The International
Journal of Modern Engineering
A Leading Journal For Modern Engineers

Volume 7, Number 2, Spring 2007

A Simulation Software for Autonomous Navigation of Unmanned Surface Vehicles Using MATLAB Environment

Ali Eydgahi
Department of Engineering & Aviation Sciences
University of Maryland Eastern Shore
aeydgahi@umes.edu

Dilshan Godaliyadda
Department of Electrical & Computer Engineering
University of Maryland
dilshan@umd.edu

Similola Falase
Department of Electrical Engineering
University of Oklahoma
simifalase@ou.edu

ABSTRACT

In this paper, a MATLAB-based simulation using a fuzzy logic system for autonomous navigation of unmanned surface vehicles is presented. The control system intelligently controls the vehicle to go from origin to destination avoiding the obstacles in its path. Depending on the anticipated danger of the obstacle, the vehicle chooses an angle of divergence which avoids the obstacles and keeps the speed of the vehicle. The simulation software is designed with a decentralized control structure for navigation system and does not require knowledge such as a global or local map about the navigation environment. The proposed control system of the vehicle has two main controllers. The first one is speed controller and the second one is rudder controller, which controls the direction of the vehicle. The inputs to navigation system consists of target position, obstacles, current location, current speed, and rudder's angular velocity. A table of comparison for navigation performance is presented to show effect of uncertainties and errors in locations of the vessel, the obstacles, and the destination.

INTRODUCTION

In the past recent years, different types of sensors such as Radar, Laser Range Finder, Forward Looking Infra Red, Day Light Camera, Sonar Sensors, GPS, Gyro compass, etc. have been utilized for autonomous navigation [1]-[4]. The navigation system for vessels

and ships is still manual for supervising, data acquisition, and processing. Manual methods are always subject to human errors.

When there is a need to guide the vehicle without human, an autonomous navigator should be installed on the vessel. Such navigators utilize the data logged from different sensors in Data Fusion Systems to estimate the locations of obstacles such as submarine, boats, or other objects that are either dynamic or static on the water or on the shore. Since information provided by the sensors is rapidly changing, the vessel should be guided toward the target instantly [5]-[8].

The non-classical methods for control include the methods that do not require the mathematical model of the system. These methods are based on the decision and thinking of the humans such as fuzzy systems and neural networks. Fuzzy systems provide a framework for incorporating the linguistic fuzzy information from the human experts but conventional controllers do not use this information. Fuzzy logic control systems are non-linear controllers. They are general and perform any non-linear control actions. If the designer carefully chooses the parameters of fuzzy logic controller, it is always possible to design a fuzzy logic controller that is suitable to control a non-linear system. The present interest in fuzzy control system is largely due to its success in the variety of applications. Some advantages of fuzzy control systems are as follows:

- They are model-free. They do not use the mathematical model of the system under control.
- They are easy to understand. Because they emulate human control strategy.
- They are easy to implement. Many VLSI chips have been developed which make the implementation of fuzzy logic controllers simple, easy, and fast.
- They are inexpensive to implement.

Fuzzy techniques are good engineering methods that are capable of making effective use of all available information. If the mathematical model of system does not exist or if it is too difficult to obtain the model, the most important information comes from sensors that provide numerical measurements and human experts that provide linguistic descriptions about the system and its control rules. In order to accomplish this, one must go through the process of fuzzification [9]-[10]. Fuzzification entails translating crisp knowledge of a certain process to a fuzzy knowledge. Fuzzy sets may be represented by a mathematical formulation often referred to as the membership function. This gives a degree or grade of membership within the set. In control system applications [11]-[16], membership values are actually measures of degree of causality in an input/output mapping.

This paper presents simulation software for autonomous navigation of an unmanned vessel using a fuzzy logic system [17] in MATLAB environment [18]. The objective of the control system is intelligently control the unmanned vessel to go from origin to destination avoiding different kinds of obstacles which may exists in the path. This simulation software attempts to produce different kinds of impediments that will be opted by the user of the system. According to user choice and depending on the anticipated danger of the obstacle, the vehicle will choose an angle of divergence which will avoid

the obstacle and will keep the vessel's speed. In order to obtain the required deviations and to control the angular speed (rudder) of the vessel an intelligent controller using fuzzy toolbox of the MATLAB is implemented instead of mathematical equations.

CONTROL SYSTEM

The proposed control system of the vessel has two main controllers. The first one is speed controller and the second one is rudder controller, which controls the direction of the vehicle. The block diagram of the control system is shown in figure 1. The main inputs to the navigation system are target position, obstacles, current location, current speed, and rudder's angular velocity. These five inputs are used to calculate the next coordinate of the position vector (a,b), the velocity (X,Y), and the rudder angle (α) that are required in order to reach a particular coordinate. The rudder angle 'alfa' (α), and the position vector (X_1, Y_1) provide the next position.

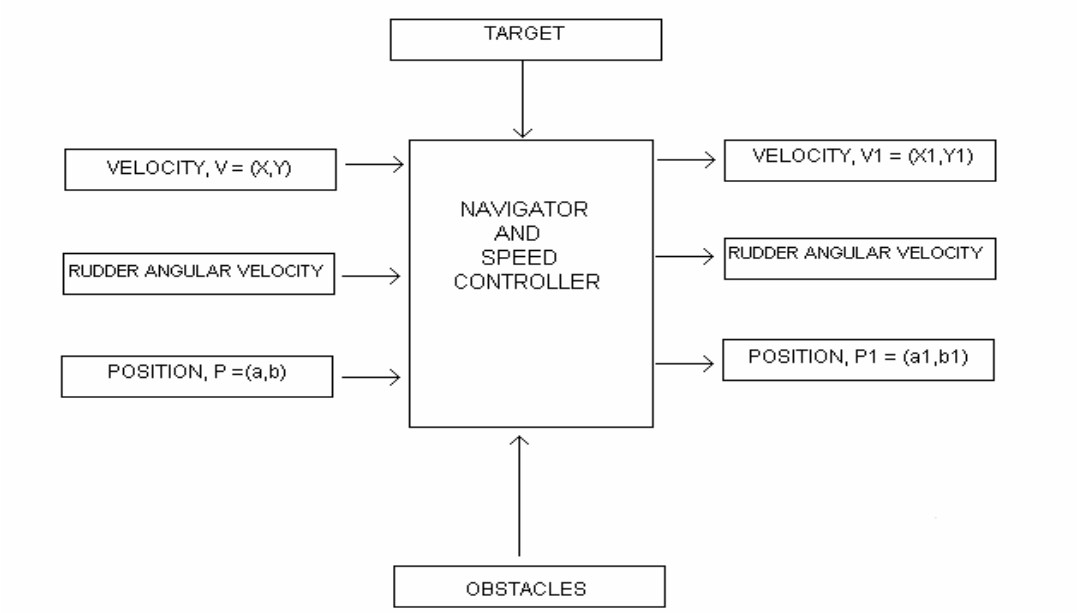

Figure 1: Structure of the Control System

The rudder angle α, is the key contributing factor which is calculated in radians according to the position vector, the speed, and type of obstacle encountered. This angle is the angle in the vector triangle made by the vectors between position and target for both X and Y coordinates as shown in figure 2. The X-Y plane can consequently be referenced to the geographic position of the vehicle.

Initially, the control system starts to navigate the vehicle at constant velocity and independent of the rudder in a straight line from 'origin' towards a 'target' with assumption of no obstacles in its linear path. However, if an obstacle is encountered in the course of travel, it then becomes an input, and the control system diverts the vessel

from its default path to travel in an alternate path to avoid the obstacle. Then, when the obstacle has successfully been avoided, the vessel will be returned to its original path. In the chain of events, the navigator is the central component which makes the ultimate decision of the required deviation. In doing so, it then computes the velocity vector (X1,Y1), position vector (a1,b1), and the rudder angular velocity (α1) for the next instant. In this software, the fuzzy toolbox of MATLAB is used to implement the fuzzy control system of the vehicle.

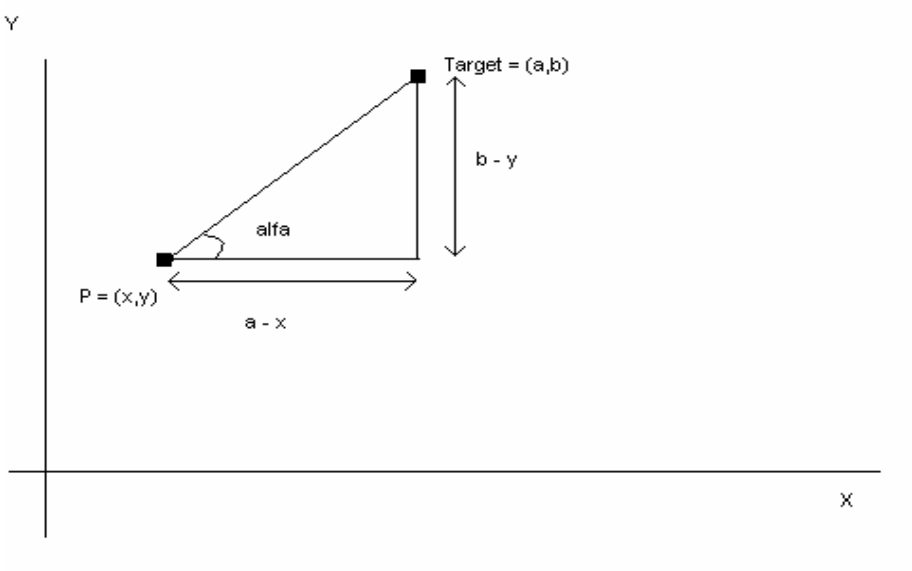

Figure 2: Coordinate System of Unmanned Vehicle

Fuzzy logic is a convenient way to map an input space to an output space. Between the input and the output, one can compose a controller using fuzzy logic that performs the work involved. The fuzzy membership function used in this project is 'evalmf'. The 'evalmf' function evaluates any membership function, when the variable range for evaluation of the membership function, the type of a membership function, and the appropriate parameters for that function are provided.

The fuzzy membership function used for the navigation controller of the proposed system is represented in figure 3, which characterizes and resembles a course of a ship's avoidance of an obstacle.

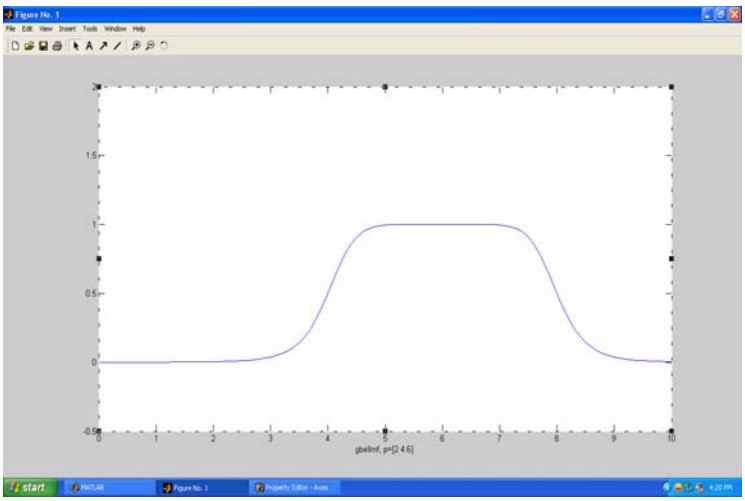

Figure 3: Fuzzy Membership Function

This particular membership function has been implemented mainly because of the fact that it has a sudden deviation from the path, although not at a right-angel, which in practice is impossible to create. Therefore, the deviation in path is quite logical in terms of an actual surface vessel.

SOFTWARE STRUCTURE

The program is implemented using MATLAB. The complete simulation software consists of a number of M-files. The most important M-files are:

1. *MFX 1* – This M-file controls the ship's diversion in path. It comprises of the fuzzy logic algorithm for the "evalmf" function. This is the control system that provides the logic necessary for the vessel to deviate from its path to the M-file Ship.m.
2. *ALFAX* – This M-file calculates the angle, α, in radians as well as the X and Y coordinates required to calculate α. This angle, α**,** is the angle in the vector triangle made by the vectors between position and target for both X and Y coordinates.

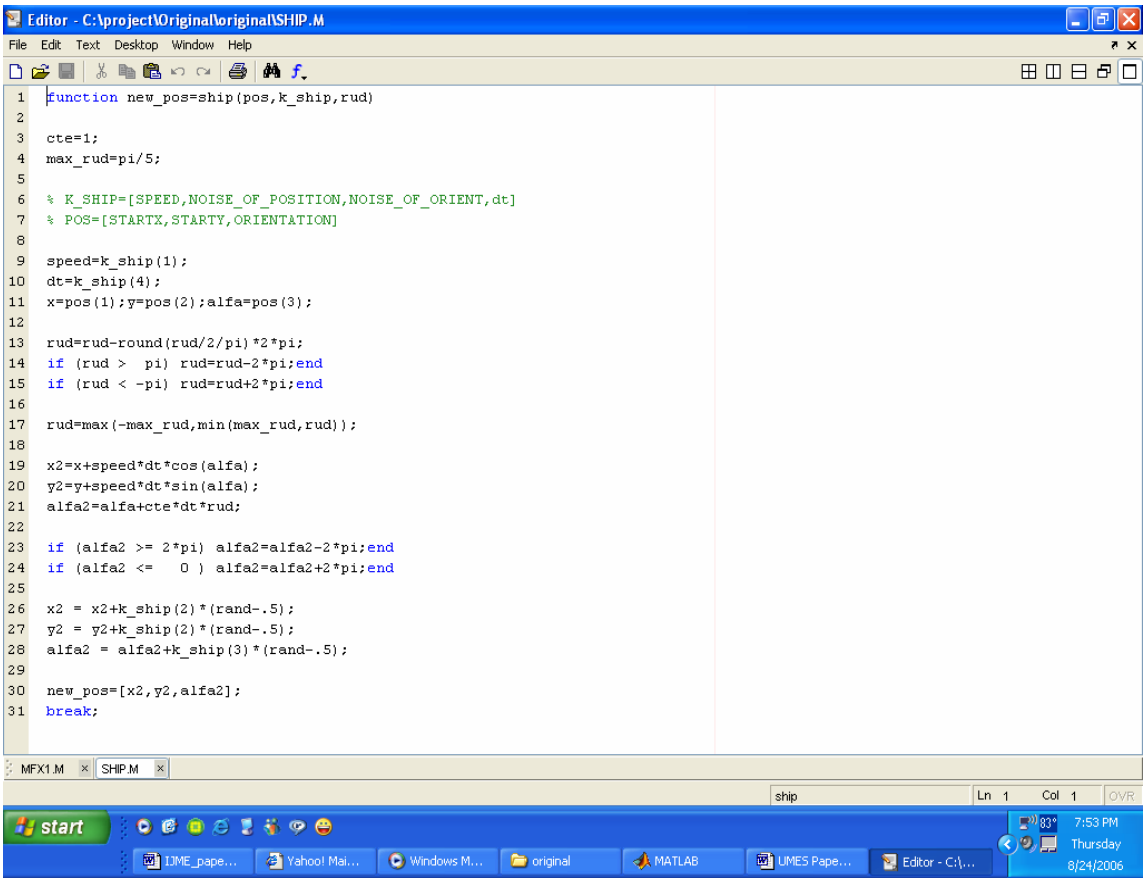

Figure 4: SHIP M-file

3. *SHIP* – Taking into consideration the logic provided by MFX1 M-file and the rudder angle calculated by ALFAX, this M-file determines the next position of the ship. This means that the rudder angle 'alfa' (α), and the position vector (X, Y) are utilized to get the next position (X1,Y1). The SHIP M-file is shown in figure 4.
4. *SELECT_PATH* – This is the M-file for the user to decipher between the two kinds of obstacles available.
5. *PATH2*, *PATH4* – These are the main M-files that contain the user interface and the plotting functions. These files also calculate the time elapsed within the journey from Origin to Target. The PATH2 M-file contains a different variety of formations of obstacles from which the user can select one. With PATH4 file, the user is able to choose the number of obstacles and their locations on the grid. The user interface from PATH4 file is shown in Figure 5.
6. *TARGETX* – This M-file computes the values for 'out,' and 'mf' required for the PATH4 M-file that helps to calculate the location of the next coordinate.

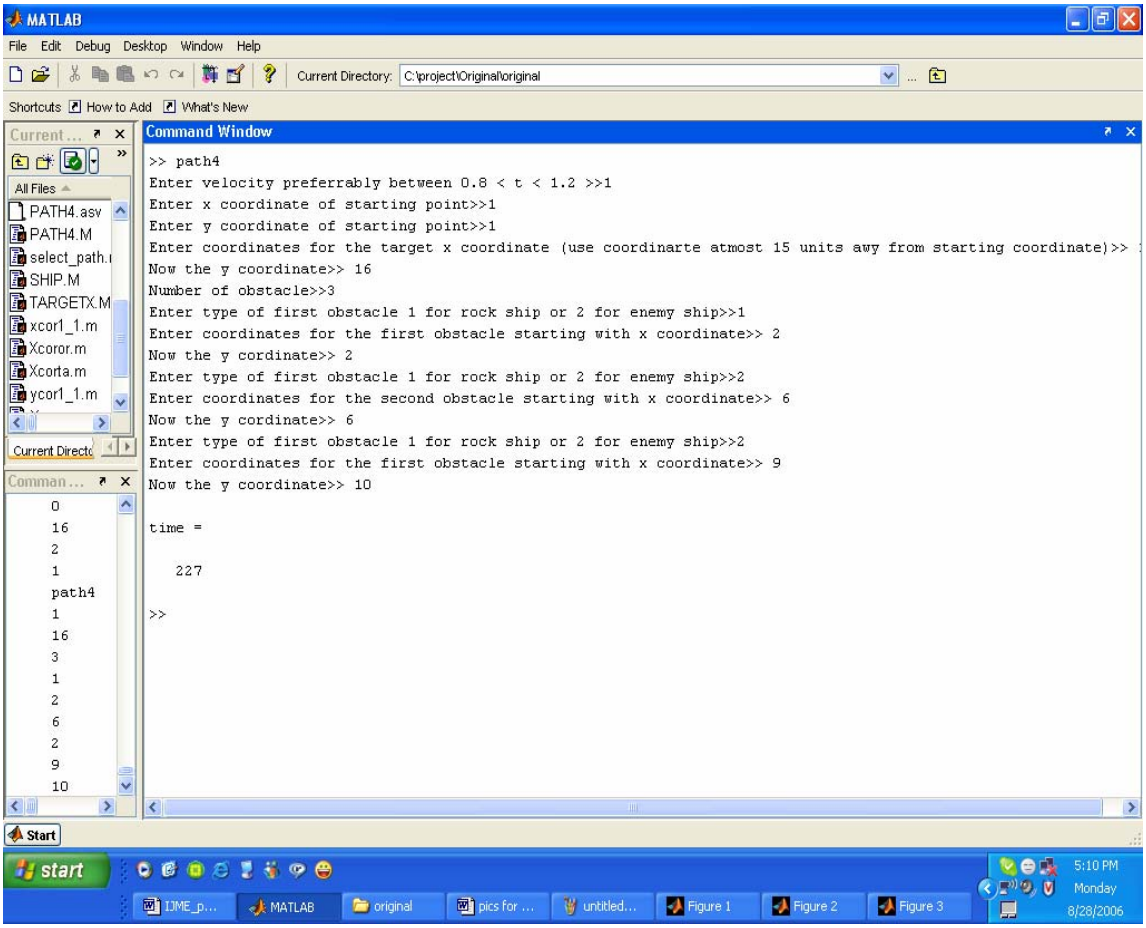

Figure 5: User Interface

7. *OBSX* – This M-file generates the non-lethal static obstacle such as a rock. Because of the non-lethal nature of it, diversion angle is set at Π/2 so that it merely avoids clashing.
8. *OBSX1*- This M-file generates a lethal static obstacle such as an enemy ship. The diversion angle is set to 3Π to grant the ship a greater divergence.

OPERATION

First the user has to enter "SELECT_PATH" on the command window. Then, out of the two available paths one should be selected. One of the paths provides the user with an option of selecting the number of individual obstacles positioned based on the user's input, and the other is where the user can choose a formation of obstacles which can be related to a configuration based on a predefined or expected occurrence of obstacles such as a cluster of icebergs or rocks. The options comprise of simple geometric arrangements in the predefined X-Y plane - a perfect square, a parallelogram shifted to the right or the left, a straight diagonal line, and a trapezoid with the parallel lines parallel to the X-axis or parallel to the Y-axis.

When the user selects the first path, the starting X and Y coordinates of the ship are requested through the interface. The position of the destination, in form of coordinates, is also requested. Next, the interface requests a total number of obstacles the user wishes to place, with a choice of one to three. The user is then able to pick the appropriate type of obstacles, either lethal or non-lethal. For each obstacle the user is also allowed to enter the position of each of the obstacles.

When the user selects the second path, the program as in the other path will ask for the user to input the starting and ending points as well as the desired velocity of the vehicle. However, now the user's choice is between the six different obstacle configurations as explained above.

OUTPUTS

For the first path there are three plots as the output. The first plot shows the displacement of the ship. This figure also displays the non-lethal obstacles in green color and lethal obstacles in red color. On the other hand, the second plot displays the Y coordinate with time in addition to the velocity-time graphs. The final plot shows the rudder activity versus time graph. The first plot is shown in figure 6. Figure 7 shows another output for this path although this time it is for two obstacles.

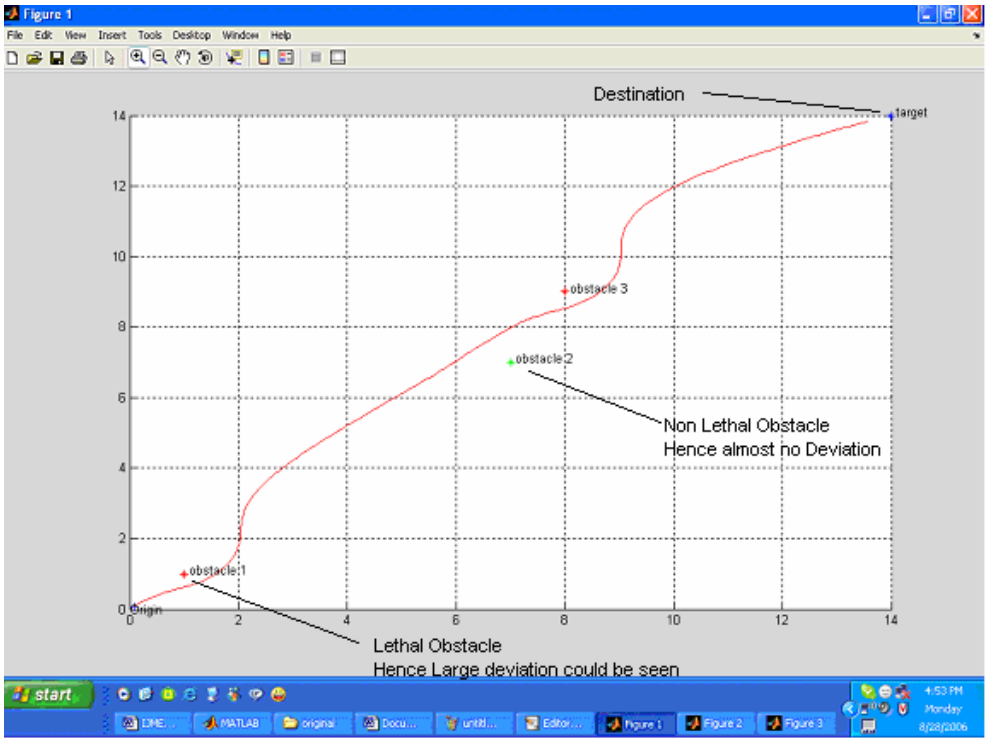

Figure 6: The Displacement of the Vehicle

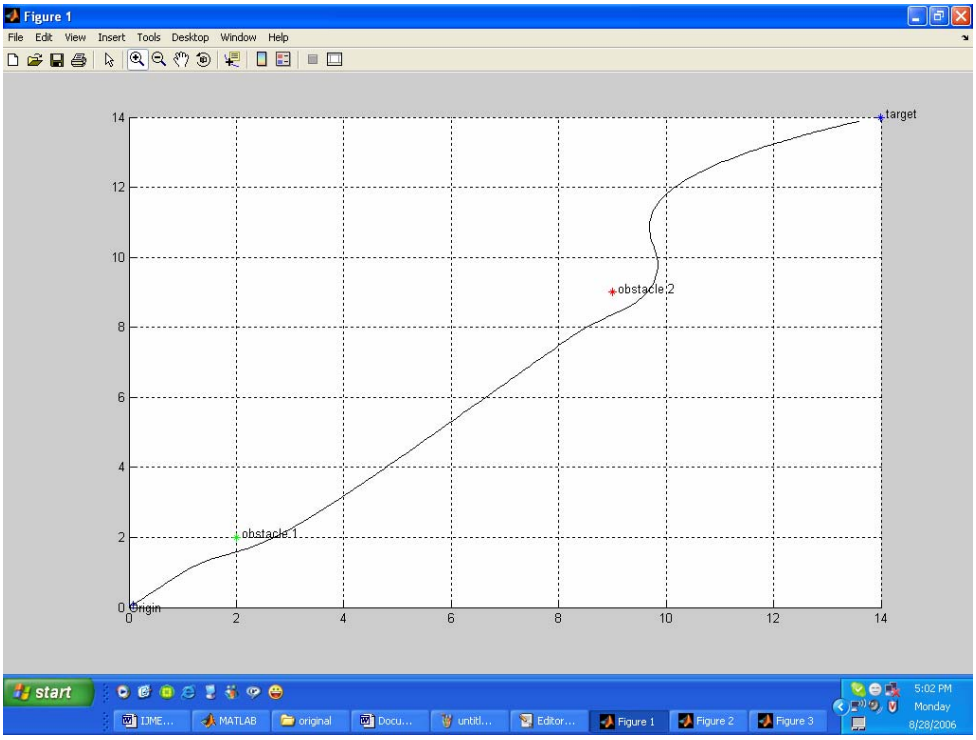

Figure 7: Output for Two Obstacles

For the output of the second path, there is one plot. This shows the obstacle formation as well as the path which is taken to reach the target. Figure 8 shows the output for one of the formations.

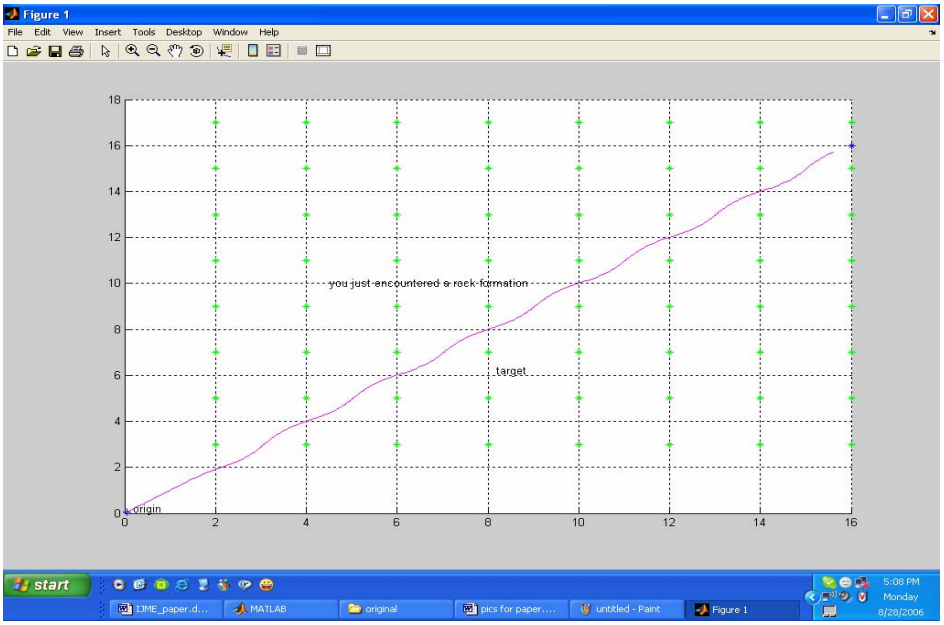

Figure 8: Output of One of the Obstacle Configurations

Figure 9 shows a case when a matrix of lethal obstacles is implemented. As shown, this resulted in vessel not reaching the target after all efforts it made to avoid the obstacles.

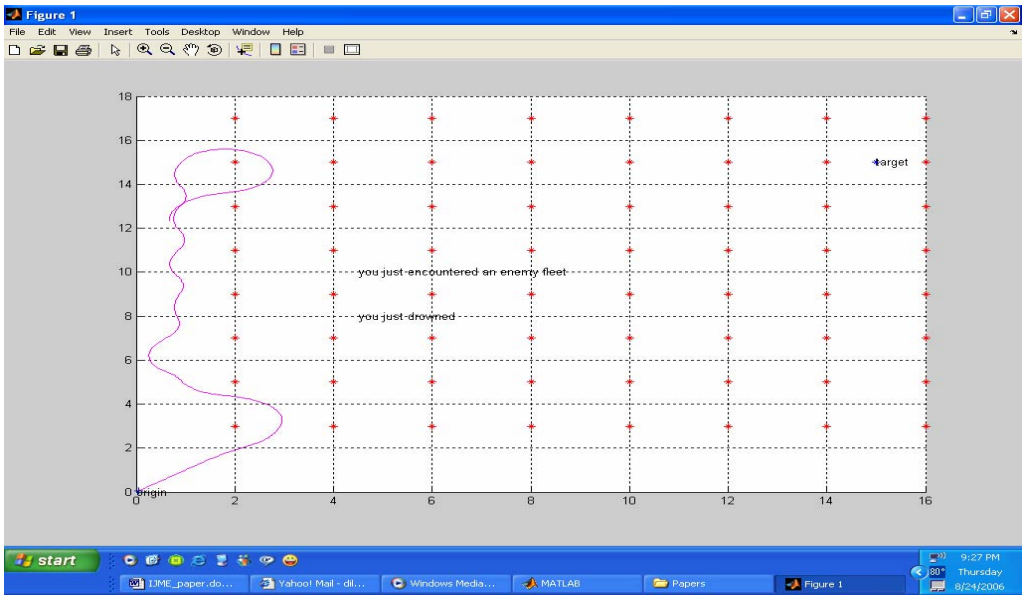

Figure 9: Output of a Lethal Obstacle Configuration

Another key output provided by this software is the time the vessel takes to go from the origin to its target position. To determine the effect of the obstacle variations towards the elapsed time, we considered different number of obstacles placed between the same origin and target at different places. Table 1 represents the data collected from this experiment.

Table 1: Time Elapsed for Different Formations of Obstacles

Number of obstacles	**Position of first obstacle**	**Position of second obstacle**	**Position of third obstacle**	**Time**
3	(4,4) non-lethal	(6,6) lethal	(8,8) lethal	206
2	(2,4) non-lethal	(5,5) non-lethal	none	208
3	(4,4) lethal	(6,6) non-lethal	(9,9) non-lethal	213
Square formation	Non-lethal placed as a parallelogram in intervals of 1 both in the X and Y directions			350
Square formation	Lethal placed as a parallelogram in intervals of 1 both in the X and Y directions			401 – before drowning

CONCLUSION

The development of a MATLAB-based simulation for navigation of an unmanned surface vehicle has been presented. The software is able to take into consideration the occurrence of obstacles along the path of travel. The successful generation and simulation of two different kinds of static obstacles, one a lethal obstacle and the other, a non-lethal obstacle have been presented.

The software has been developed in such a way that it can be easily incorporated into other related applications for unmanned navigation designs.

ACHNOWLEDGMENT

This work was supported by a grant from NSF Advanced Curriculum and Technology-Based Instructional Opportunities Network program through University of Maryland Eastern Shore.

REFERENCES

[1] Pettersen, K.Y. and T.I. Fossen, "Underactuated Dynamic Positioning of Ship-Experimental Results," IEEE Transactions on Control Systems Technology, pp. 856-863, September 2000.

[2] Petterson, K.Y. and O. Egeland, "Exponential Stabilization of an Underactuated Surface Vessel," Proc. 35th IEEE Conf. on Decision and Control, pp. 967-970, December 1996.

[3] Leonard, N.E., "Periodic Forcing, Dynamics and Control of Underactuated Spacecraft and Underwater Vehicles," Proc. 34th IEEE Conference on Decision and Control, pp. 3980- 3985, December 1995.

[4] Hussein, A S., "Numerical Simulation of Neural Network-Controlled Unmanned Undersea Vehicle." WSEAS Transactions on Circuits and Systems. Vol. 2, no. 3, pp. 608-615. July 2003.

[5] Meystel, A., "Planning in a Hierarchical Nested Autonomous Control System," SPIE Conference on Mobile Robots, Eds. W. J. Wolfe and N. Marquina, Cambridge, MA, pp. 42-76, 1987.

[6] Enge, P., T. Walter, S. Pullen, C. Kee, C., Y.C. Chao, and Y.J. Tsai, "Wide Area Augmentation of the Global Positioning System," Proceedings of the IEEE, Volume 84, No. 8, August 1996.

[7] Moshiri, B., A. Eydgahi, M. Najafi, and R.H. Nezhad, "Multi-Sensor Data Fusion Used in Intelligent Autonomous Navigation," Proceedings of IASTED International Conference on Control and Applications, Banff, Canada, pp. 515-520, July 1999.

[8] Alparone, L., F. Argenti, G. Benelli, V. Cappellini, and A. Mecocci, "A Fuzzy Complete SAR Processing Chain for Ship Detection and Velocity Estimation," European Trans. Telecommunications Relations Technology, Vol. 2, No. 6, pp. 689-694, 1991.

[9] Zadeh, L.A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," IEEE Trans. on Systems, Man, and Cybernetics, pp. 28-44, 1973.

[10] Mamdani, E.H., "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," IEEE Transactions on Computers, pp.1182-1191,1977.

[11] Eydgahi, A.M. and M. Fotouhi, "A Fuzzy Knowledge-Based Controller to Tune PID Parameters," Computers in Education Journal, Vol. 10, No 2, pp. 2-7, April/June, 2000.

[12] Beigy, H. and A.M. Eydgahi, "An Adaptive Fuzzy Model Reference Control for Robot Manipulators," Proceedings of International Conference on Automation, Indore, India, pp. 59-62, December 1995.

[13] Beigy, H., A.M. Eydgahi, and S.D. Katebi, "Elastic Fuzzy Logic for Self-Learning Control Systems," Proceedings of 1995 IEEE International Conference on Neural Networks, Perth, Western Australia, pp. 622-626, November/December 1995.

[14] Jamshidi, M., R. Marchbanks, K. Bisset, R. Kelsey, S. Baugh, and D. Barak, Computer-Aided Design of Fuzzy Control Systems: Software and Hardware Implementations, Elsevier Science publishers, pp. 81-89, 1992.

[15] Zhang, Y.G., J. Xiang, and J. Xiao, J. "Design of Decentralized PDC Controllers for Discrete-Time Fuzzy Large-Scale Systems: An LMI Method," Control and Decision, Vol. 19, No. 3, pp. 351-354, 2004.

[16] Tong, S.C. and T.Y. Chai, "Direct Adaptive Fuzzy Output Feedback Control for Uncertain Nonlinear Systems," Control and Decision, Vol. 19, No. 3, pp. 257-261, 2004.

[17] McNeill, F., Fuzzy Logic: A Practical Approach, Academic Press, 1994.

[18] Mathworks Inc., Fuzzy Logic Toolbox, World-Wide Web URL http://www.mathworks.com

BIOGRAPHIES

ALI EYDGAHI is a Professor and Chair of Engineering and Aviation Sciences Department, Director of the Center for 3-D Visualization and Virtual Reality Applications, and Technical Director of Space Vehicle Mission Planning Laboratory at the University of Maryland Eastern Shore. Dr. Eydgahi has received a number of awards including the Dow Outstanding Young Faculty Award from American Society for Engineering Education in 1990 and the Silver Medal for outstanding contribution from International Conference on Automation in 1995. Dr. Eydgahi has served as a session chair and a member of scientific and international committees for many international conferences and has published more than hundred papers in refereed international and national journals and conference proceedings.

DILSHAN GODALIYADDA is a senior student in Electrical Engineering program at the University of Maryland. Dilshan is currently a member of Etta Kappa Nu, the ECE Honor Society. He is a transfer student from the University of Maryland Eastern Shore where he was president of the IEEE student chapter. He has worked as an undergraduate research assistant and tutor. His research interest is in the area of data visualization and simulation.

SIMILOLA FALASE is currently a senior student in the department of Electrical Engineering at Oklahoma University. She was an undergraduate student in Electrical Engineering program and an officer of IEEE student branch at the University of Maryland Eastern Shore for two years. Similola research interest is in design and development of intelligent systems for control industry.