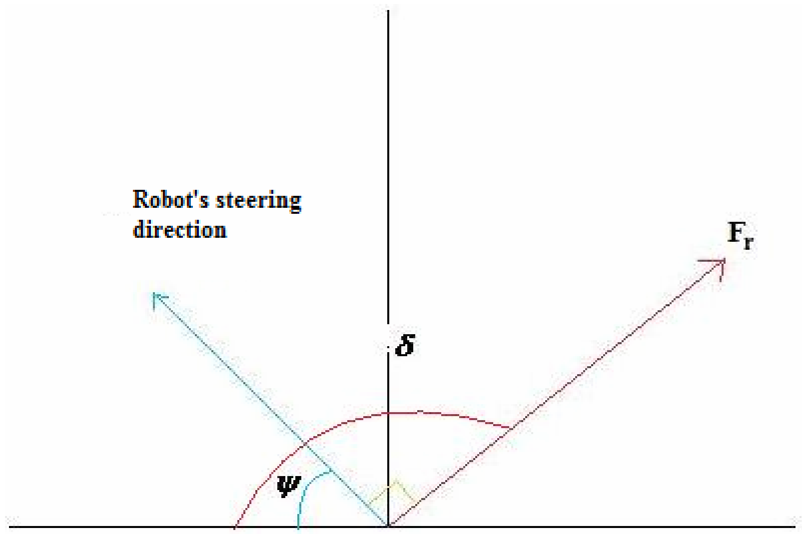

Figure 6 Determining steering angle when $\delta > \pi/2$

3.2.1 Special cases

The value of δ varies under certain special cases.

$$\delta = 0 \text{ If } F_r < \gamma \tag{10}$$

Where
γ = Threshold.
This is for the case when there is no obstacle in front of the robot. By setting $\delta = 0$ the final steering angle the resultant steering angle becomes $\pi/2$ and the robot does not change direction.

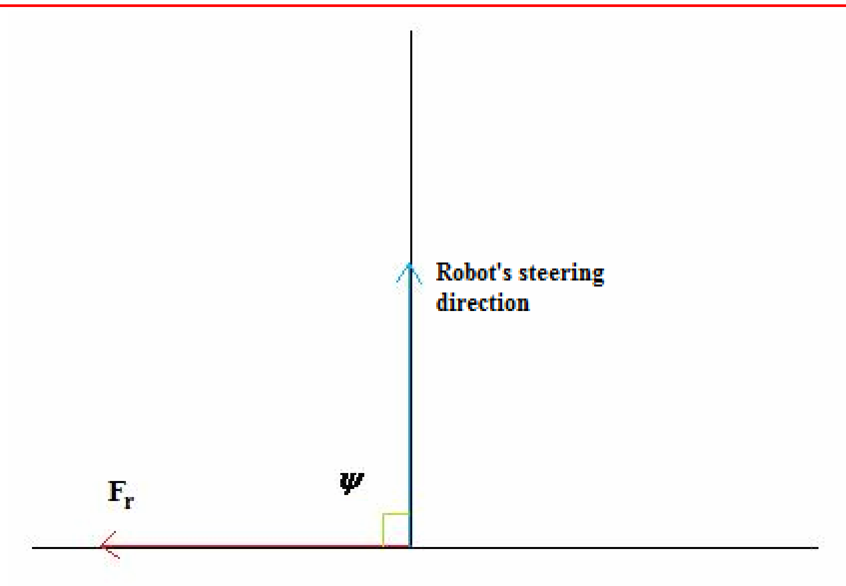

Figure 7Steering direction when $\delta = 0$

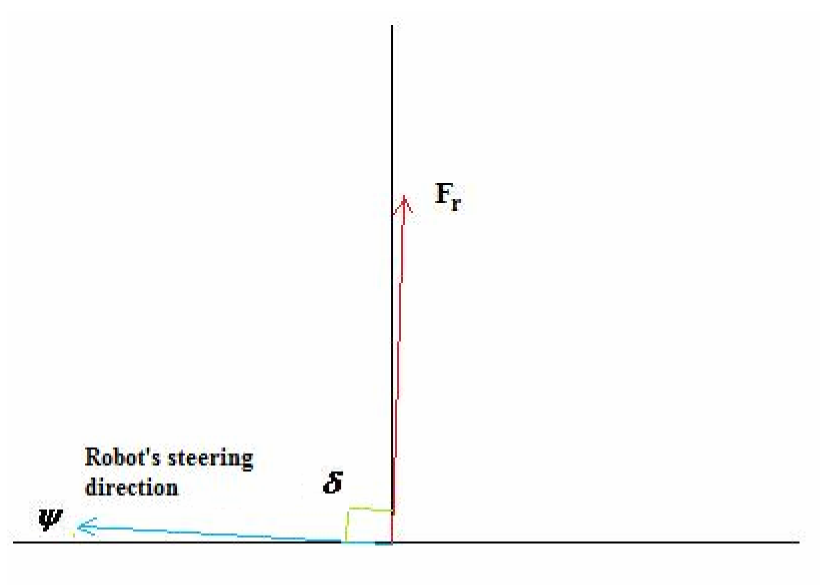

Figure 8 Steering direction when $\delta \approx \pi/2$

Simultaneously the robot is acted upon by an attractive force, which pulls the robot towards the target. The attractive force F_t originates from the target. The target may be waypoint whose coordinates are known to the robot. The magnitude of the force is given by

$$F_t = F_{ct}\left[\frac{x_t - x_0}{d_t}\hat{x} + \frac{y_y - y_0}{d_t}\hat{y}\right] \tag{11}$$

Where

F_{ct} = Force Constant (attraction to target)
d_t = Distance between target and the robot
x_t, y_t = Target coordinates

The final resultant force **R** is the vector sum of resultant and repulsive forces.

$$R = F_t + F_r \tag{12}$$

The direction of the resultant R, $\delta = R/|R|$, is used to determine the steering rate command Ω of the robot.

$$\Omega = K_s\left[\min(\theta - \delta)\right] \tag{13}$$

Where
K_s = Proportional constant for steering (in sec^{-1})
θ = Current direction of travel

min(a-b) is the shortest rotational difference between two angels. The result is always between -180 and 180.

4. GPS SENSOR MODEL

Our robot system is designed to find a path to various way points using only local information. Our robot uses a GPS system and a compass act as a soft sensor to determine the heading correction needed from the current position to reach the way point in a straight line.

The robot reads in the GPS information from the sensor and calculates the heading change needed based on the heading it is currently on and the heading needed from the current position to drive straight to the way point. This information is then sent to our summation module. This module is responsible for fusing them together.

5. SUMMATION MODULE

This module is responsible for making the final decision on the heading correction the robot undertakes. Each sensor on our robot makes a decision on the change and sends its local decision to the summation module. The summation module receives this information and fuses it together by mathematically combining the sensors suggestions. The suggestion each sensor model sends is in the form of wheel commands, w_i, where *i* is the corresponding robot wheel . The basic formula is below

$$w_i = \sum_{s=1}^{=k} \lambda_s w_{s,i} \tag{14}$$

Where w_i is the wheel command for wheel *i, s* is the sensor model, *k* is the number of sensors λ_i is the weight applied to that sensor and $w_{s,i}$ is the wheel command for wheel *i* from sensor *s*.

This method allows for all data coming into the summation module to be fused together into one cohesive command to the motion controller. The λ_s for each sensor is determined through experimentation and it ranges from 0-1. When λ_s is set to one the sensor has full control over the robots actions, when λ_s is set to

0 it has no control over the robots actions. The λ_s works as a parameter to determine the behavior of the robot. A low λ_s on the GPS sensor will result in a less optimal path to the way point but stronger obstacle avoidance. And the converse is true for applying a small λ_s on the laser sensor.

The initial experiments with this formulation were mixed. It was determined that the sensors were fighting each and one sensor would essentially negate the other sensor. In particular this would happen when one sensor would indicate a change that was a negation of another sensor. This would then result in a fused wheel command that was near 0 even though the sensor had suggested had suggested a change in direction.

This causes major issues when the robot is approaching an obstacle and the laser suggests a hard turn in one direction and the GPS suggests a hard turn in the opposite direction. The end result is essentially no change in direction for the robot. This is illustrated in following figure.

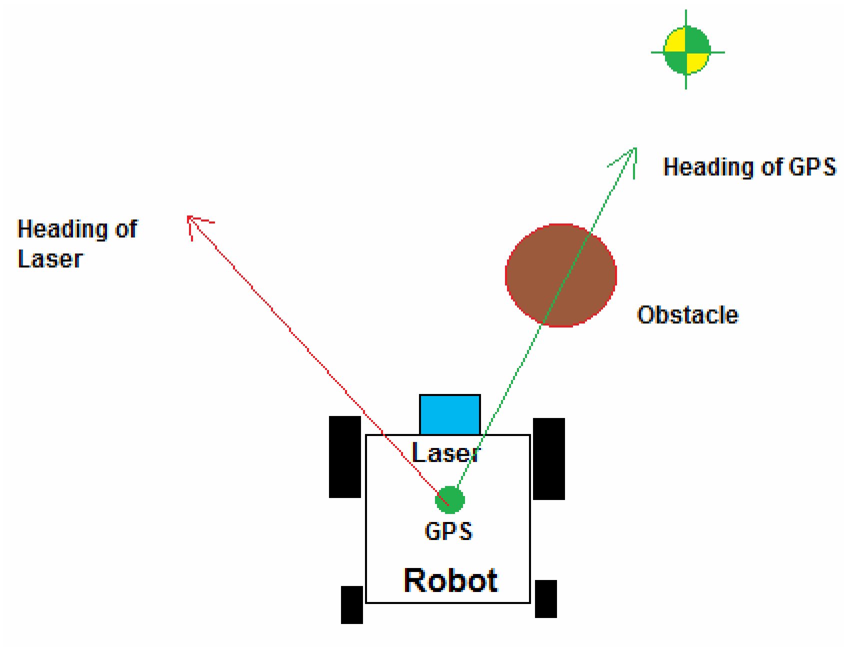

Figure 9 Sensor conflict

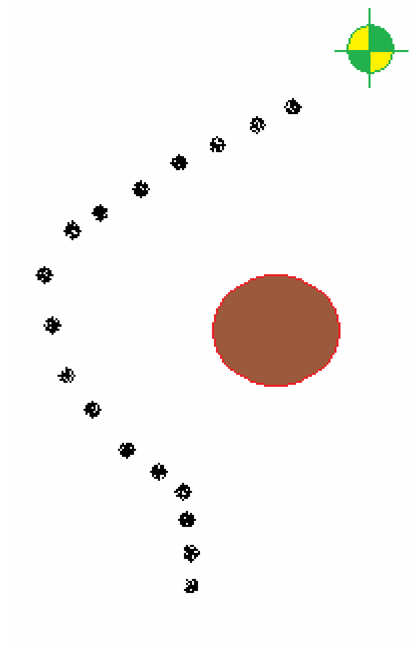

Figure 10 Path taken by robot

Our solution to this problem adds another level of decision to the summation module. The summation module has to look at the data provided by the two sensors and compare the values. If the wheel commands are not in the same direction, as shown above, then the decision is made to just use the laser sensors values and ignore the GPS sensor. This circumvents the issue of conflicting wheel commands and pushes the decision to find the path to the way point into the future.

This method allows the robot to use reactive control law to guide its motion and create a plan of action. All computation is based around what is happening now locally. Our system has removed the temporal dimension and greatly reduced the complexity of path planning. And it scales to any number of heterogeneous sensors.

6. TEST RESULTS

The algorithm was tested in obstacle course with various GPS waypoints. The robot running on the algorithm success navigated through the course. One of the obstacles was a GPS coordinate enclosed in a circular fence with two openings.

Figure 11 GPS waypoint enclosed in a fence

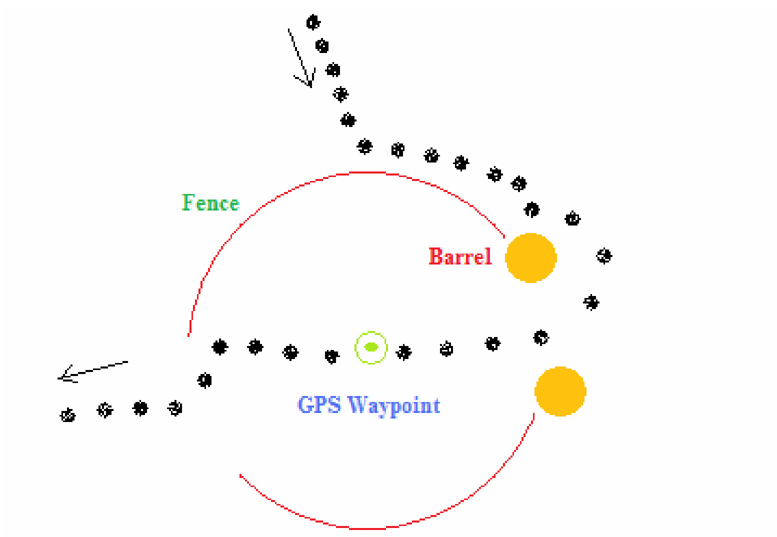

Figure 12 Approximate path taken by robot

7. CONCLUSION

Our work presented here presents a new method for obstacle avoidance. It utilizes the polar coordinate system which is far more suitable for laser range finders. This paper also presents our summation module which fuses sensors suggested motion commands into one direction. This method presents a more scalable approach and allows for any number of sensors.

REFERENCE

1. O. Khatlib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots:, *IEEE Int. Conf. On Robotics and Automation*, 500-505 (1985).
2. H.P. Moravec, “High resolution maps from wide angle sonar”, *IEEE Int. Conf. On Robotics and Automation,* 116-121 (1985).
3. J. Borenstein and Y. Koren, “Real-time Obstacle Avoidance for Fast Mobile Robots”, *IEEE*

Transactions on Systems, Man, and Cybernetics 19(5) 1179-1187 (1989).

4. F. Zambonelli, N.R. Jennings and M. Wooldridge, "Developing multiagent systems: The Gaia methodology", *ACM Trans. Software Eng. Mehtodologies* 12(3) 317-370 (2003).
5. D. Weyns, et al. "Environments for Multiagent Systems: State-of-the-Art and Research Challenges", *Environments for Mult-Agent Systems*, Springer Berlin/Heidelberg 1-47 (2005).