Proceedings of the ASME 2013 Dynamic Systems and Control Conference
DSCC2013
October 21-23, 2013, Palo Alto, California, USA

DSCC2013-3838

EXPERIMENTAL EVALUATION OF APPROACH BEHAVIOR FOR AUTONOMOUS SURFACE VEHICLES

Ivan R. Bertaska, Jose Alvarez, Armando Sinisterra, Karl von Ellenrieder, and Manhar Dhanak
Department of Ocean & Mechanical Engineering
Florida Atlantic University
Dania Beach, FL 33004-3023
ibertask@fau.edu

Brual Shah, Petr Švec, and Satyandra K. Gupta
Maryland Robotics Center
Department of Mechanical Engineering
University of Maryland
College Park, MD 20742

ABSTRACT

This article presents an experimental assessment of an Unmanned Surface Vehicle (USV) executing an approach behavior to several stationary targets in an obstacle field. A lattice-based trajectory planner is implemented with a priori knowledge of the vehicle characteristics. In parallel, a low-level controller is developed for the vehicle using a proportional control law. These systems are integrated on the USV control system using the Lightweight Communications and Marshalling (LCM) message passing system. Filtered vehicle-state information from onboard sensors is passed to the planner, which returns a least-cost, dynamically feasible trajectory for achieving the ascertained goal. The system was tested in a 750 m by 150 m area of the US Intracoastal Waterway in South Florida in the presence of wind and wave disturbances to characterize its effectiveness in a real-world scenario. The vehicle was able to replicate behavior predicted in simulations when navigating around obstacles. The approach distance to each target was favorably lower than the user-defined limit. Owing to the fact that the USV uses differential thrust for steering, the vehicle tracked the planned trajectories better at lower speeds.

INTRODUCTION

Currently, unmanned surface vehicles (USVs) are not truly autonomous, most rely on waypoint navigation and scripted operations for trajectory control. In order for USVs to operate in complex environments against adversaries that may use evolving tactics, a combination of deliberative, reactive and reflexive path planning is required. This may be accomplished through a realized set of behaviors that can be used to react to classes or types of situations in pre-programmed ways. However, the manual design of behaviors for a large variety of missions with unpredictable surface traffic, multiple vessels with different physical capabilities and in uncertain environments would require a significant effort and is not scalable. A proposed alternative is to automatically generate or tune the necessary behaviors using machine learning within a virtual simulation environment [1], [2]. The work reported herein is the first phase of a collaborative effort between the University of Maryland (UMD) Robotics Center and Florida Atlantic University (FAU) to develop and tune an approach behavior using simulation of the vehicle dynamics, and evaluate its performance on a physical unmanned surface vehicle platform. The evaluation scenario includes an interceptor USV approaching a target USV through an obstacle field.

In this paper, preliminary field tests of the approach behavior are presented. In these experiments, an interceptor USV navigates through a sequence of checkpoints in an obstacle field. The vehicle's trajectory is computed in real time and can be seen to change slightly from run to run owing mostly to the variation in the heading of the vehicle caused by environmental disturbances such as wind and waves. Prior experiments have utilized a follow behavior, where an interceptor USV tracks a target's trajectory, maintaining a set distance [4], but did not include the presence of an obstacle field when generating the interceptor's trajectory. In real world scenarios, impediments are very common and avoidance is crucial to the safe navigation of the vehicle. Kuwata et al. [3] adapted the velocity obstacles planner for reactive, COLREGS-compliant obstacle avoidance. In contrast, in this paper we utilize a lattice-based trajectory planner that considers the dynamics of the USV when computing a global, dynamically feasible trajectory to avoid static obstacles.

Using the developed trajectory planner as the main part of the approach behavior produces collision-free trajectories that are compliant with the vehicle dynamics. This allows tight integration of the planner with the lower-level controller, which leads to improved, collision-free maneuvering capability in a dense obstacle field. We decouple the search for the trajectory by separately considering its spatial and temporal aspects, i.e., we design a discrete control action set (also known as a set of motion primitives) and use it for planning. We then apply a lower-level controller to closely follow this trajectory. The control action set needs to be sufficiently rich to be able to compute dynamically feasible trajectories that can robustly guide the vehicle in a complex obstacle field. The number and type of control actions need to be determined so that they reflect the allowable maneuvering range of the vehicle and as such balance between the smoothness of the trajectory and computational requirements. The tuning process is carried out using a high-fidelity simulation environment. The dynamic model of the USV is developed through a combination of system identification and physical modeling. The near future aim is to utilize a dynamic model of the vehicle to not only evaluate and tune the developed behaviors in simulation but also use the model directly in the planning process in real-time.

The outline of this paper roughly mirrors the development process of the behavior generation system:

1) A dynamic simulation of a USV is developed through a combination of systems identification and physical modeling.

2) The real time, dynamically-feasible trajectory planner is tuned using simulation.

3) The planner is implemented in hardware as a high-level backseat driver that furnishes a trajectory to a USV's low-level controller as a set of waypoints.

4) Assessment and further testing of the system's performance is conducted through on-water testing.

DYNAMIC SIMULATION OF THE USV

The vessel utilized for preliminary testing of the planner is named the DUKW-Ling. This amphibious USV is a 1/7th scale model of a tracked, small waterplane area twin hull (SWATH), vehicle design concept called the DUKW-21, which was developed by the Center for Innovation in Ship Design at the Naval Surface Warfare Center, Carderock Division [5]. When water-borne, as in the tests reported here, the vehicle is propelled using a pair of 2-bladed, 12 Volt electric trolling motors. Each propulsion unit is capable of producing up to 133 N of thrust, giving the vehicle a top speed of about 1.5 m/s. A detailed description of the USV and its performance can be found in [8] and [9].

Equations of Motion

A three degree of freedom (surge x, sway y, yaw ψ – see Fig. 2.) dynamic simulation of the DUKW-Ling was created.

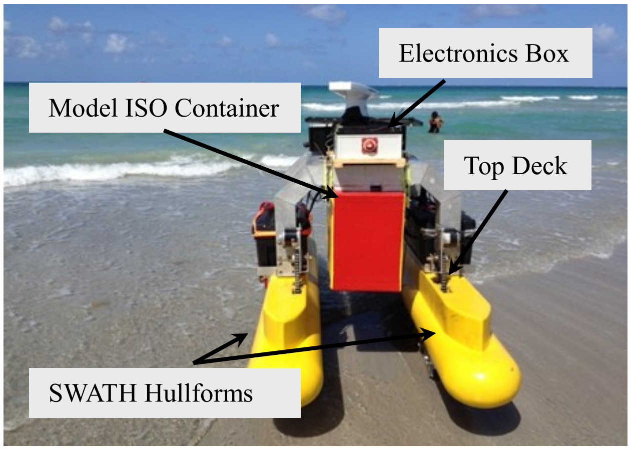

FIGURE 1 - THE DUKW-LING AMPHIBIOUS SURFACE VEHICLE

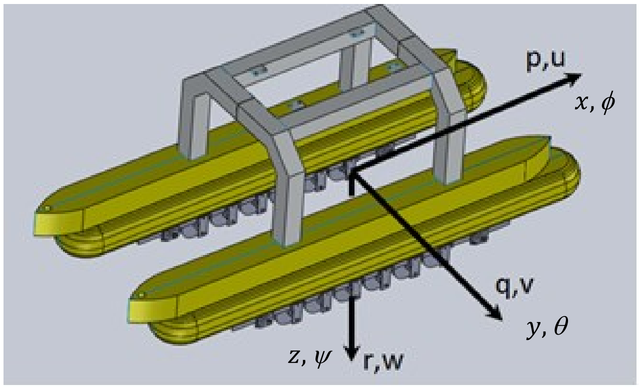

FIGURE 2 - VEHICLE COORDINATE SYSTEM

The origin of the body-fixed coordinate system is fixed at the USV's center of gravity. Assuming the USV is symmetrical about the xz plane, the equations of motion in the body-fixed frame can be reduced as in [7]:

$$
\begin{aligned}
X &= (m - X_{\dot{u}})\dot{u} - (m - Y_{\dot{v}})vr + Y_{\dot{r}}r^2 - X_u u \\
Y &= (m - Y_{\dot{v}})\dot{v} - Y_{\dot{r}}\dot{r} + (m - X_{\dot{u}})ur - Y_v v - Y_r r \\
N &= (I_z - N_{\dot{r}})\dot{r} - Y_{\dot{r}}\dot{v} + (X_{\dot{u}} - Y_{\dot{v}})uv - Y_{\dot{r}}ur \\
&\qquad - N_v v - N_r r
\end{aligned}
\tag{1}
$$

where (u, v, r) are the vehicle's surge, sway and yaw velocity, respectively, m is the vehicle's mass, I_z is the mass moment of inertia about the z-axis and (X, Y, Z) are the control forces in each direction. The uppercase values are drag and added mass coefficients that relate subscripted terms to the uppercase terms, e.g., X_u is the drag coefficient in the X direction due to velocity in the X direction and $X_{\dot{u}}$ is the hydrodynamic coefficient relating the acceleration in the X direction to the force in the X direction [7]. The vehicle speed is correlated to surge and sway velocity as $U = (u^2 + v^2)^{1/2}$.

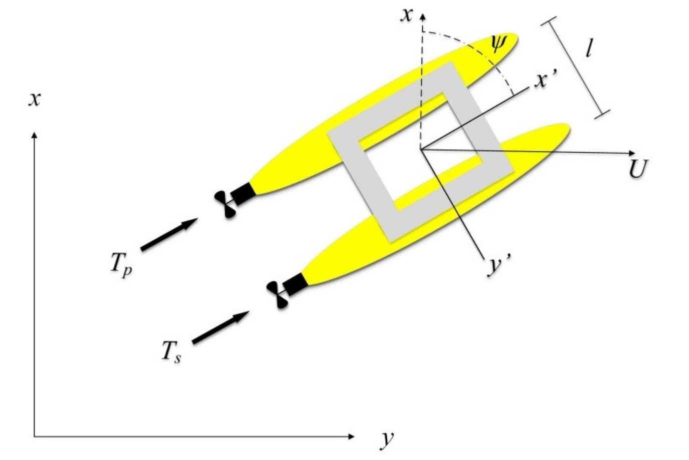

FIGURE 3 – PLANAR FORCE MODEL OF THE DUKW-LING

The DUKW-Ling is an under-actuated vehicle with control forces and moments acting only in the surge and yaw degrees of freedom. The USV relies on differential thrust for steering, as demonstrated in Fig. 3. A benefit of this is that rudder drag terms do not need to be estimated for the full equations of motion [7]. However, heading control cannot be decoupled from the surge speed, and the maximum moment generated by the propellers is inversely dependent on the surge thrust. Taking these factors into account, the equations of motion in Eq. (1) can be expanded to include the applied forces and moments (relevant values can be found in Tab. 1).

$$\begin{aligned} X &= T_p + T_s \\ Y &= 0 \\ N &= (T_p - T_s)\frac{l}{2} \end{aligned} \qquad (2)$$

While the vehicle's mass and dimensions were directly measured, physical modeling was used to estimate the added mass and drag coefficients in Eq. (1). As part of the effort to fully characterize the DUKW-Ling, a series of systems identification field tests were performed [8]; these tests included pull-out tests, circle tests and zig-zag tests. Open loop dynamic simulations of the vehicle using the low fidelity model in Eq. (1) were validated with data from the systems identification field tests.

TABLE 1 - DUKW-LING VEHICLE PROPERTIES

Vehicle Property	Variable Name	Value
Length Overall	L	2.69 m
Beam Overall	B	1.14 m
Separation of Propellers	l	0.82 m
Max. Thrust	$T_{p,max}$, $T_{s,max}$	133 N
Mass	m	295 kg

TRAJECTORY PLANNING AND FOLLOWING

The task for the trajectory planner is to find a collision-free trajectory between two given poses of the USV. More formally, given

(i.) a discrete state space $X \in \mathbb{R}^2 \times \mathbb{S}^1$ in which each state $\mathbf{x} = [x, y, \psi]^{\mathrm{T}}$ consists of x and y position coordinates in the North, East, Down (NED) coordinate system [8], and the orientation ψ (i.e., roll, pitch and heave state variables are neglected in our application),

(ii.) the current state $\mathbf{x_I} = [x_I, y_I, \psi_I]^{\mathrm{T}} \in X$ of the USV and its target state $\mathbf{x_G} = [x_G, y_G, \psi_G]^{\mathrm{T}} \in X$,

(iii.) a discrete set $\mathrm{U_c}(\mathbf{x_j}) = \{\mathbf{u_{c,1}}, \mathbf{u_{c,2}}, \dots, \mathbf{u_{c,M}}\} \in \mathbb{R}^2 \times \mathbb{S}^1$ of dynamically feasible control actions (i.e., motion primitives), where each control action $\mathbf{u_{c,k}} = [u_d, \delta t, \psi_d]^T$ consists of the desired surge speed u_d, the execution time δt and the desired heading ψ_d in NED, and is used to map the state $\mathbf{x_j} \in \mathrm{X}$ to its neighboring state $\mathbf{x_{j,k}} \in X$ for $\mathrm{j} = 1,2, \dots, |X|$, and $\mathrm{k} = 1,2, \dots, |\mathrm{U_c}(\mathbf{x_j})|$, and

(iv.) a discrete set $X_{obs} \subset X$ of obstacle states.

one must compute, a collision-free, dynamically feasible trajectory $\tau: [0, t] \rightarrow X_{free}$ such that $\tau(0) = \mathbf{x_I}$ and $\tau(t) = \mathbf{x_G}$, where t is the execution time of the trajectory and $X_{free} = X \backslash X_{obs}$ represents the obstacle free region of the state space.

The USV experiences motion uncertainty caused by ocean waves and wind, which may cause it to drift towards obstacles. Hence, to avoid obstacles and handle pose errors while approaching the target, the trajectory τ needs to be recomputed in each planning cycle by the USV.

State-Action Space Representation

The pose-based search is carried out over a multi-layered 3D lattice structure $L: \mathbf{x_j}, \mathbf{u_{c,k}} \rightarrow \mathbf{x_{j,k}}$ that maps states $\mathbf{x_j} \in X$ using control actions $\mathbf{u_{c,k}} \in U_c(\mathbf{x_j})$ to their neighboring states $\mathbf{x_{j,k}} \in X$ for $j = 1,2, \dots, |X|$, and $k = 1,2, \dots, |U_c(\mathbf{x_j})|$. Each layer in the lattice thus represents a specific 2D planning space with the fixed vehicle heading ψ. The state $\mathbf{x_j}$ represents a squared region of X inside of which the pose $\mathbf{x_j}$ is considered to be constant.

The control actions are determined with the help of the simulation model of the USV, as presented above. The model is used to define suitable neighboring poses that can be reached by the vehicle, given its speed. Hence, in the lattice, each $\mathbf{u_{c,k}} \in U_c(\mathbf{x_j})$ generates a local pose $\mathbf{x_{k,g}} = [\Delta\, x_k, \Delta\, y_k, \Delta\, \psi_k]^T$ (defined in the body frame of the vehicle) that is guaranteed to be reachable from the vehicle state. Each pose is conveniently selected to be the center of a corresponding grid cell and its orientation component ψ is required to match one of the discrete heading values.

$$\mathbf{x_{j,k}} = \begin{bmatrix} x_j \\ y_j \\ \psi_j \end{bmatrix} + \begin{bmatrix} cos\Delta\psi_j & -sin\Delta\psi_j & 0 \\ sin\Delta\psi_j & cos\Delta\psi_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta\psi_k \end{bmatrix} \qquad (3)$$

The lattice representation reduces the computational demand of the search (e.g., in comparison to sampling-based motion planning approaches [11]), while explicitly considering the differential constraints of the vehicle. It is up to the user of the planner to determine the resolution of the state space, the number of control actions and their characteristics.

Trajectory Computation

The planner searches the lattice in the least-cost, heuristic A* fashion [11] using the cost function $f(\mathbf{x}) = g(\mathbf{x}) + \epsilon h(\mathbf{x})$ in which $g(\mathbf{x})$ is the optimal cost-to-come from $\mathbf{x_I}$ to $\mathbf{x}$, $h(\mathbf{x})$ is the heuristic cost-to-go between $\mathbf{x}$ and $\mathbf{x_G}$, and ϵ is the inflation factor. The cost-to-come is computed as $g(\mathbf{x}) = \sum_{q=1}^{Q} l(\mathbf{u_{c,q}})$ over Q planning stages, where $l(\mathbf{u_{c,q}})$ is the execution time of the control action $\mathbf{u_{c,q}}$. The time $l(\mathbf{u_{c,q}})$ is set to ∞ if the control action makes the vehicle transition to a collision state $\mathbf{x_{col}} \in X_{obs}$ from $\mathbf{x} \in X_{free}$. The heuristic $h(\mathbf{x})$ is expressed as the required time to reach $\mathbf{x_G}$ along a straight line from $\mathbf{x}$. The heuristic increases the speed of the search by focusing it upon the most promising regions of the state space, while still allowing resolution optimal planning. The inflation factor $\epsilon \geq 1$ allows determining a delicate balance between the computational demand of the planner and the optimality of the trajectory.

The computed trajectory $\tau = \{\mathbf{u_{c,1}}, \mathbf{u_{c,2}}, \dots, \mathbf{u_{c,Q}}\}$ is translated into $Q+1$ waypoints $\{\boldsymbol{w_1}, \dots, \boldsymbol{w_{Q+1}}\}$ for waypoint following as described in the next subsection. Each waypoint $\boldsymbol{w_i} = [x_i, y_i, u_i]^T$ consists of x and y coordinates of the desired position and the surge speed u. In addition, we smooth the trajectory using the algorithm introduced in [12]. This results in a realistic trajectory without unnecessary deviations, which leads to reduced oscillations in the vehicle's motion (if a low-resolution control action set is used).

Waypoint Following

Owing to the under-actuation of the system, perfect course-keeping between waypoints is not feasible, since no control force is present in the sway direction. Thus, the vehicle utilizes a line-of-sight (LOS) guidance system driven by a proportional heading controller for trajectory following, causing the vehicle to point to the next consecutive waypoint, rather than focusing on reducing cross-track error for the path between those waypoints. For the purpose of this experiment, this was deemed acceptable, as the distance between consecutive waypoints was substantially smaller than that found in a typical cross-track error mitigating scenario.

Due to the differential steering, the vehicle heading controller differs from those that utilize rudder steering systems as in [4]. Differential steering imposes a moment about the vehicle that causes it to turn. There are distinct advantages to this type of system; primarily, the turning radius for the vehicle is substantially reduced, as well as the addition of the capability to steer without the requirement of a forward speed. With this configuration, the controller manages vehicle direction by two separate proportional control laws. Each of these controllers focuses on a single motor. Owing to the yaw NED convention, the port controller would have a positive slope, while the starboard controller would have a negative slope. A forward speed is denoted by a percentage of full-scale motor power and is used to bias the proportional control law to the desired steady-state motor percentage. This bias value is constant between both motors and determines the vehicle's surge speed, and, inversely, how much motor capacity is reserved for differential thrust. To better handle real world scenarios, a tolerance region is implemented where the error is less than five degrees, as well as a saturation value above 100% and below -100%. This is to account for heading disturbance errors caused by wind and wave forces. When within this tolerance region, both motors would have the same percentage and the same thrust, ideally causing the vehicle to follow a straight line. Figure 4 displays an example of the proportional control law for both motors with a positive thrust. The absolute value of the slope of the motor command corresponds to the proportional gain value. More information on the vehicle controller can be found in [8].

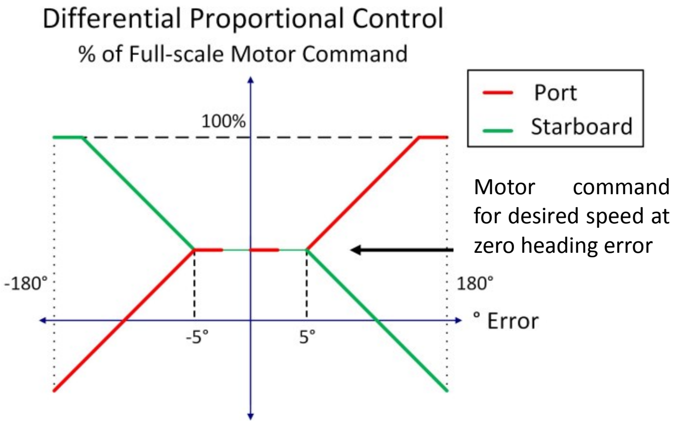

FIGURE 4 - HEADING CONTROLLER FOR DIFFERENTIAL THRUST ON THE DUKW-LING

HARDWARE IMPLEMENTATION

Owing to its low-latency, high data transmission rate, and adaptability to different operating systems [5], [10], the Lightweight Communications and Marshalling (LCM) message passing system was chosen for handling communication between the high-level trajectory planner and the low-level vehicle controller. Vehicle state information is determined using a differential GPS and a double gimbaled compass, whose data output is filtered with a Kalman filter [8]. The information from these sensors is packaged in an LCM structure and sent to the high-level planner, which returns an array of waypoints in another LCM structure, corresponding to the desired vehicle trajectory to its goal.

Vehicle Hardware Architecture

For this experiment, the DUKW-Ling is furnished with two onboard computers. The UMD backseat driver operates on a ruggedized Windows 7-based laptop, while the low-level vehicle controller is operated on a Technologic Systems TS-7800 single-board computer (SBC) running the Squeeze distribution of Debian Linux. The two computers are networked through a single Ethernet cable, allowing the passing of data between the two levels of the Guidance, Navigation and Control (GNC) architecture.

A sensor suite is implemented on the vehicle to obtain vehicle state information for both the low-level controller and the high-level planner. The suite consists of a Hemisphere Inc. differential global positioning system (DGPS), an Ocean Server Inc. OS-5000 tilt-compensated compass and an MTi-G inertial measurement unit (IMU). The first two sensors provide measurements of the vehicle position and heading, while the last provides a measurement of the vehicle's sway and surge speeds into the state estimator. The readings from these devices are processed by individual drivers on the SBC, as described later in this paper.

The vehicle also contains a Polulu Mini-Maestro, which is used as a pulse-width modulation (PWM) generator, and a radio frequency (RF) modem. The former directs the port and starboard motor controllers individually. This device is configured to allow supervisory override control with the implementation of a radio control (RC) receiver. The RF modem is used in conjunction with a corresponding modem at a shore base to access the low-level vehicle controller, as well as display real-time vehicle state information.

VEHICLE SOFTWARE ARCHITECTURE

Implementation in LCM

As LCM has been shown to work well for unmanned vehicle systems [10], it was used as the framework of the software architecture for the system. LCM is a communication protocol that relies on a publish/subscribe model to pass messages between processes and threads, further known as interprocess communication (IPC). It utilizes a user datagram protocol (UDP) multicast to dispatch messages to all systems in scalable networks, thus negating the need for a main "server" mediator to arbitrate between subscriber and publisher. Furthermore, the choice of UDP allows for a fast data transfer rate with failed messages simply being dropped, without being checked by the publisher, as in transmission control protocol (TCP).

Low-Level Software Architecture

Figure 5 demonstrates the vehicle software architecture with the implementation of LCM. Each sensor described in the previous section is handled by a separate driver operating in parallel and has an exclusive LCM channel. These drivers parse the sensor information and distribute it to the discrete LCM channels. The pertaining information is then received by the state estimator, which processes the vehicle data by applying Kalman filters to obtain a vehicle state measurement [8]. This process handles the publishing of this information to the low-level controller, as well as to the high-level trajectory planner.

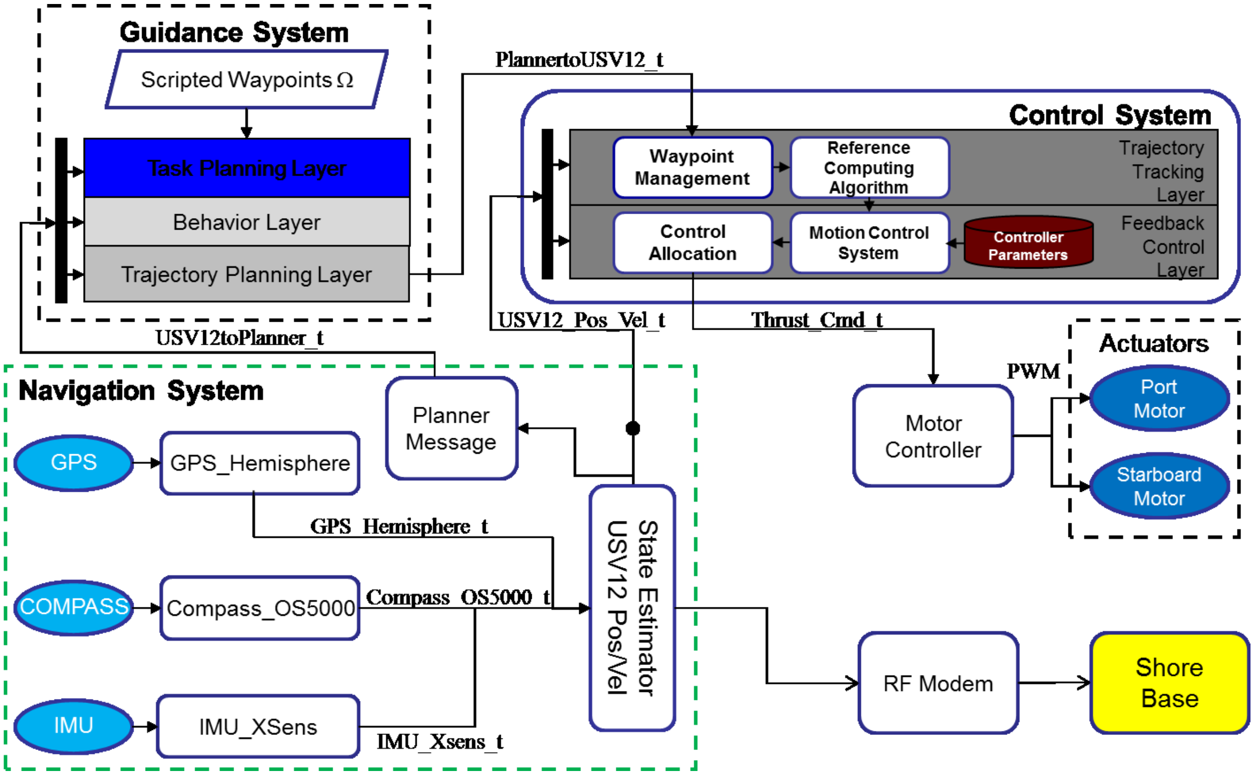

FIGURE 5 - LCM COMMUNICATION SCHEMATIC BETWEEN SENSORS (NAVIGATION SYSTEM BLOCK), LOW-LEVEL CONTROLLER (CONTROL SYSTEM BLOCK), AND TRAJECTORY PLANNER (GUIDANCE SYSTEM BLOCK)

When the data reaches the heading controller, the vector between the current vehicle position and the next waypoint in the trajectory is used to calculate a reference heading and a distance, which are passed to the controller. For modularity and debugging motives, these two processes are separated and run in parallel. Using the control law proposed in the previous section, an error between the headings is found, and two motor commands are computed and sent to both actuators.

LCM-Based Communication with Backseat Driver

Once the estimated state is established, the backseat driver is updated through another LCM message. To calculate the required trajectory, the planner requests the vehicle's state, as well as the ultimate goal of the route with the desired heading. Included in these calculations are any obstacles present in the field that the vehicle has identified.

The backseat driver takes into account vehicle dynamics and its minimum turning radius, to compute a user-configured trajectory between the current vehicle pose and the goal pose. Configurable parameters include mission area boundaries in the NED coordinate system, publish rate, the number of vehicle actions (i.e., motion primitives) and their types to consider in planning.

A timer is set between the communication from the controller to the backseat driver, so that a new trajectory is calculated only after a user-set period. The planner then publishes this new message at a much faster rate between vehicle update periods. The waypoint management system takes this into account, and only processes trajectories that differ from the last known update. It then publishes the first waypoint and surge velocity to the vehicle controller.

EXPERIMENTAL SETUP

The main objective of the experiments described here was to characterize vehicle performance in an approach behavior with the addition of the high-level planner in an obstacle-laden environment. The approach behavior constitutes a maneuver typical of a scenario where a USV would advance towards a target vessel in a static position. This example can be illustrated by a security USV approaching a moored suspect vessel. For this experiment, the static vehicle was mocked with reference GPS checkpoints. To better determine the conduct of the DUKW-Ling, a series of these checkpoints were established so that the vehicle would progress from one checkpoint to the next, before finally returning to the dock.

Mission Area Location and Obstacle Field

In choosing the location for this experiment, the target mission area had to possess certain criteria. It had to be large enough to not confine the vehicle's path with the inclusion of several obstacles; it had to be sheltered from moderately rough wave and current disturbances; and it had to display a low amount of boat traffic. Taking these considerations into account, a suitable destination was found at a partially enclosed inlet off the US Intracoastal Waterway in Hollywood, Florida known as North Lake, pictured in Fig. 7.

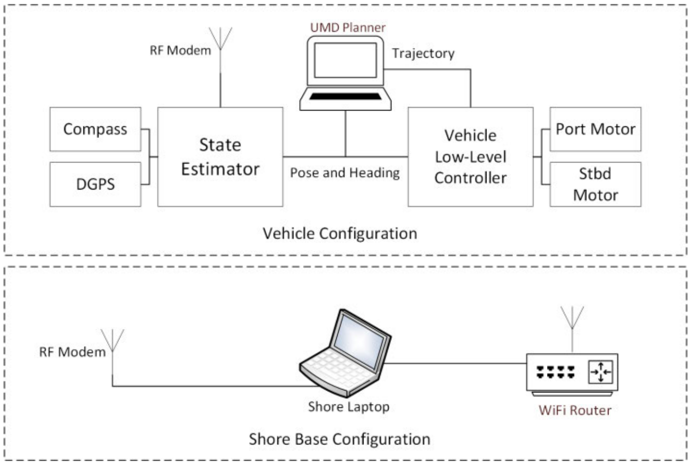

FIGURE 6 - EXPERIMENTAL SETUP FOR APPROACH BEHAVIOR

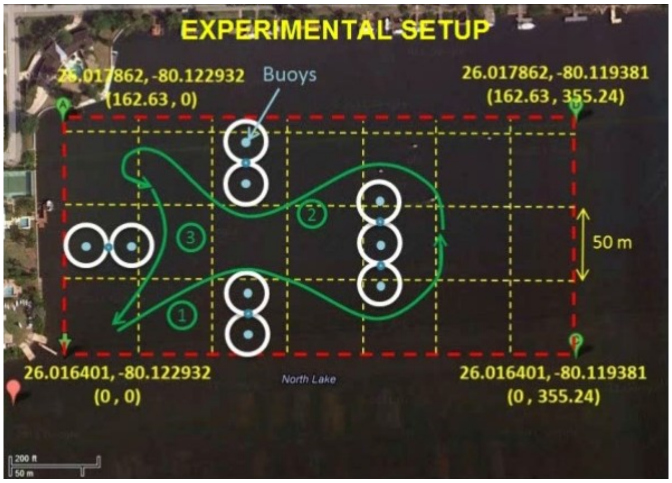

FIGURE 7 - EXPERIMENTAL EVALUATION OF VEHICLE APPROACH BEHAVIOR

A mission area was configured around the geographical location of the lake, with boundary areas encompassing the four sides in a rectangular fashion. A set of obstacles was placed in the arrangement depicted in Fig. 7, with an approach behavior destination set so that the vehicle would encounter an obstruction in its path. Each white circle in the picture represents the outer circumference of an area that the planner has taken to be completely filled in with obstacles. To visually assess vehicle behavior, a buoy is placed at the center of these areas.

The vehicle was loaded with a priori knowledge of the target approach checkpoints. This is to simulate future development in this project where the system will autonomously recognize its target. Three checkpoints were chosen at opposing sides of obstacles, with the final one corresponding to a return to dock. In a cyclical fashion, the vehicle would visit each of these points, querying the planner with a new location once the vehicle has achieved a user-set distance from the target.

Hardware Setup and Experiment Conduction

For this test of the vehicle, there were two main configurations utilized for communicating with the vehicle. The vehicle was loaded with the low-level controller and sensor suite, as well as the planner software on a separate laptop networked to the system. To access the vehicle during the mission, a shore base configuration was introduced with both an RF modem and a wireless router. The latter was used to create a remote desktop on the planner laptop at ranges close to shore, while the former was used to access the SBC at all ranges during the experiment, as shown in Fig. 6. Visual feedback was provided through the remote access to the vehicle controller to estimate the vehicle response and determine whether it had encountered an issue. As a safety precaution, a human-operated chase boat was taken to the mission site, with the ability to initiate supervisory override control. All tests in this paper occurred on November 20, 2012 and December 11, 2012, between the hours of 9:00 AM and 5:00 PM and 8:30 AM and 4:30 PM, respectively. This exposed the vehicle to both calm and wind-prone conditions to test the response of the proportional controller in varying environments.

Experimental Results and Evaluation

The experiment was judged on three criteria: first, how accurately the vehicle ran with respect to the given trajectory; second, the behavior of the vehicle around the soft obstacles; and lastly, the vehicle response at each approach behavior goal. With these three criteria in mind, it was imperative to accurately characterize the interaction between the backseat driver and the low-level controller. The three parameters that could be changed within the planner to accommodate the vehicle controller were the publish rate of new trajectories to the vehicle, the distance between waypoints in those trajectories and the distance away from the goal where the planner would no longer be producing feasible trajectories (i.e. the minimum distance from the approach behavior).

Prior to the experiments, a series of tests were used to fine tune those variables to produce an acceptable result. Problems incurred with incorrect values included oscillating responses to new trajectories and skipping of intermediate waypoints. It was determined that the optimal value for the publish rate was set at 20 second intervals; faster rates would cause the heading controller to compensate too quickly and cause an oscillation against the desired trajectory.

The planner was configured by setting the resolution of x and y state variables to 1 m, and the resolution of the heading ψ was set to 0.785 rad. This resulted in eight discrete orientation levels of the vehicle. The control actions and their corresponding local poses $x_{k,g}$ were selected conservatively in the body frame of the vehicle in order to guarantee that they could be reached even with the maximum vehicle speed of 1.5 m/s. For this experiment, three control actions for a particular orientation and their corresponding final poses, e.g., for $\psi = 0$,$\mathrm{x}_{1,\mathrm{g}} = [6\text{ m}, -4\text{ m}, 5.495\text{ rad}]^{\mathrm{T}}$, $\mathrm{x}_{2,\mathrm{g}} = [7\text{ m}, 0\text{ m}, 0\text{ rad}]^{\mathrm{T}}$, and $\mathrm{x}_{3,\mathrm{g}} = [6\text{ m}, 4\text{ m}, 0.785\text{ rad}]^{\mathrm{T}}$ were defined in the body frame of the vehicle. Hence, the length of each projected control action was about 7 m. The final states of the control actions for the remaining orientations were determined similarly. The radius of the circle-of-acceptance for the final goal was set at 15 m, while for all other intermediate waypoints was set at 5 m. Results from one of the runs of the experiment can be found in Fig. 8. The vehicle behaved according to predictions as it ran a tight path around the southern and northern edge of the easternmost obstacles. However, due to wave and wind disturbances, a considerable drift to the south occurred, and the vehicle entered the margin of some of the soft obstacles. This was deemed an acceptable error, and future versions of the planner would account for a larger area of soft obstacles. As seen in Fig. 8, the distance to the goal position was much lower than the user-configured limit, indicating a close approach behavior, which is favorable in this maneuver.

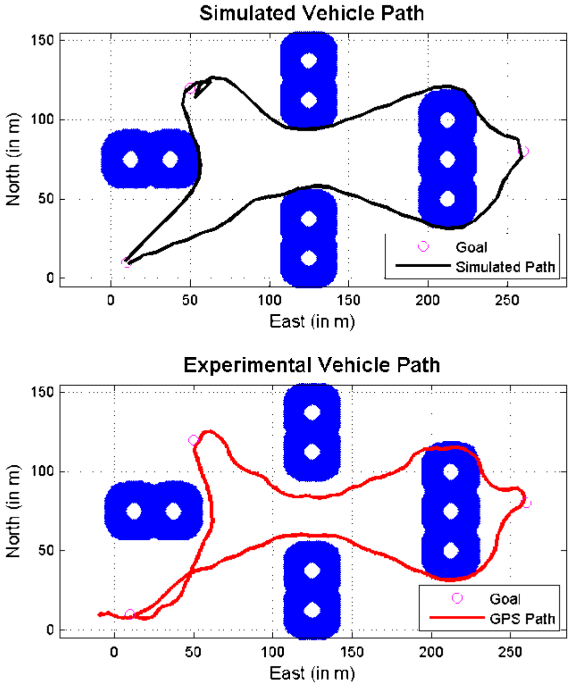

FIGURE 8 - SIMULATED AND EXPERIMENTAL VEHICLE TRAJECTORY AT TEST SITE WITH 40% MOTOR SPEED

During all tests, the DUKW-Ling USV was running between 40% to 70% of motor capacity reserved for forward motion; thus, the heading controller only had the remaining percentage to correct for heading, as described in vehicle dynamics above. The former caused a scenario where the vehicle ran slower, but had a higher moment generated about it, while the latter had the vehicle running faster but with less ability to control its heading. All tests performed with 40% of motor capacity dedicated to surge speed proved successful with the vehicle obtaining ample power to overcome wind and wave disturbances that perturbed the trajectory.

The November 20, 2012 experiment resulted in a successful run with 70% motor power, while the December 11, 2012 experiment proved unsuccessful at 70% due to high wind gusts in excess of 8.2 m/s in combination with large wave disturbances. Although the vehicle did complete the run, it was underpowered and could not maintain a steady bearing. This caused the vehicle to enter the same obstacle field twice. The additional tests carried out that day used 40% and 60% motor capacity.

Table 2 displays the success of each run as determined by motor capacity and environmental conditions. All environmental conditions were obtained from the National Oceanic and Atmospheric Administration (NOAA) Station PVGF1 located approximately 8 kilometers from the test site.

TABLE 2 - OBSTACLE AVOIDANCE PER RUN – DECEMBER 11, 2012

Run Number	Motor % for Surge Speed	Number of Obstacles Entered	Average Sustained Wind Speed	Average Wind Gust
1	40%	2	4.5 m/s	6.0 m/s
2	60%	2	5.3 m/s	7.1 m/s
3	70%	3	5.8 m/s	8.1 m/s

CONCLUSIONS AND FUTURE WORK

Preliminary on-water tests of a USV approach behavior have been conducted. The behavior encompasses a lattice-based trajectory planner that is capable of computing trajectories compliant with the vehicle dynamics. The USV behaved according to simulated predictions when navigating around obstacles, and it was found that the approach distance to each target was favorably lower than the user-defined limit. Owing to the fact that the USV uses differential thrust for steering, the vehicle tracked the planned trajectories better at lower speeds.

As part of the ongoing research project, we will enhance and implement the same GNC system on a USV with a higher design speed, as to better model an interceptor. This will also be used in conjunction with a speed controller, as the planner will have the ability to determine the time the USV reaches each waypoint. Both levels of the GNC system will be arranged so that a higher update rate between the low-level controller, and the backseat driver would not cause a large oscillation in vehicle trajectory. The static approach behavior will be augmented into real-time vehicle tracking, where the interceptor will follow a target vehicle, maintaining a set distance from it while it navigates an obstacle-laden environment. The target vehicle will be semi-autonomous following a pre-defined path, with the occasional involvement of supervisory override control to test the interceptor. Similarly, a combination of a global and local physics-aware, trajectory planner will be developed to robustly handle complex COLREGs situations.

ACKNOWLEDGMENTS

This work was supported by the U.S. Office of Naval Research under Grants N00014-11-1-0423 and N00014-12-1-0502, managed by K. Cooper and R. Brizzolara, respectively.

REFERENCES

[1] Švec, P. and Gupta, S., 2012. "Automated Synthesis of Action Selection Policies for Unmanned Vehicles Operating in Adverse Environments". *Autonomous Robots*, **32**(2), pp. 149-164.

[2] Thakur, A., Švec, P., and Gupta, S., 2012. "GPU Based Generation of State Transition Models Using Simulations for Unmanned Sea Surface Vehicle Trajectory Planning". *Robotics and Autonomous Systems,* **60**(12), pp. 1457-1471.

[3] Kuwata, Y., Wolf, M., Zarzhitsky, D., and Huntsberger, T., 2011. "Safe Maritime Navigation with COLREGS Using Velocity Obstacles". *IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 4728-4734.

[4] Bibuli, M., Caccia, M., Lapierre, L., and Bruzzone, G., 2012. "Guidance of Unmanned Surface Vehicles". *IEEE Robotics and Automation Magazine*, September, pp. 92–102.

[5] Huang, A., Olson, E., and Moore D., 2010. "LCM: Lightweight Communications and Marshalling". *International Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan.

[6] Flom, B., 2009. "DUKW-21 Autonomous Navigation: Autonomous Path Planning for an Amphibious Vehicle". NSWCCD-CISD–2009/008, Ship Systems Integration & Design Department Technical Report, Naval Surface Warfare Center, Carderock Division, West Bethesda, MD.

[7] Fossen, T., 1994. Guidance and Control of Marine Vehicles, 1st ed., John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex PO19 1UD, England, Chap. 2, pp. 25–38.

[8] Alvarez, J., 2013. "Nonlinear Adaptive Control of an Amphibious Vehicle". In Proc. *ASME Dynamic Systems Control Conference*, 2013.

[9] Marquardt, J., Alvarez, J., and von Ellenrieder, K., 2013. "Characterization and System Identification of an Unmanned Amphibious Tracked Vehicle". *IEEE J. Oceanic Engineering.* (submitted, under review)

[10] Bingham, B, Walls, J., and Eustice, R., 2011. "Development for a Flexible Command and Control Software Architecture for Robotic Marine Applications".

Marine Technology Society Journal, **45**(3), May/June, pp. 26-36.

[11] LaValle, S., 2006. Planning Algorithms, Cambridge University Press.

[12] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., and Hoffmann, G., et al., 2007. "Stanley: The robot that won the DARPA Grand Challenge". The 2005 DARPA Grand Challenge, *Springer Tracts in Advanced Robotics*, **36**, pp. 1-43.