[1]

Multiple Autonomous Vehicle Solutions to Minefield Reconnaissance and Mapping

A. J. Healey, J. Kim
Naval Postgraduate School
Monterey, CA 93943
http://web.me.nps.navy.mil/~me/healey.html

ABSTRACT

Modeling and Simulation is an important tool for the evaluation of new concept systems. In particular, new system concepts are being developed for minefield reconnaissance and neutralization using robot vehicles. Also, with an emphasis on low cost, these systems are being focussed on multi-robot capabilities using fleets of similar and dissimilar vehicles in cooperative behaviors. The problems of operating in the very shallow water areas (VSW) are increased by the action of waves and currents and uneven bottom topography.

This paper will discuss the elements of modeling and simulation methodology for the study of system performance analysis in minefield reconnaissance and object mapping in Very Shallow Water (VSW) environments. Crawling and swimming vehicles are considered, although the focus is on the first. Vehicle locomotion models are proposed. Wave and current models are discussed by reference to other ongoing research. The modeling of object detection sensors, and vehicle navigation sensors are also given.

Using these principles given above, reference is made to the importance of two types of simulator - a graphics based visualization simulator that views the interactive behavior of robots and environmental objects, and a Monte Carlo low resolution simulator that allows the study of system effectiveness. In an example of a VSW operation with crawling vehicles, results are given that illustrates the effect of control logic parameters, on the time it takes to complete the reconnaissance mission. Also, other control parameters are studied including the effect of changes in the detection range of the primary sensor.

Key words: Robotics, Search, Mines, Control, Autonomous Systems

1. INTRODUCTION

As the U.S. Navy looks for low cost solutions to minefield reconnaissance, the use of modeling and simulation technologies become paramount in the search for tactics for the use of multi agent robotic platforms. Both swimming and crawling vehicles carry target and obstacle detection sensors to the search area. Navigation, targeting, communications, and detection errors lengthen the search process. Different strategies for the deployment of multiple cooperating robots are being studied using modeling and simulation efforts. At the heart of simulation models lie issues of vehicle motion control, target and obstacle detection/classification, obstacle and other vehicle avoidance, navigation, and system wide coordination.

Locating threat objects and mapping obstacles is one of the reconnaissance missions in very shallow water using crawling vehicles. Vehicles such as the Foster Miller Lemming[1] may be used employing tracks to provide locomotion and steering. The vehicles are equipped with detection sensors and navigate by dead reckoning with a compass as the heading reference. Odometry is obtained using a track speed sensor (encoder wheel on the motor shafts). Robot control algorithms have been shown to require both high and low level controllers, where the low level control takes the form of well known servo control,

[1] Australian-American Conference on MCM, Sydney, July 15, 1999

and the high level control is a sequence control of the mission, described by multiple cooperating Finite State Machines / Petri Nets, FSM/PN[2,3,4].

Generally, the authors have found that 2 forms of models are necessary to understand system operation. Firstly, a low resolution model is used where vehicle motion is simplified to a point velocity vector so that time step updates are relatively long. This model allows for rapid solution and testing of the control logic embedded in the operational FSM. Also, since navigation errors and target placement are random, multiple runs of the same scenario (i.e., Monte Carlo Simulation) are required to generate meaningful statistics of the performance. Secondly, a high resolution simulation including details of the lower level servo control modeling and obstacle avoidance algorithms is required to view robot behaviors. This type of simulation is also a time step simulation where the time step is much smaller and the models include dynamic responses, and will usually have a graphics based output for visualization.

2. VEHICLE MOTION MODELING

A reasonable model for a crawling vehicle is given by the following equations of motion[5].

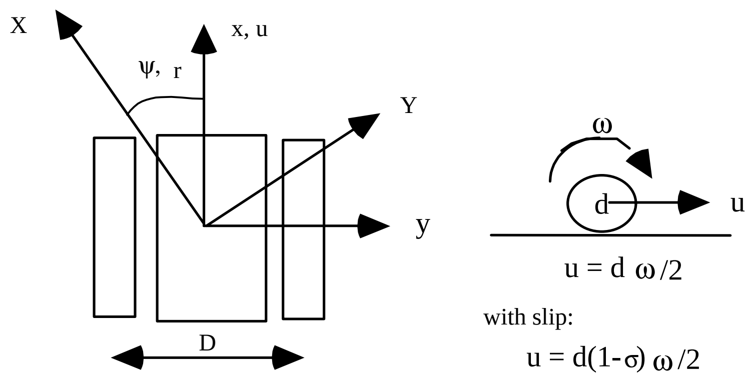


Figure 1

Forward Speed ,U and Rotational Rate, r are Related to the Average and the Differential Track Speeds

The forward speed, u, is the average of the two track speeds while the rotation rate is given by their difference. The remaining 3 equations are kinematic relations converting body fixed velocities into navigation frame motions and positional updates. *dt* is the time step used in an 'Euler' integration scheme. Without slip[6], the equations are,

$$
\begin{aligned}
&u(kdt) = 0.5*(\omega_l(kdt) + \omega_r(kdt))d \\
&r(kdt) = (\omega_l(kdt) - \omega_r(kdt))d / D \\
&\psi((k+1)dt) = \psi(kdt) + r(kdt)*dt \\
&X((k+1)dt) = X(kdt) + u(kdt)\cos(\psi(kdt)) \\
&Y((k+1)dt) = Y(kdt) + u(kdt)\sin(\psi(kdt))
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&r_{com} = K(\psi_{com}(t) - \psi(t)) \\
&u_{com} = \begin{Bmatrix} u_{\max} & \text{in transit} \\ u_{search} & \text{in search} \end{Bmatrix} \\
&\psi_{com}(t) = tan^{-1}\frac{X_k - X(t)}{Y_k - Y(t)} + \psi_{oa}(t)\ ; \\
&\textit{where } (X_k,\ Y_k) \textit{ is the coordinate of the next target}
\end{aligned}
\tag{2}
$$

While the right / left motor speeds are derived from

$$\omega_l(t) = 2u_{com} / d + Dr_{com} / d;$$
$$\omega_r(t) = 2u_{com} / d + Dr_{com} / d;$$

Control is assumed to have two commands, one for forward speed, u_{com}, and the second for rotational rate, r_{com}. The control signals are computed through inverse kinematics and then distributed to the two track motors. The motion control commands are linked to the guidance law using 'line of sight' or other guidance law[7], where r_{com}. is obtained proportionally to the heading error. These equations represent the high resolution model while the low resolution version reduces to

$$
\begin{aligned}
&y\,(kdt) - y_{com}(kdt) \\
&u(kdt) = u_{com}(kdt)
\end{aligned}
\qquad (3)
$$

Equations 3 suffice for the low resolution model for swimmers as well as crawlers.

High resolution modeling for swimmer vehicles is more complex - generally involving six degree of freedom models[7, 8, 9, 10], and wave / current effects[11]. The reader is refered to these other works for more detail.

3. DETECTION AND OBSTACLE AVOIDANCE SENSOR MODELS

In the absence of higher order sensor models, it is usual to consider the detection sensor as a device that detects targets that are located within a swath or disk, with equal probability over the area of the disc (a 'cookie cutter' model). Thus the probability of detection is uniformly distributed over the area of the disc. While this is not best representation of reality, it is used in the absence of a higher order model (probability of detect is a function of position in the disc). Detection probability p(r) is constant if a target appears inside the radius r < R. It is assumed that the sensor is located at the center of the vehicle body.

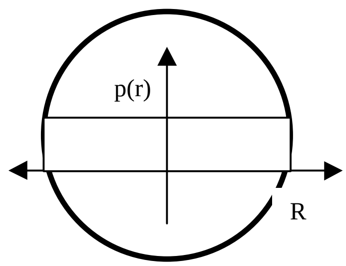


Figure 2. Detection Range Functions

3.1 Probabalistic Detection

A detection of a threat object occurs if the sensor detection area lies over the threat (2-D world assumptions). In the general case, given that event, two hypotheses are made.

H1: Target is present
H2: No target is present

Under H1, the two outcomes are
- H1a: Target is correctly detected (probability *p*)
- H1b: Target is not detected (probability (1-*p*)

Under H2, the two outcomes are
- H2a: False positive (probability q)
- H2b Correctly detected no target (probability (1-q)).

To represent these outcomes, two independent uniformly distributed random numbers (*k l)* in a range [0,1] are called once only. If H1 is true, $k < p$ is used to declare H1a. If H2 is true, $l < q$ is used to declare H2a. This random number call is only made once since multiple calls while the object lies in the detection circle, increases the probability of detect unduly.

In the case of side scan models, *p* and *q* depend on the distance, *r*, away from the vehicle centerline.

3.2 Obstacle Avoidance Detection and Command Arbitration

Obstacle avoidance is critical to the success of multi robot operations in the VSW. Using the example diagram in figure 3,

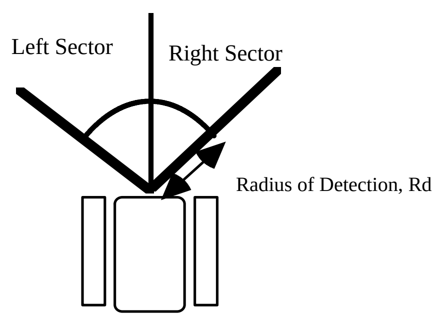


Figure 3 Obstacle Avoidance Sensor Model - Radius and a Sector of Sensitivity
Implemented with IR beams or a Camera / Sonar

A check is made obstacle after each step in the simulation to find the range from each vehicle to each target/obstacle. If any obstacle lies within the detection range of the avoidance sensor, some a new control behavior is entered. One such avoidance logic is given by the pseudo code below. This has been found to be successful under a variety of conditions appropriate to the low resolution model.

```
While obstacle Detect radius < Rd
        Rotate left; end;
        Move Forward one increment step
If BUG turned Left and Moved by One full step (1m.) turn right;
If BUG moved 1 step but not turned left, head to goal point;
Begin Searching or Dropping if BUG is within Goal Neighborhood;
```

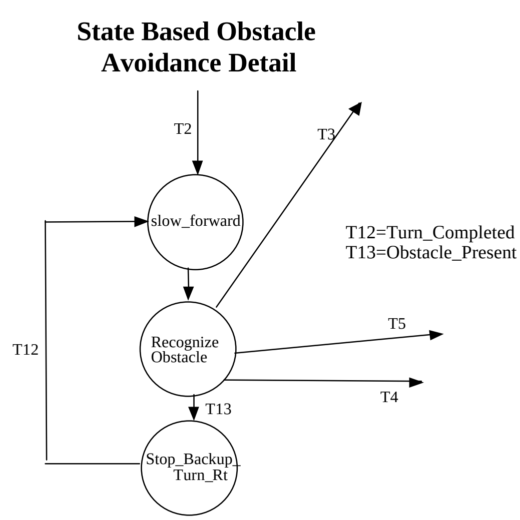
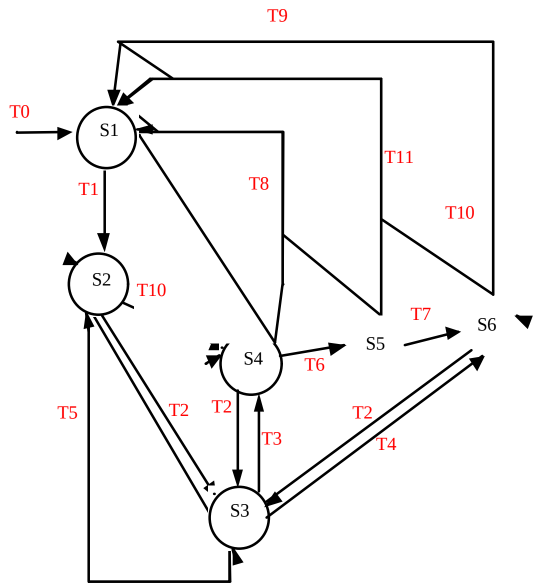


Figure 4. State Based Model Using Stop-Back-Turn Method

Alternatively, a 'state based' obstacle avoidance scheme can easily be implemented using the 'Stop-Back-Turn ' principle but this is slow as far as search performance is concerned. The transitions T4, T5, etc. in Figure 4 are linked to the operational FSM shown in Figure 5.

BUGS Canonical State



LIST of STATES

S1=Read_Next_Target_Location
S2=Do_Waypoint _Transit_to _Next_Target
S3=Do_Obstacle_Avoidance
S4=Do_Local_Search
S5=Perform_Pick_Up
S6=In_Transit_to_Pile

LIST OF TRANSITION SIGNALS

T0=Start
T1=Receive_Stimuli
T2=Obstacle_Detected
T3=Obstacle_Clear&&In_Local_Area
T4=Ostacle_Clear&&In_Transit_to_Pile
T5=Obstacle_Clear && In_Waypoint_Transit
T6=Target_Identified
T7=Pick_up_Done
T8=Time_Out_Searching_Exceeded
T9=Dropped_On _Pile
T10=At_Target_Area
T11=Time_exceeded_Pick_Up

Figure 5. Operational Finite State Machine For Autonomous Systems in Search and Locate Missions Taken from [12]

A more detailed obstacle avoidance model, suitable for the high-resolution simulator, arbitrates between heading commands that are taken from several different command modes. These depend on whether the obstacle is detected to the right or left side of the vehicle and what control mode the vehicle is in at the current time. For instance, in search, the heading command is taken from a randomized set of values, while in transit to a fixed point the heading command is taken from a guidance law (line of sight). The avoidance heading command is an increment that is added to the current heading when an object appears in the field of view. The event triggers a prioritizing signal, which increases the weight on the additive heading command summed in computing the final heading command. This method is suitable for use with any simulator as the Box in Figure 6 has no dynamics.

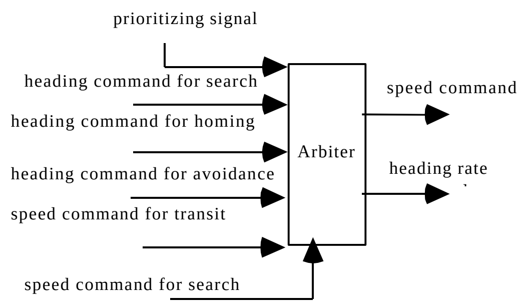


Figure 6. Arbitration Of Heading Commands Based On Control Mode And Obstacle Detection

4. SEARCH AND BOUNDING CONSTRAINTS

The theory of search has been studied for many years (see Koopman, 1954 for example [13]). Basically, a complete area coverage search at constant rate produces a linear coverage with respect to time, and a propbability of complete detection based on *p*. Multiple 'looks' increases the overall probability of finding all targets. In particular, if *m* views of a target are used, it is well known that the overall probability of detect is increased to

$$p_{overall} = [1-(1-p)^m]$$

Because of navigational errors, an overlapping complete area coverage at 3:1 overlap is often used which reduces the sweep rate by a factor of 1/3. At this point, a randomized search becomes attractive which has a detection function for the percentage detected [14] see figure 7. The value α, is called the characteristic search rate[14].

$$\frac{\bar{n}(t)}{n_0} = (1-e^{-\alpha p t})$$

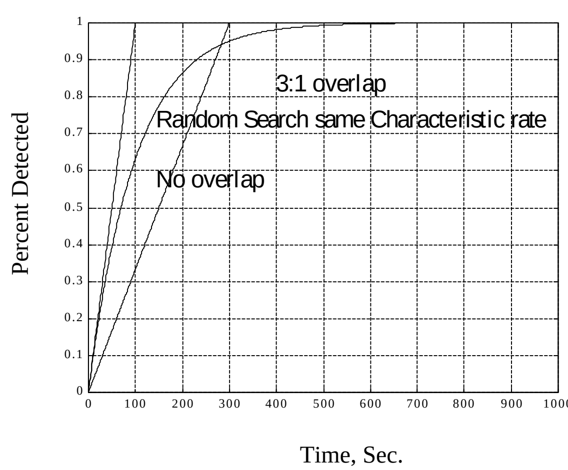


Figure 7. Comparison of random search with overlapping direct search and non-overlapping direct search showing that 3:1 overlap may be worse than random seaarch.[Results are for assumed uniformly distributed targets with independently random search paths with no obstacles, no other searchers, and no boundaries].

Randomization of the vehicle search is accomplished by a random change in heading after some time delay. If $\psi_r(i)$ is a random heading with values lying in the interval, [-90, 90] for each *i*, the mean of $\psi_r(i)$ is zero. Adding a basis heading, $\psi_{basis}(kdt)$ produces the heading command. Randomization occurs when for every k, i is taken to be i=mod(k/j), with j being a positive integer. The heading command is taken as

$$\psi_{basis}(kdt) + \psi_r(i)$$

j must be chosen to have a sufficiently large value otherwise the search does not progress. $j*dt$ lying betweem 5 and 20 seconds has been found to be a practical range for vehicles moving at 50 cm./sec..

At boundaries, the basis heading is changed to produce a complementary reflection at the boundary edge. Either a navigational system is used to trigger the basis change, or it may be assumed that an 'electronic fence' [17], produces a trigger signal indicating proximity to the boundary. Changing the heading basis is used for containment of vehicles inside the search area.

5. RESULTS

An example of the use of the low resolution model to reconnaissance and mapping of an approach lane is illustrated in Figure 8.

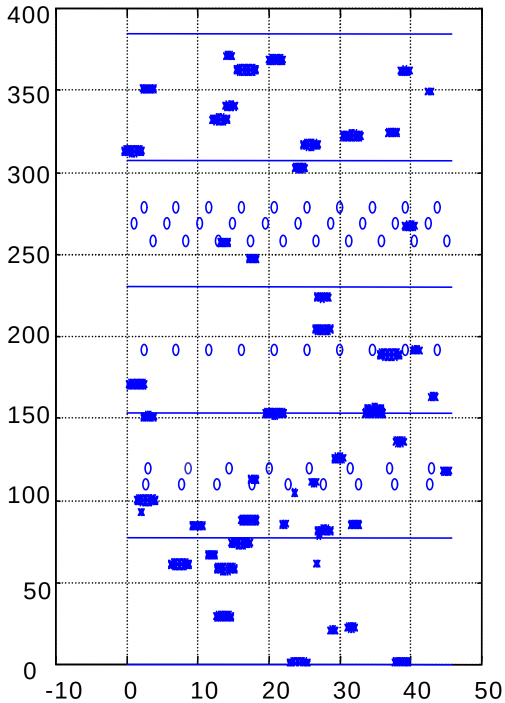


Figure 8. Example of reconnaissance lane with mines and obstacles. Obstacles are to be mapped and mines identified and located [14] Distances in meters. Circles are mines. Larger clumps are obstacles / rocks. The Obstacle and Threat Field are Unknown at the Start.

The field is to be searched by 25 vehicles moving at speeds of 50 cm./sec with obstacle detection ranges of 2-5 ft. and threat detection ranges of 2 - 5 ft. Assumed, is that each vehicle is capable of perfect navigation. Five vehicles are deployed into each of the five zones above that has been arbitrarily selected with no assumed prior knowledge of the presence of rocks and threats. The vehicles begin randomized searching in each zone. Using their navigational suites, the vehicles reflect off the zone boundaries using the change of basis heading rule. When an obstacle is detected, it is mapped by moving around its perimeter until it reaches the point where it began the mapping. As the threats are detected, they are located in the report file. Results are obtained by plotting the number of threats identified versus time. Figure 9 shows a typical path plot for an enlargement during the search process.

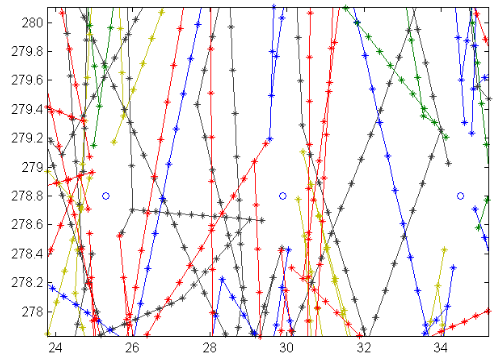


Figure 9. Random Paths Through the Field. Vehicles Stop/back/turn on Detection of Threats.

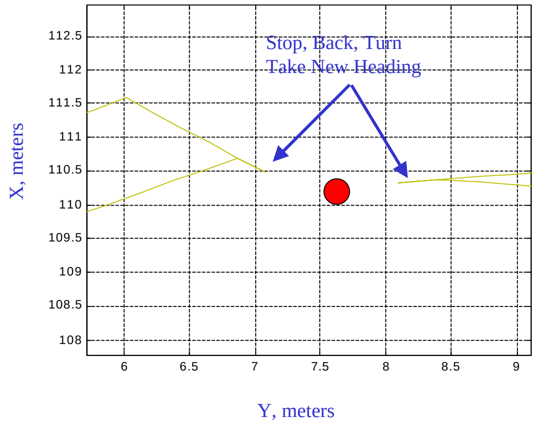


Figure 10 Path Plot During the Search Showing Threat Detection Response

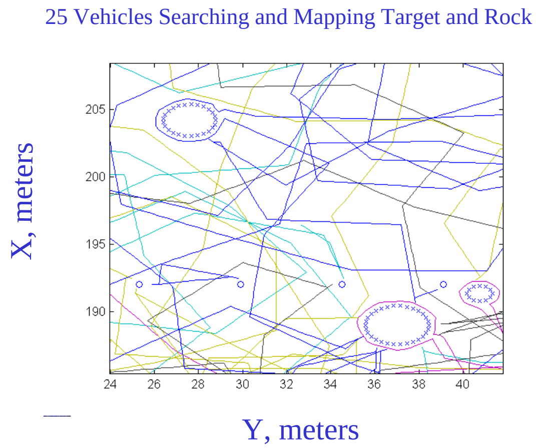
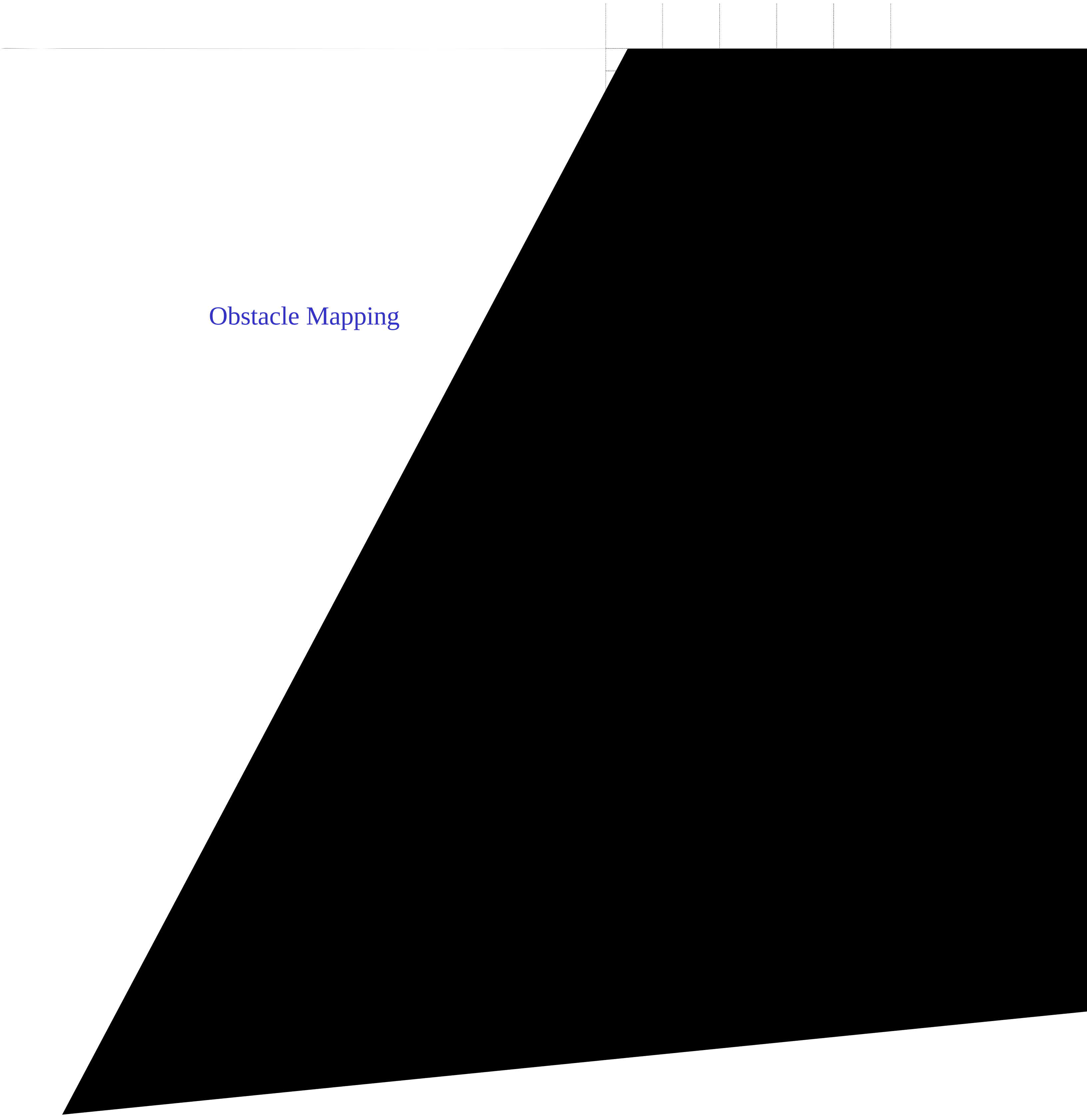


Figure 11 Vehicles Map Obstacles by Circling Using the O/A Algorithm Until Initial Position is Regained. Time Out Protects Against Deadlock.

In Figure 11, the process of obstacle mapping begins when a vehicle makes contact and the O/A algorithm keeps the vehicle searching around the extent of the obstruction. The initial contact location is stored, and when the vehicle returns to a close proximity of that location, the mapping is completed and a return to random search is triggered.

Effect of Sensor Detection Radius
2, 4 and 5 Feet



Obstacle Mapping

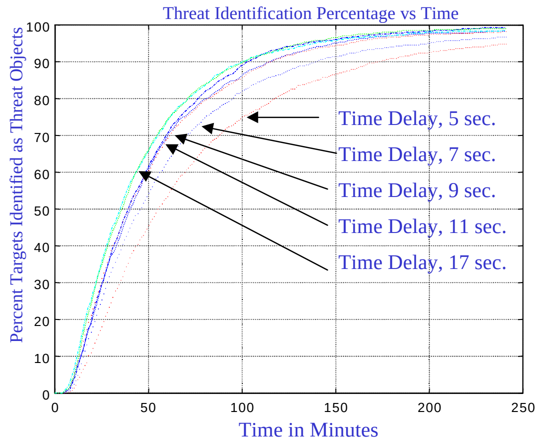


Figure 14 Effects of Randomization Update Time Delay on Threat Identification Process.

6. CONCLUSIONS

Modeling and Simulation methodologies are essential to the evaluation of multi vehicle usage for reconnaissance missions. Control logic parameters have a significant impact on the performance of the system operation. Since these models are non-linear and have parameters that are random, Monte-Carlo methods are needed with results for many replications of the same basic scenario to produce means and variances of detection rates. Through these models, control logic can be tested and operation tactics can be evaluated. Much more work is needed to determine if optimization of control parameters is possible.

7. REFERENCES

1. Foster Miller Inc., Waltham Mass.
2. Marco, D. B., Healey, A. J., McGhee, R.B., Brutzman, D. P.,"Control Systems Architecture, Navigation, and Communication Research Using the NPS Phoenix Underwater Vehicle" 6th International Advanced Robotics Program Workshop on Underwater Robotics, Toulon, France, March 27-29, 1996. http://web.nps.navy.mil/~me/healey/papers/iarp_96.pdf
3. Marco, D. B., Healey, A. J., McGhee, "Autonomous Underwater Vehicles: Hybrid Control of Mission and Motion", Journal of Autonomous Robots 3,169-186 (1996) http://web.nps.navy.mil/~me/healey/papers/jar.pdf
4. Healey, A. J., Marco, D. B., McGhee, R.B., "Autonomous Underwater Vehicle Control Coordination Using a Tri-Level Hybrid Software Architecture", Proceedings of the IEEE Robotics and Automation Conference, Minneapolis, April 1996, pp. 2149 - 2159. http://web.nps.navy.mil/~me/healey/papers/fin.pdf
5. Healey, A. J., Kim, J. "Basic Unexploded Ordnance Gathering System (Bugs): Robot Control And UXO / Minefield Clearance Characteristics *Report No. NPS-ME-96-005*, ***Naval Postgraduate School***, Monterey, California, 1996
6. Lewis, T., "Simulation of Small Robotic Vehicle Performance During UXO Gathering Operations Using Discrete Event Control", Report # NPS-ME-96-27, Naval Postgraduate School, Monterey, CA. September 1996.
7. Healey, A. J., Lienard, D., "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles", *IEEE Journal of Oceanic Engineering* Vol. 18, No. 3, July 1993 pp. 1-13
8. Chen, X., Marco, D. B., Smith, S. M., An, E. P., Ganesan, K., Healey, A. J.,"6 DOF Nonlinear AUV Simulation Toolbox", ***Proceedings IEEE, Oceans 97***, Halifax, Oct. 1997 IEEE CD-ROM 0-7803-4111-2 http://web.nps.navy.mil/~me/healey/papers/6dof_model.pdf
9. Fossen, T. "Guidance and Control of Ocean Vehicles" John Wiley and Sons, 1994, Chichester, UK

10. Yuh, J., "Modeling and Control of Underwater Vehicles" *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 6, 1990, pp 1475-1483
11. Riedel, J.S., Healey, A.J., "Shallow Water Station Keeping Of AUVS Using Multi-Sensor Fusion For Wave Disturbance Prediction And Compensation***", Proceedings of IEEE Oceeans*** **98**, Nice, France, September 28-30, 1998
12. Healey, A. J., Kim, Y., "Control of Small Robotic Vehicles In Unexploded Ordnance Clearance", ***IEEE Proceedings, ICRA 97***, Albuquerque, New Mexico, 1997. http://web.nps.navy.mil/~me/healey/papers/icra_paper_97.pdf
13. Koopman, B. O, "The Theory of Search. Pt. I and Pt. II", Operations Research , 4. 324-346., 1956
14 Stone, L, Theory of Optimal Search, Academic Press, 1975
15. Starr, J. A., "Multiple Autonomous Vehicles for Minefield Reconnaissance and Mapping", MSME Thesis Naval Postgraduate School, Monterey, CA, 1997
16. Healey, A. J., Pratt, S. M., McGhee, R. B., "BUGS: Basic UXO Gathering System Simulator Development" Naval Postgraduate School Report No. NPS-ME-94-008, December 1994
17. Jenkins, D. A., "BUGS: Basic Unexploded Ordnance Gathering System Effectiveness of Small Cheap Robotics ", *MSME Thesis*, Naval Postgraduate School, Monterey, CA. June, 1995.