Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Obstacle Avoidance for Unmanned Sea Surface Vehicles: A Hierarchical Approach [1]

P. Krishnamurthy * **F. Khorrami** *,** **T. L. Ng** *

* *IntelliTech Microsystems, Inc., 4931 Telsa Drive, Suite B, Bowie, MD 20715. (e-mails: pkrishnamurthy@imicro.biz, khorrami@smart.poly.edu, tlng@imicro.biz)*
** *Control/Robotics Research Laboratory (CRRL), Dept. of Electrical and Computer Engg., Polytechnic University, Brooklyn, NY, 11201.*

Abstract: In this paper, we describe a hierarchical system for path planning and obstacle avoidance for totally autonomous Unmanned Sea Surface Vehicles (USSVs). The proposed system is comprised of three major components: a wide-area planner based on the A^* graph-search algorithm, a local-area planner based on our low-resource path-planning and obstacle avoidance algorithm GODZILA, and an inner-loop nonlinear tracking control law. The performance of the proposed system is demonstrated through simulations using our high-accuracy real-time Six Degree-of-Freedom (DOF) Hardware-In-The-Loop (HITL) simulation platform whose design and implementation have been documented in our recent papers. The HITL platform is capable of simultaneously simulating multiple USSVs and passive obstacles and incorporates a nonlinear dynamic model of the USSV including detailed characterizations of hydrodynamic effects, emulation of sensors and instrumentation onboard the USSV, and the actual hardware and software components used for USSV control in the experimental testbed. The performance of the inner-loop controller has been validated through experimental tests which are described briefly in this paper and the experimental validation of the complete obstacle avoidance system is currently underway.

1. INTRODUCTION

In this paper, we describe the design and implementation of a path planning and obstacle avoidance system for Unmanned Sea Surface Vehicles (USSVs). The various civilian and military applications of USSVs are well-recognized and their control and navigation problems have been studied in the literature (Fossen [1994], Goclowski and Gelb [1966], Katebi et al. [1997], Fossen and Grovlen [1998], Loria et al. [2000], Fang et al. [2001], Mazenc et al. [2002], Do et al. [2002], Lefeber et al. [2003]). The proposed obstacle avoidance system (OAS) is of a hierarchical structure and is comprised of three major components:

(1) A Wide-Area Planner (WAP) based on the A^* graph-search algorithm
(2) A Local-Area Planner (LAP) based on our low-resource reactive obstacle avoidance algorithm called GODZILA (Game-Theoretic Optimal Deformable Zone with Inertia and Local Approach) (Krishnamurthy and Khorrami [2005])
(3) An inner-loop robust nonlinear dynamic controller which computes actuator commands to track the reference trajectory generated by the WAP and LAP. Experimental results on the inner-loop controller component were presented in (Krishnamurthy et al. [2007]).

The performance of the proposed OAS is demonstrated through Hardware-In-The-Loop (HITL) simulation studies on our high-accuracy real-time Six Degree-of-Freedom (DOF) Hardware-In-The-Loop (HITL) simulation platform (Krishnamurthy et al. [2007]). The HITL simulation platform utilizes the detailed 6-DOF USSV dynamic model developed in (Krishnamurthy et al. [2005]) and is capable of simulating multiple USSVs and passive obstacles simultaneously in real-time. Additionally, the HITL simulation platform incorporates emulation of the instrumentation onboard the USSV including the sensors and actuators and the interface to these hardware components through a Controller Area Network (CAN) bus. The HITL simulation platform provides the computer which runs the controls software with the exact environment which it sees when operating in the experimental USSV testbed.

The paper is organized as follows. The architecture of the proposed hierarchical OAS is described in Section 2. The three major components of the proposed OAS, namely, the WAP based on A^*, the LAP based on GODZILA, and the inner-loop controller, are described in Sections 2.1, 2.2, and 2.3. The experimental setup and the HITL simulation platform are described in Section 3 and HITL simulation studies of the proposed obstacle avoidance system are discussed in Section 4.

2. ARCHITECTURE OF THE PROPOSED OBSTACLE AVOIDANCE SYSTEM FOR USSVS

The overall architecture of the proposed hierarchical OAS is illustrated in Figure 1. Given a desired vehicle trajectory (specified as a sequence of desired locations by a manual operator or generated from a high-level mission planner), the objective of the designed OAS is to track the desired vehicle trajectory as closely as possible while avoiding obstacles. This problem objective is attained through a hierarchical design based on the three major components shown in Figure 1. The WAP and the LAP together address the path planning and obstacle avoidance tasks while the task of the inner-loop controller is to generate appropriate actuator commands to track the motion direction and velocity commands computed by the WAP/LAP combination. The block labeled "Inertial Navigation Algorithms" in Figure 1 is comprised of quaternion-

[1] This work was supported in part by the Office of Naval Research (ONR) under contract Nos. N00014-04-M-0181 and N00014-06-C-0051.

based integration and Kalman filtering algorithms and is responsible for generating linear and angular position and velocity estimates based on the readings from the GPS and the Inertial Measurement Unit (IMU). These linear and angular position and velocity estimates are used by all three major components of the hierarchical OAS.

The WAP and the LAP address the far-field (or global) and the near-field (or local) aspects of path planning and obstacle avoidance. The WAP is designed based on the A^* graph-search algorithm (see Section 2.1) and operates based on Radar data to generate a recommendation as to the trajectory to be followed. For reasonable computational requirements, the environment representation used by the WAP is in terms of a coarse grid (with granularity of, for instance, 15 m), but extends out to a reasonably large range (of around 10 km or higher depending on mission specifications). In contrast, the task of the LAP is the avoidance of local obstacles which could be moving or too small to show up in the coarse map used by the WAP. The LAP is required to operate at a much higher sampling rate than the WAP and is therefore based on our low-resource path planning and obstacle avoidance algorithm GODZILA (see Section 2.2) which is specifically designed to achieve low computational complexity to allow deployment on small autonomous vehicles. Based on the local obstacle topology, the LAP computes the required local perturbations on the global trajectory recommendation provided by the WAP and outputs USSV motion direction and velocity commands to the inner-loop controller which then computes the actual control signals to be output to the physical actuators (rudder and propeller).

2.1 Wide-Area Planner

The wide-area planner addresses the far-field (or global) path planning and obstacle avoidance problem utilizing an environment map with large range but low resolution. As such, it takes into account larger obstacles such as other ships/USSVs (which could be stationary or moving), but not smaller obstacles such as buoys or debris which are addressed by the LAP. The WAP is designed based on the A^* graph-search algorithm (Hart et al. [1968], Dechter and Pearl [1985]) which computes the optimal trajectory from a given initial position to a given final position taking into account obstacles which are represented through an occupancy grid. The A^* algorithm utilizes an iterative search based on a cost function of the form $f = g + h$ with two components: g which reflects the cost of the path so far and h which is a heuristic estimate of the remaining cost to reach the desired final position. The presence of the heuristic cost in the A^* algorithm makes it straightforward to incorporate a penalty for close approach to obstacles and enables finding of a trajectory solution with some desired clearance to obstacles.

In conjunction with the A^* algorithm, the WAP also includes an implementation of the rules of the road (i.e., NAV Rules), a set of rules prescribed in COLREGS (Collision Avoidance Regulations) which specify appropriate actions in response to approach of other marine vehicles. For instance, if a vessel is approaching head-on, NAV Rules state that the USSV should alter her course to starboard to ensure that each vessel passes on the other's port side. The implementation of NAV Rules is also based on the Radar data which are passed through a Xenex processor which is capable of identifying obstacles and tracking them temporally.

2.2 GODZILA

In this section, we briefly describe the GODZILA (Game-Theoretic Optimal Deformable Zone with Inertia and Local Approach) algorithm (Krishnamurthy and Khorrami [2005]) for path planning for mobile robots and unmanned vehicles. GODZILA is applicable to both two-dimensional (2D) and three-dimensional (3D) scenarios and provides a solution to the navigation problem in completely unknown environments without requiring the building of an obstacle map. The algorithm follows a purely local approach using only the sensor measurements at the current time and requiring only a small number of stored variables in memory. This minimizes the memory and computational requirements for implementation of the algorithm, a feature that is especially attractive for small autonomous vehicles.

The trajectory is generated through online minimization of an optimization cost at each sampling instant. It is shown that the optimization cost can be chosen so that the minimizer can be obtained in closed form. The optimization cost has three terms which penalize, respectively, motion in directions other than the direction to the target, motion towards obstacles, and back-tracking. In addition to the optimization algorithm, GODZILA includes two components, a local straight-line planner utilized if the target is visible and navigation towards a random target. Since the algorithm follows a local approach, it is possible to be caught in a local minimum. When a local minimum or a "trap" is detected, navigation towards a random target is initiated to escape the trap. It is proved in (Krishnamurthy and Khorrami [2005]) that GODZILA guarantees convergence to the target in finite time with probability 1 in any finite-dimensional space. GODZILA is designed to be particularly efficient as a lightweight path planning and obstacle avoidance algorithm suitable for small autonomous vehicles in "urban"-like or "office"-like environments (i.e., featuring multiple obstacles, but with a simple topology, unlike "maze"-like environments) and is capable of operating with a small number of range sensors.

2.3 Inner-Loop Controller

The inner-loop controller utilizes the backstepping-based design presented in (Krishnamurthy et al. [2005]). Given a planar reference trajectory $(x_{\text{ref}}, y_{\text{ref}})$, the control law (see (Krishnamurthy et al. [2005]) for details) is given by the form

$$z_1 = \dot{x}_{\text{ref}} - k_{x,p}(x - x_{\text{ref}}) - k_{x,i}\int_0^t (x - x_{\text{ref}})d\tau \tag{1}$$

$$z_2 = \dot{y}_{\text{ref}} - k_{x,p}(y - y_{\text{ref}}) - k_{x,i}\int_0^t (y - y_{\text{ref}})d\tau \tag{2}$$

$$\theta_{z,\text{ref}} = \text{atan2}(z_1, z_2) \tag{3}$$

$$v_{t,x,\text{ref}} = \frac{z_1}{\cos(\theta_{z,\text{ref}})} = \frac{z_2}{\sin(\theta_{z,\text{ref}})} \tag{4}$$

$$\begin{aligned} F_T = &-M_{H,22}v_{t,y}v_{r,z} + d_{x,1}v_{t,x,\text{ref}} + d_{x,2}|v_{t,x,\text{ref}}|v_{t,x,\text{ref}} \\ &-k_{x,d}(v_{t,x} - v_{t,x,\text{ref}}) + M_{H,11}\dot{v}_{t,x,\text{ref}} \end{aligned} \tag{5}$$

$$\begin{aligned} \tau_R = &-(M_{H,11} - M_{H,22})v_{t,x}v_{t,y} + d_{\theta_z,1}v_{r,z,\text{ref}} \\ &+d_{\theta_z,2}|v_{r,z,\text{ref}}|v_{r,z,\text{ref}} - k_{\theta_z,p}(\theta_z - \theta_{z,\text{ref}}) \\ &-k_{\theta_z,d}(\dot{\theta}_z - \dot{\theta}_{z,\text{ref}}) + M_{H,66}\ddot{\theta}_{z,\text{ref}} \end{aligned} \tag{6}$$

where F_T is the forward thrust, τ_R is the rudder torque, and $k_{x,p}$, $k_{x,i}$, $k_{x,d}$, $k_{\theta_z,p}$, and $k_{\theta_z,d}$ are control gain parameters. It is to be noted that these parameters would be gain-scheduled in practice to achieve good dynamic performance over a wide range of operating conditions. Also, the control law (6) could be augmented with an

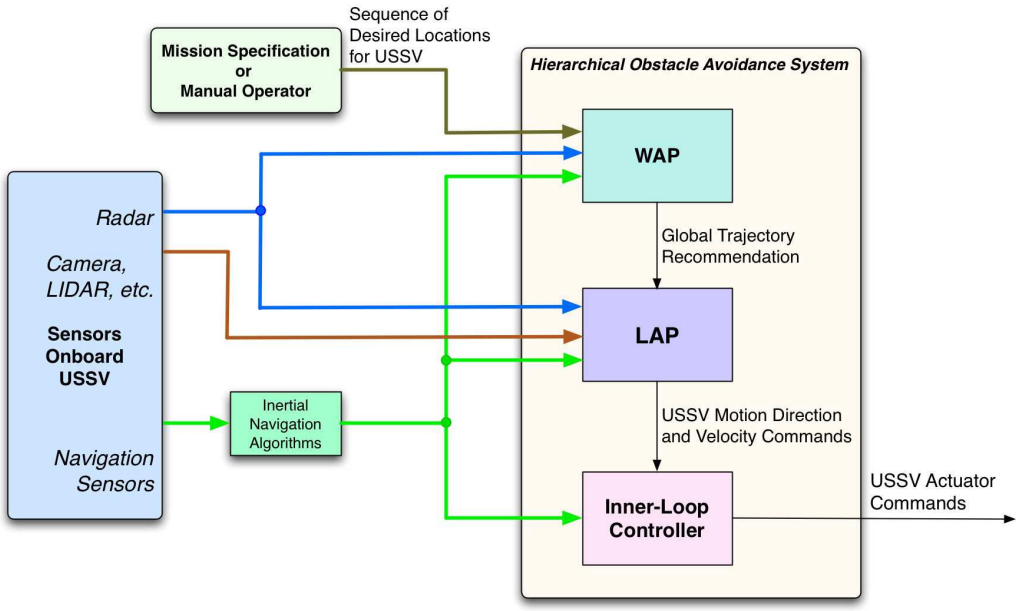

Fig. 1. Overall architecture of the proposed OAS.

inner-loop rudder feedback controller to compensate for rudder dead zone and delay and to improve the rudder transient performance. A controller based on (6) has been tested on the USSVs described in Section 3.1 and it has been observed that the performance of the controller is definitely adequate for use in our OAS as the inner-loop controller component which receives motion direction and velocity commands from GODZILA and then closes the loop on the actuators onboard the USSV to achieve the motion commanded by GODZILA. The performance of the controller for a few sample experimental tests of trajectory tracking are illustrated in Figures 2-3. In Figure 2, the controller was commanded to visit two waypoints (the two green circles in Figure 2) and then, after visiting the second waypoint, to track a straight line of heading 135 degrees (the red +'s in Figure 2). In Figure 3, two more experimental runs for waypoint tracking are shown wherein the USSV is commanded to visit a sequence of waypoints.

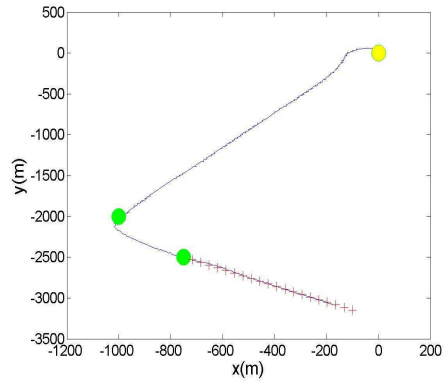

Fig. 2. Experimental Results: Position in a coordinate frame locally tangential to Earth's surface (with y axis pointing due north). Yellow circle: initial location. Green *'s: specified waypoints. Red +'s: specified straight line trajectory to be tracked after visiting the second waypoint.

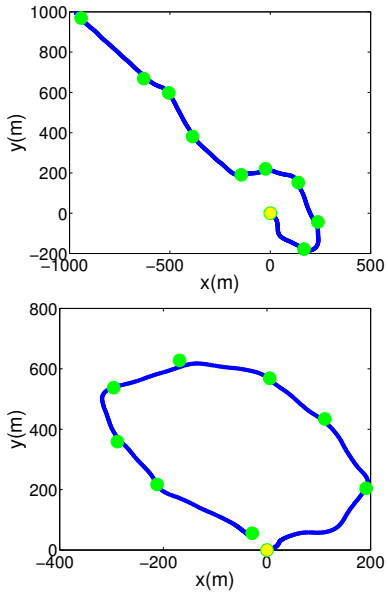

Fig. 3. Experimental tests for waypoint tracking: Position in a coordinate frame locally tangential to Earth's surface (with y axis pointing due north). Yellow circle: starting position; Green circles: specified waypoints.

3. EXPERIMENTAL SETUP AND HITL SIMULATION PLATFORM

The HITL platform is designed to exactly mimic the operating conditions seen by the controls hardware and software in operation onboard the USSV. The hardware architecture of the experimental USSV testbed (also see Section 3.1) is illustrated in Figure 4. The control algorithms are implemented on a notebook computer (with a Pentium M 1.6GHz processor and running Windows XP). The sensors and actuators on the USSV are all connected to a common high-speed CAN bus. A Kvaser USB-to-CAN adapter is utilized to access the CAN bus from the notebook computer. The currently available sensors on the USSV include a compass, a GPS, and water speed and

depth sensors and the available actuator inputs include rudder angle and port and starboard throttles. In addition, our avionics box (which is the same as used in our work on helicopter control (Ng et al. [2005])) is interfaced with the notebook computer via a serial port. This avionics box provides a six degree-of-freedom IMU with an update rate of 50 Hz.

The developed HITL simulation platform is illustrated in Figures 5 and 6. The HITL simulation testbed includes a real-time USSV dynamics simulation software (running on Computer 1) and the notebook computer (Computer 2) which is used to run control algorithms onboard the USSV. The USSV dynamics simulation software on Computer 1 is based on the detailed six degree-of-freedom USSV dynamic model developed in (Krishnamurthy et al. [2005]) which incorporates detailed characterizations of the various external forces and torques (including hydrodynamic effects, environmental disturbances, control surfaces, propellers, etc.) and is designed to be flexible with all USSV dynamics parameters specified at run-time through text-based configuration files which can be manipulated either directly or through a GUI front-end. Computer 2 receives serial IMU data (which emulates data from the IMU in our avionics hardware) from Computer 1 with an update rate of 50 Hz. Computer 2 also interacts with Computer 1 through USB-to-CAN adapters. The software on Computer 1 includes a complete emulation of the CAN interface which will be seen by Computer 2 during operation on the USSV including all sensor messages and actuator status messages with the proper formats and update rates. The software on Computer 1 receives actuator commands through the CAN interface and computes a full six degree-of-freedom dynamic simulation of the ship. A PIC is used to provide an accurate timing source for real-time fidelity of the HITL simulation. The result of the dynamic simulation is visualized using an OpenGL GUI front-end (screenshot in Figure 7) which can be displayed on Computer 1 or can be exported to another computer via a network socket interface.

The control software running on Computer 2 is structured as a multi-threaded application with separate threads for the following tasks:

- Acccessing the sensors and actuators by reading messages from and writing messages to the CAN bus in the appropriate CAN message formats. The CAN messages from the various sensors and actuators are not synchronized and have different update rates.
- Serial communication with the avionics box to read IMU data.
- Number crunching required for navigation and control algorithm computations.
- Communicating via a network socket with a front-end program which provides a graphical user interface.

These threads communicate using a client-server architecture with double buffering.

The HITL simulation testbed has been designed to be able to support multiple USSVs simultaneously (limited only by the processing, graphics, and I/O port capabilities of the computers being utilized). Furthermore, the testbed has been designed to be flexible so that depending on available hardware, a subset of the USSVs could be simulated in HITL mode while the dynamics of the rest of the USSVs could be simulated purely in software. This feature of the HITL simulation testbed is of great utility (see Figure 11) in the testing of the proposed OAS for USSV applications.

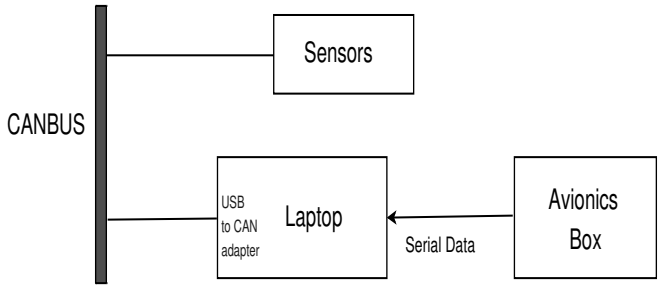

Fig. 4. Architecture of Experimental Setup.

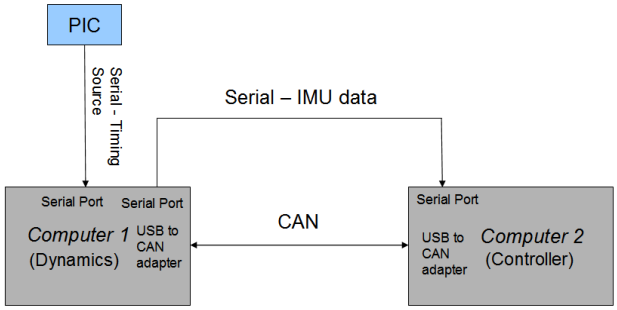

Fig. 5. Architecture of USSV HITL Simulation Testbed.

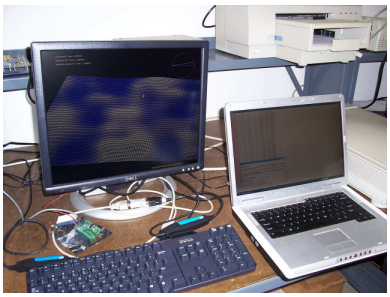

Fig. 6. USSV HITL Simulation Testbed.

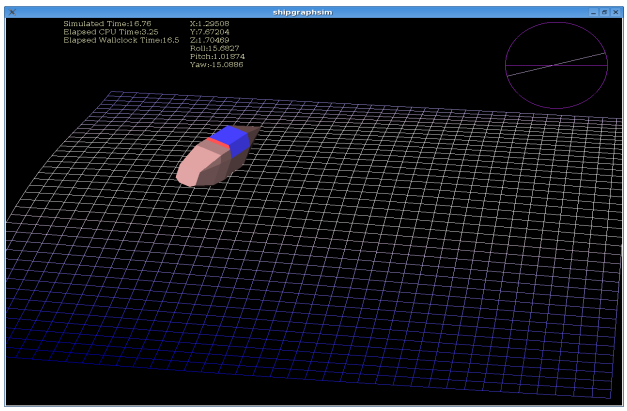

Fig. 7. Screenshot of USSV simulation package.

3.1 HITL Simulator Validation Using Experimental Data

The fidelity of the USSV dynamic model and the HITL platform were tested using experimental data collected from two different USSVs (shown in Figure 8) in the Atlantic ocean. The parameters of the USSVs (which are approximately of length 12 m and mass 9000 kg) were identified via least squares identification based methods using

the USSV dynamic model developed in Krishnamurthy et al. [2005] and extensive experimental data collected with various excitation signals. Some comparisons between experimentally observed USSV response and simulations using the identified USSV parameters for the PowerVent boat shown in Figure 8:Top are shown in Tables 1 and 2 and Figure 9. Due to confidentiality requirements, further details about ship parameters and experimental observations cannot be included in this paper. Based on the data in Tables 1 and 2 and additional comparisons that have been performed with a wide variety of excitation signals, it is seen that the developed HITL simulation platform accurately captures the dynamic response of the USSVs and the sensor and actuator behaviors.

Fig. 8. USSVs used in experimental tests: Top: U. S. Navy PowerVent APTD (Advanced Propulsion Technology Demonstrator); Bottom: U. S. Navy USSV-HTF (High Tow Force).

Table 1. Turn circle radius r_t as a function of percent throttle and rudder angle†

% Throttle	Rudder Angle	r_t (Experimental)	r_t (Simulation)
20	11.5	1.000	1.003
40	11.5	1.412	1.397
60	11.5	2.546	2.594
20	17.5	0.910	0.890
40	17.5	1.322	1.281
60	17.5	0.699	0.710
20	30	0.725	0.752
40	30	0.334	0.412
60	30	0.890	0.857

†Normalized so that experimental reading of turn circle radius at 20% throttle and 11.5^o rudder angle is 1 unit.

Table 2. Steady-state velocity v_{st} as a function of percent throttle at zero rudder angle‡.

% Throttle	v_{st} (Experimental)	v_{st} (Simulation)
10	1.000	0.955
30	1.409	1.500
40	1.864	1.909
50	2.182	2.227
60	3.409	3.500
80	8.909	8.727

‡Normalized so that experimental reading of steady-state velocity at 10% throttle is 1 unit.

4. SIMULATION STUDIES OF THE PROPOSED OBSTACLE AVOIDANCE SYSTEM

The performance of the proposed OAS for USSVs has been verified through extensive simulations (see screenshots in Figures 10, 11, and 12) utilizing the multi-USSV dynamic simulator and HITL simulator which support simultaneous real-time simulation of multiple USSVs with full six-degree-of-freedom dynamic simulation for each USSV. In addition, the simulator can accommodate moving or stationary passive obstacles in the environment. The simulator includes range computation and proximity sensor models which mimic data obtained from a radar in experimental deployment. Each USSV navigates using the path-planning and obstacle avoidance system. Optionally, obstacles moving along pre-specified or randomly generated trajectories can also be incorporated in the dynamic simulation to model the possible presence of debris or malfunctioning USSVs. In Figure 10, the USSV starts from an initial position (marked by a yellow circle) near the bottom left and is required to reach a target location (marked by a green circle) which is near the top right of the figures while avoiding six obstacles. Of the six obstacles, four are stationary (the cuboids in Figure 10) and two are moving (the ellipsoids in Figure 10). Four screenshots as the USSV moves from the initial position to the target position while avoiding the stationary and moving obstacles are shown in Figures 10(a)-(d). It is seen that the USSV has reached the target position in Figure 10(d). In Figure 11, three USSVs are deployed in the same waterspace with each having its own initial and target locations. Each USSV needs to avoid both the other USSVs and also the stationary and moving obstacles while finding a path to the target location.

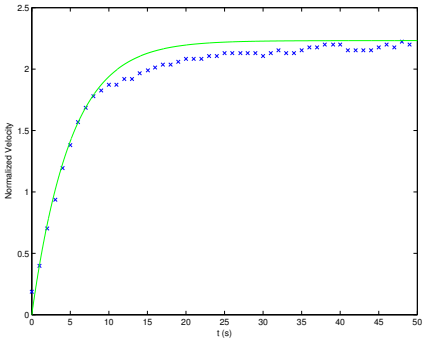

Fig. 9. Acceleration profile at 50% throttle (normalized so that experimental reading of steady-state velocity at 10% throttle is 1 unit): Experimental - blue 'x' line; Simulation - green solid line.

5. CONCLUSION

We have described the development of a hierarchical OAS for USSVs and its testing using a high-accuracy real-time multi-USSV HITL dynamic simulation platform. The proposed OAS comprises of a wide-area planner based on the A^* graph-search algorithm, a local-area planner based on GODZILA and a robust nonlinear inner-loop controller. Experimental testing and validation of the proposed OAS on the USSVs described in Section 3.1 is currently ongoing.

ACKNOWLEDGMENT

The authors would like to thank Dr. Robert Brizzolara of ONR for support of the effort and technical review of the manuscript. We would also like to thank Mr. Eric Hansen for his support and for the pictures of the PowerVent ship. We would also like to record our appreciation for Jason Altice of WRSystems and Al Frontin and Sam Calabrese of NSWCCD for their on-ship support.

REFERENCES

R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A^*. *Journal of the ACM*, 32(3):505–536, July 1985.

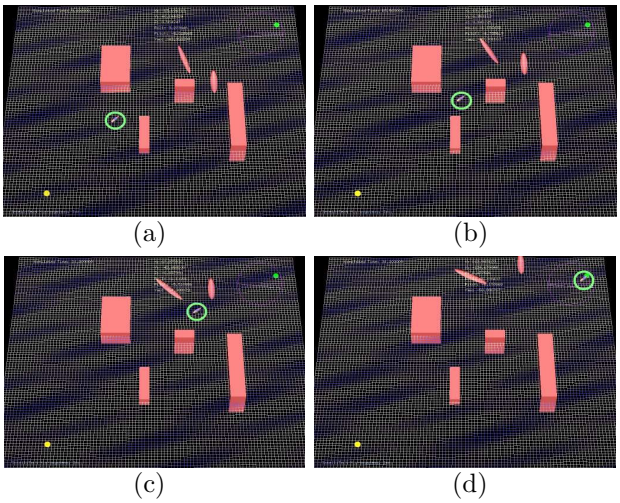

Fig. 10. Screenshots of local obstacle avoidance testing with moving obstacles(Yellow circle: starting location; Green circle: target location). For clarity,the position of the USSV has been highlighted by a light green ring in all four screenshots.

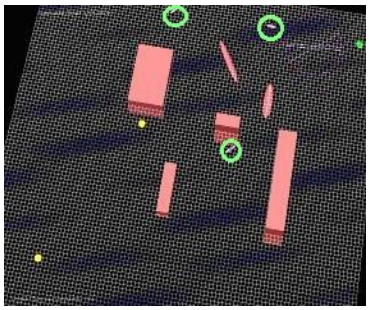

Fig. 11. Screenshot of local obstacle avoidance testing with 3 USSVs in multi-USSV simulator. Positions of the 3 USSVs have been highlighted by light green rings.

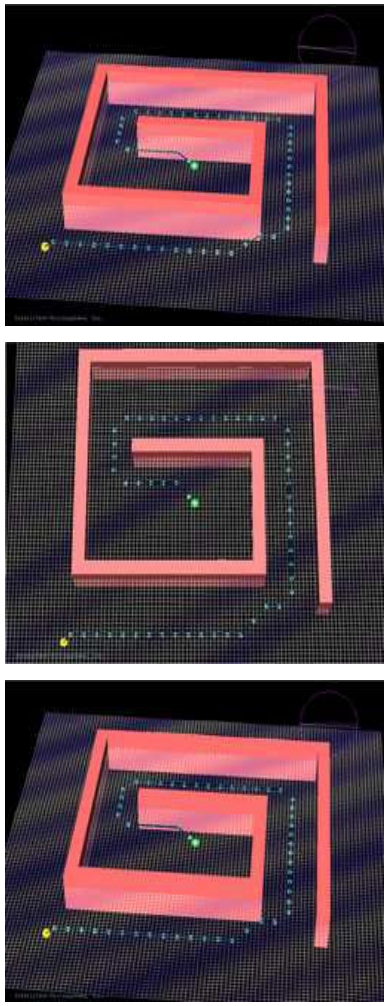

Fig. 12. Simulation tests of WAP performance. Yellow circle: starting location; Green circle: target location.

K. D. Do, Z. P. Jiang, and J. Pan. Underactuated ship global tracking under relaxed conditions. *IEEE Transactions on Automatic Control*, 47(9):1529–1536, Sep. 2002.

Y. Fang, E. Zergeroglu, M. S. de Queiroz, and D. M. Dawson. Global output feedback control of dynamically positioned surface vessels: an adaptive control approach. In *Proceedings of the American Control Conference*, pages 3109–3114, Arlington, VA, June 2001.

T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley and Sons, New York, 1994.

T. I. Fossen and A. Grovlen. Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. *IEEE Transactions on Control Systems Technology*, 6(1):121–128, Jan. 1998.

J. Goclowski and A. Gelb. Dynamics of an automatic ship steering system. *IEEE Transactions on Automatic Control*, 11(3):513–524, July 1966.

P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2):100–107, July 1968.

M. R. Katebi, M. J. Grimble, and Y. Zhang. H_∞ robust control design for dynamic ship positioning. *IEE Proceedings - Control Theory and Applications*, 144(2): 110–120, Mar. 1997.

P. Krishnamurthy and F. Khorrami. GODZILA: A low-resource algorithm for path planning in unknown environments. In *Proceedings of the American Control Conference*, pages 110–115, Portland, OR, June 2005. Also in the Journal of Intelligent and Robotic Systems, vol. 48, no. 3, pp. 357-356, March 2007.

P. Krishnamurthy, F. Khorrami, and S. Fujikawa. A modeling framework for six degree-of-freedom control of unmanned sea surface vehicles. In *Proceedings of the IEEE Conference on Decision and Control/European Control Conference*, pages 2676–2681, Seville, Spain, Dec. 2005.

P. Krishnamurthy, F. Khorrami, and T. L. Ng. Control design for unmanned sea surface vehicles: Hardware-in-the-loop simulator and experimental results. In *Proceedings of the 2007 International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct. 2007.

E. Lefeber, K. Y. Petterson, and H. Nijmeijer. Tracking control of an underactuated ship. *IEEE Transactions on Control Systems Technology*, 11(1):52–61, Jan. 2003.

A. Loria, T. I. Fossen, and E. Panteley. A separation principle for dynamic positioning of ships: theoretical and experimental results. *IEEE Transactions on Control Systems Technology*, 8(2):332–343, Mar. 2000.

F. Mazenc, K. Pettersen, and H. Nijmeijer. Global uniform asymptotic stabilization of an underactuated surface vessel. *IEEE Transactions on Automatic Control*, 47 (10):1759–1762, Oct. 2002.

T. L. Ng, P. Krishnamurthy, F. Khorrami, and S. Fujikawa. Autonomous flight control and hardware-in-the-loop simulator for a small helicopter. In *Proceedings of the IFAC World Congress*, Prague, Czech Republic, July 2005.