A USV Platform for Surface Autonomy

Armando Sinisterra, Manhar Dhanak, and Nicholas Kouvaras
SeaTech, The Institute for Oceans and Systems Engineering
Department of Ocean and Mechanical Engineering
Florida Atlantic University
Dania Beach, USA

***Abstract*—Development of an autonomous unmanned surface vehicle (USV) that may serve as a mother ship for a small unmanned aerial vehicle (UAV) and a small autonomous underwater vehicle (AUV), in support of surface autonomy in sensing and surveillance missions, is described. The USV development includes implementation of low-level controls, stereovision perception, and a task-oriented high-level planner, which includes vehicle localization, mapping of the surrounding and path planning as well as accounting for navigation regulations such as COLREGS. Results of the ongoing development and execution of waypoint-based, and vision-based autonomous missions for both the USV and the UAV using the Pixhawk / Ardupilot framework are described.**

Keywords—marine robotics; artificial intelligence; stereovision; simultaneous localization and mapping; path-planning; low-level controls; Pixhawk; Ardupilot; unmanned surface vehicles.

I. INTRODUCTION

Making unmanned surface vehicles (USVs) fully autonomous as well as undertake cooperative tasks with other unmanned vehicles continues to present significant challenges. These challenges pertain to interactions between multiple systems and sensors onboard the USV, control of the orientation and speed of the USV, navigation, obstacle avoidance, inter-vehicle communication and intelligent adaptive path planning. A suite of onboard sensors and actuators are required, including a perception system in support of sensing and modeling the physical world in which the USV will operate, while maintaining awareness of local static and dynamic obstacles. The stochastic nature of the motion of the USV in a variable environment and the inherent sources of error in the measurements of the sensors require a probabilistic approach that complements the perceived information and enables computation of a better estimate of its surroundings as well as the geo-localization of the USV. This is accomplished through the Simultaneous Localization and Mapping (SLAM) techniques (see for example, [1] and [2]).

World modeling can be expanded through use of a collection of unmanned vehicles (UXVs) working in a cooperative autonomous mission. Each vehicle having its own perception system and a particular version of its operational context, which can be shared with other vehicles either through a centralized network as in [3],[4],[5], or a decentralized network as in [6],[7],[8], where each robot populates its own map and incorporates the information from others if they lie within the communications range. Once a vehicle has estimated a map of the surroundings, and has solved the localization problem, it then has to compute a trajectory to get to his next goal. A number of path planning methods and improvements have been developed over the last few years. Popular algorithms include Dijkstra's shortest path algorithm [9], and its heuristic-based variation: the A* algorithm [10], both of which are deterministic techniques. Alternatively, the probabilistic roadmap algorithm is the most popular randomized technique [11], [12], [13].

A computed trajectory to a goal needs to be translated into actuator commands for the vehicle, in a way that its motion traces the computed path as closely as possible. This constitutes the low-level control. A dynamic or kinematic model of the robot is required, along with the most suitable control technique, which will ultimately depend on the type of motion and domain of the vehicle. Particularly for USVs, where the vehicle has to constantly overcome the forces induced by waves, wind and ocean currents, some low-level controllers have proved to be more effective than others [14].

Here, work in progress involving development of a USV to serve as a mother ship for an unmanned aerial vehicle (UAV) as well as a small autonomous underwater vehicle (AUV), in support of surface autonomy in marine environments is described. In Section II the wave adaptive modular vehicle (WAM-V), as the vehicle of choice and its relevant characteristics, along with the design, and construction of the main control unit (MCU), the Pixhawk navigation controller, and its open source software Ardupilot that have been implemented on the USV are first described. Next, the Odroid-XU4 single board companion computer selected as the high-level mission planner onboard the vehicle and the software used to communicate with Pixhawk are described. This is followed by a description of the ground control station software (Mission Planner) as part of the Pixhawk framework. In Section III results of autonomous missions executed in a marine environment at low sea-states, along with work developed for autonomous launch and recovery (L&R) operations is presented. Ongoing work in the area of USV autonomy is described in Section IV and concluding remarks are provided in Section V.

978-0-692-94690-9/$31.00 ©2017 MTS

II. USV Platform Description

A. WAM-V: Wave Adaptive Modular Vehicle

The 16 ft WAM-V USV (Fig. 1) by Marine Advanced Research (MAR), Inc [15] provides a suitable developmental platform for implementing autonomous capability on a USV. It is an overall small and lightweight structure with a high payload capacity, allowing easy handling and accommodation of all the required sensors, electronics, and smaller autonomous vehicles in its center tray. An independent suspension system per (inflatable) pontoon isolates the center tray from motions induced by incident waves, allowing for a stabilized deck so that it is suitable for the data acquisition process of all sensors onboard the WAM-V and a convenient landing spot for UAVs.

The center tray of the WAM-V connects to the pontoons through a set of articulated bars and the suspension system. Two electric outboard engines and its corresponding linear actuators (for steering) make up the propulsion unit at the back end of each pontoon. Two lithium batteries are used as the battery bank that powers the propulsion and steering systems as well as the navigation, communication or other supporting equipment on board. A propulsion control unit (PCU) is used to control and monitor the state of the thrusters and linear actuators. TABLE I provides a description of the main specifications of the vehicle.

B. Main Control Unit

The main control unit (MCU) includes the guidance, navigation and control systems of the vehicle, incorporating low and high level control. Fig. 2 shows the components of the MCU box, enumerated as: (1) 12V / 7Ah Pb battery, (2) floating mode battery charger, (3) transceiver, (4) PWM to serial adapter, (5) R/C receiver, (6) Pixhawk low-level controller, (7) external GPS and compass module, (8) Odroid single board companion computer, (9) router.

1) Pixhawk: low-level controller. The low level controller is based on the Pixhawk platform, which is an independent, open-hardware/open source platform aimed at providing high-end autopilot hardware at low cost and easy access, primarly for *[16]*. It includes a set of built-in navigation sensors such as accelerometers, gyroscopes, magnetometers and barometric pressure sensor. It provides the low level controllers for the GPS and inertia based navigation sensors. It reads from an external GPS and compass unit and has the capability to communicate to a ground control station (GCS) through a telemetry transceiver. Pixhawk's open-source software include a selection from the following options:

- PX4 Flight Stack: flight control software that consists of a collection of individual modules and runs on top of the PX4 middleware layer.
- APM Flight Stack: flight control software that consists of a single module and also runs on top of the PX4 middleware layer.

Our particular implementation runs on the APM Flight Stack, through the use of one of the variations of the Ardupilot *[17]* platform called Ardurover, version 3.1.0.

2) Odroid XU-4: high-level planner. In the standard configuration of an autonomous vehicle, this single board computer constitutes the companion computer, and its main purpose is to act as a virtual captain onboard the WAM-V. It gathers information from the navigation and inertial sensors, as well as from the perception system (such as the stereo vision camera, Lidar, Radar, etc), and computes intelligent behaviors in the form of safe navigation trajectories, avoiding collision with potential obstacles. The capability is been further expanded to comply with navigation regulations (such as COLREGS), and subsequently to consider such complexities as keeping a particular formation in a swarm of autonomous vehicles, or other specific task-related behavior. These behaviors are then translated into specific commands sent from the low level controller (Pixhawk) to the propulsion

TABLE I. WAM-V especifications

Length (m)	4.88
Beam (m)	2.44
Payload (Kg)	113.40
Propulsion	2 x electric motors (outboards)
Batteries	2 x 105 Ah Li NMC
Draft (m)	Up to 5.66
Platform weight (Kg)	181.44
Draft (m)	0.1524 - 0.4064

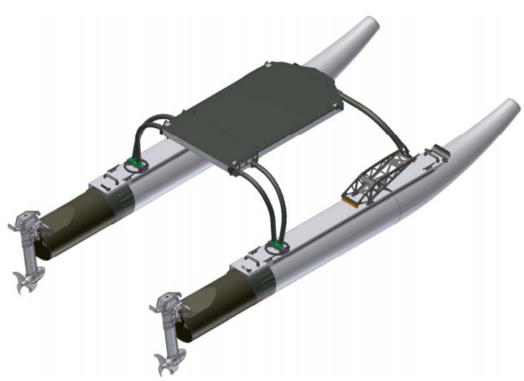

Fig. 1. Wave adaptive modular vehicle (WAM-V)

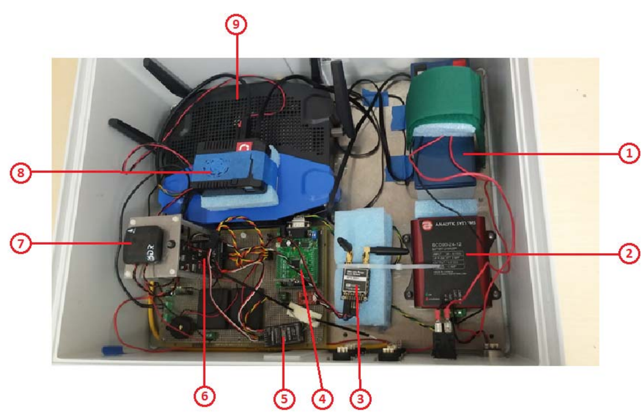

Fig. 2. Main control unit (MCU)

system.

Odroid XU-4 runs on Ubuntu, Linux operating system, and is loaded with the Dronekit-Python libraries which augment the autopilot capabilities of the Pixhawk-Ardupilot framework by executing computationally intensive tasks, such as, for example, path-planning, and computer vision-based object detection, which also require a low latency link. It basically provides a user-friendly interface to control the vehicle, configure waypoint-based autonomous missions, and manage all communications between Pixhawk and Odroid from Python scripts, using the Micro Air Vehicle Link (MAVLink) communications protocol. MAVLink is also used for communications between Pixhawk and the ground control station (GCS), exchanging bidirectional information from navigational and inertial sensors and the mission plan.

3) Ground Control Station (GCS). The GCS (Fig. 3), is referred to a particular program running a graphical user interface (GUI) where you can plan an autonomous mission and monitor the state of the vehicle by streaming data to and from Pixhawk. Thus GCS serves to monitor the health of the vehicle and provide situational awareness of its performance as well as communicate commands to the vehicle. Several options are available for GCS. One of the most widely used is Mission Planner (MP) for Windows, which also runs on a more limited version on Linux, called APM Planner. QGroundControl is another powerful GCS that can be used with either Ardupilot or PX4 autopilots. MP characteristics and functionality are used here, however, most of its capabilities can be extended to other GCS. Another important command line based GCS is Mavproxy *[18]*, which is also widely used, especially by developers.

In general, the GCS consists of the following:

- A computer where the GCS app can run, with recommended but not required internet connection.
- The Mission Planner software [19] installed on the computer.
- A telemetry transceiver [20].
- A remote control transmitter.
- An operator.

MP allows configuration of the entire system using a convenient and easy to use GUI, from that enables performance of the following actions:

- Loading the firmware (Ardupilot software) into the autopilot (Pixhawk) that controls the vehicle.
- Setup and configuration of sensors such as the compass, gyros and accelerometers, as well as the R/C transmitter.
- Modification of the autopilot parameters and control gains, which define the overall configuration and response of the vehicle.
- Planning, saving and loading of autonomous missions onto the autopilot with simple point-and-click waypoint entry using Google maps or alike.
- Downloading and analysis of mission logs created by the autopilot.
- Interfacing with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.

Appropriate telemetry hardware allows:

- Monitoring and plotting of the vehicle's status while in operation, including data from all the sensors, as well as exchanging input / output PWM values with Pixhawk.
- Recording telemetry logs that enable the user to playback a simulated version of the mission at a later time.
- Viewing and analysis of the telemetry logs.

Fig. 3. Ground Control Station (GCS) computer, running Mission Planner (MP) software, connected to a telemetry transceiver.

III. System Configuration And Results

This section describes results from a series of experiments using the WAM-V and the Pixhawk framework

A. Configuration of the Vehicle

As described in Section II, the Ardurover controller from Ardupilot was used for this application since it offered a convenient way to control the WAM-V using differential thrust with both of the outboard electric motors. This was possible due to the similarities with skid-steer controlled rovers in terms of PWM signal commands. For this, the SKID_STEERING_OUT parameter was set to "1" using MP. This setup allowed the vehicle to be controlled with a conventional single throttle RC transmitter setup, as well as control the vehicle from a custom script Python programming language along with the available Dronekit-Python libraries.

A number of tests were conducted before starting the autonomous missions, in order to fine-tune the PID controller gains, and other navigation parameters, also configurable from MP. The first set of missions involved travelling to a number of predefined waypoints at a given speed (2 to 3 m/s), while maintaining straight trajectories (no waving) and a suitable turning rate in the transition from reaching one waypoint to start heading to the next one.

B. Waypoint Based Mission

These missions were configured and executed under two different approaches. The first was setup from the GCS using the MP software to set a number of waypoints and a Home (reference) location by clicking over the map displayed in the GUI. Then, autonomous navigation was initiated, involving moving across each of the waypoints along a straight line path. The autopilot considers a successful arrival of the vehicle to a specific waypoint whenever it hits any point within a circular area around it. This area is established by setting up a particular parameter in the parameter list from MP, which defines the value of the corresponding radius.

Fig. 4 shows the MP interface at two different moments in the experiment, from which the waypoints can be set and visualized on the map. It also shows the position, heading and trajectory of the WAM-V as well as a brief summary of live sensor data. One can refer to another tab in the GUI in order to get the entire streaming of live sensor data. The figure also shows the circular threshold area around each waypoint denoting a successful arrival to that particular location. The top left image overlayed in both snapshots illustrates the footage captured at the same instance of time in the experiment. The yellow squares depict the location of the vehicle while the red circles highlight the location of the nearby buoys.

In the second approach, the mission was configured and executed from a Python script using the Dronekit library. This library provides an entire set of functions in order to either retrieve data from Pixhawk (such as sensor data, or values from parameters) or to set up the same parameters list as done before using MP. It also provides a set of commands in order to move the vehicle, to a certain waypoint, at a desired speed. Unfortunately, Ardurover is not as versatile as Arducopter, where it is possible to control the motion of the vehicle using a velocity controller and a set of waypoints in north-east-down (NED) coordinates, enabling a smooth travel, especially when the vehicle is following a moving target, when there is a high rate of updates. Arducopter also allows controlling the yaw on a UAV. The difference can be attributed to the limited control capability of the degrees of freedom of a rover and its dependency on whether the vehicle is under-actuated or over-actuated, which is not the case for UAVs, where all degrees of freedom of the vehicle can be controlled in a standard way. The mission however can still be displayed and monitored using MP, once the waypoints have been loaded either manually or from a file to MP.

Fig. 4. Waypoint mission configuration and monitoring using MP. Live footage overlayed in the top-left corner.

C. Vision-Based Target Following Mission

The goal of the mission was for the WAM-V to track and follow a target boat autonomously, by using a stereo vision-based system and Pixhawk as the low-level controller for the motion of the vehicle. The stereo vision system performed two main tasks: the first was to detect and track the target boat even if its configuration changed over time from its initial appearance due to relative translations and rotations between the vehicles. The second task was to provide the system with information regarding the position, speed and heading of the target boat, from visual data alone (not relying on any other sensor onboard the target boat). Once the position of the target boat was computed from the stereo vision system, a point located 3 m behind it (to avoid collision with the vehicle) was then sent to Pixhawk as a waypoint update.

Fig. 5 depicts the selected configuration of the USV for the vision-based target following autonomous mission. The components are enumerated as follows: (1) electric outboard motors, (2) system batteries, (3) PCU, (4) vision box, (5) MCU, (6) stereo vision camera. The vision box encloses the computer running all the computer vision related tasks.

The stereo vision system functionality was developed using a combined approach based on the implementation of the Tracking-Learning-Detection (TLD) [21] algorithm for the object detection task, and by incorporating an extended Kalman filter (EKF) to process the raw stereo measurements regarding the position of the target boat, using suitable models for the motion of the vehicle (target boat) and measurement

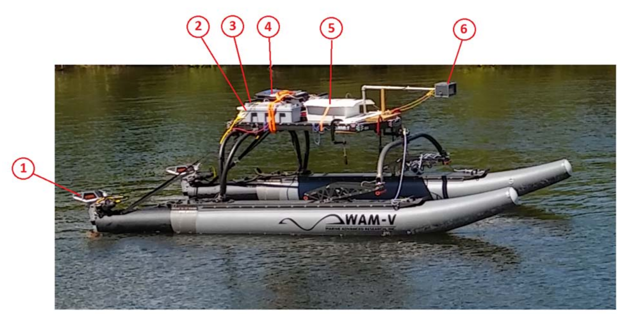

Fig. 5. USV configuration for vision-based autonomous mission

(stereo camera equations), in order to improve the tracking accuracy of the vehicle in the physical world [22]. By coupling the EKF with the object detector and the raw stereo vision data, the algorithm was able to accurately estimate not only the position of the target boat, but also its speed and heading. This makes it ideal for future incorporation of a path-planner which considers both, static and dynamic obstacles.

Fig. 6 shows the effect of combining the use of the EKF with the object-detector and the raw stereo measurements. As can seen, the filtered data (blue) closely matches the reference values (green) acquired from sensors onboard the target boat (acquired here only for data-comparison purposes) for all the variables defining the state of the vehicle, such as relative position (between the vehicles), speed and heading (of the target boat). The plots show the improvement in estimates of the state of the target boat (blue) in using EKF over estimates computed solely using raw visual data (red).

Fig. 7 illustrates in a sequence of snapshots the response of the TLD object-detector in an actual vision-based mission. One can see that the algorithm is able to detect and track the target boat while the USV is following it, despite changes in the perceived appearance of the vehicle over time. The algorithm however fails to keep track of the target boat after some time (as indicated by the yellow square), especially when abrupt changes in appearance occur (as was the case), or when either glare or poor illumination conditions are present. The effect of a sudden change in appearance is depicted in the last snapshot of Fig. 7. Work underway will aim to improve performance under these circumstances, including complementary capabilities of other ranging sensors to be included in the perception system of the USV.

Fig. 7. TLD performing over and actual vision-based autonomous mission.

Fig. 8 depicts the data-flow among all the different processes and computers involved in the system's network. The process can be described beginning with Pixhawk sending all the sensor data to the Odroid companion computer that runs the mission script. These data are in turn sent to the vision-computer, which then fuses the information along with the position of the target boat relative to the USV (acquired with the Bumblebee2 stereo vision camera [23]) in the EKF, in order to minimize the tracking errors, resulting in better estimates regarding the state of the target boat. Once the absolute position of the target boat has been estimated (in the vision-computer which runs vision tasks and the EKF), the information is sent back to Odroid, which interprets it as a waypoint that is subsequently sent to Pixhawk to enable execution of the corresponding commands to the propulsion system. Odroid communicates with Pixhawk using the MAVLink protocol via the Dronekit libraries, while Pixhawk streams data through the telemetry transceiver to the GCS, using the same protocol. Odroid and the vision-computer both communicate using the Lightweight Communications and Marshalling (LCM) protocol.

D. *UAV Autonomous Mission*

One of the main objectives of this mission is to expand the perception capabilities of the USV, aided by a multi-domain approach, in this case provided by a UAV. This will improve the effectiveness of such autonomous mission as in rescue and surveillance operations, as well as any other task that require a high level of logistics and planning. At this stage, the immediate objective of the mission consists of the UAV taking off from the center tray of the USV, flying to a specific target location (GPS-waypoint), circling around it for a given duration, and finally approaching a waypoint near to the USV position, where it initiates vision-based landing onto the center tray, in spite of any change in the position of the USV, even if it is in motion, as schematically depicted in Fig. 9. The mission has been developed for the most part [24], but only has been tested

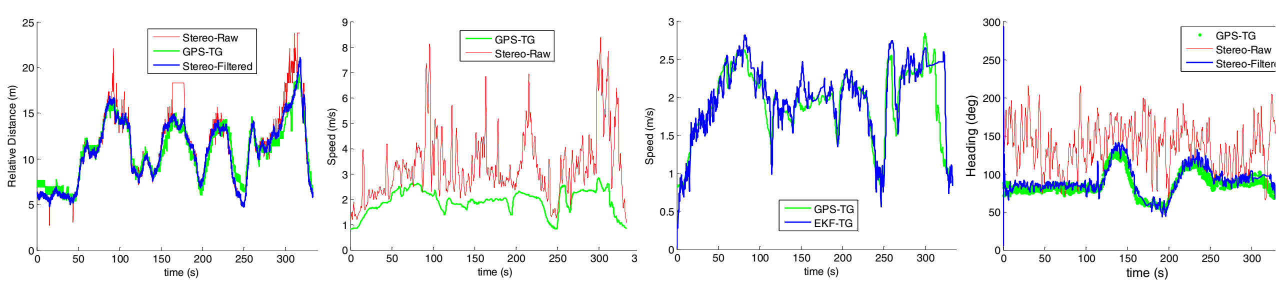

Fig. 6. Field-test data, comparing the reference values (green) with the filtered (blue) and raw values (red). From left to right: relative position between both vehicles, raw speed, filtered speed and heading.

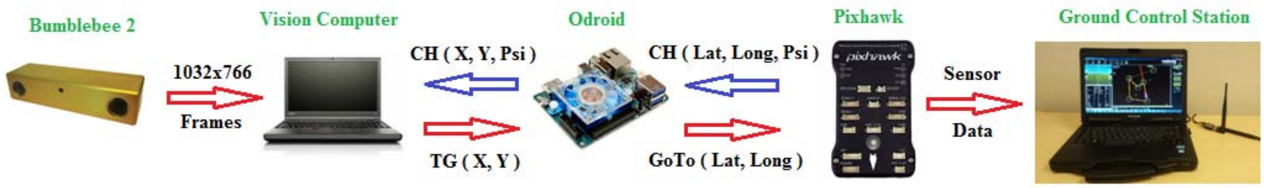

Fig. 8. Data-flow for the vision-based navigation system configuration.

in a safe ground open area. Once a receptacle has been developed for launch and recovery of the UAV on the center tray of the USV, and the performance of the vision-based landing has been improved, full at-sea trials will be carried out.

Fig. 10 shows graphically the result of an actual test mimicking the desired behavior of the autonomous mission. Every different color defines a trajectory segment from one waypoint to the next one. This picture was generated using the data flash log file onboard the microSD card on Pixhawk, which used along a MP command produces a ***.kmz** file depicting the mission trajectory which then can be visualize using Google Earth.

The UAV is the 3DR' X8+ octacopter [25], which consists of four structural arms, each with a double propulsion system. The flight controller also uses Pixhawk, with it running Arducopter. It also has the same capabilities as discussed for the Ardurover implementation since it is part of the Pixhawk / Ardupilot framework. The autonomous mission runs in an Odroid-U3 companion computer using Python programming language and the Dronekit libraries. The UAV has a downward-looking onboard camera that is activated by a computer-vision function in the program that is triggered when the UAV reaches the final GPS-waypoint, roughly located 5 m above the landing spot, in order to start the vision-based landing operation. The purpose of the vision-based landing algorithm is to overcome the intrinsic inaccuracies of a GPS-based landing operation, by detecting a circular red spot (the landing spot) on the USV and attempting to land at its center.

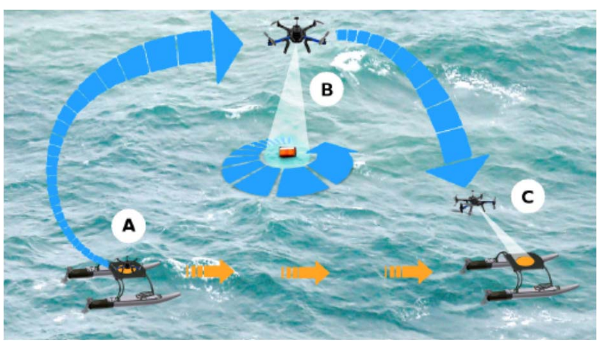

Fig. 9. UAV autonomous mission goal

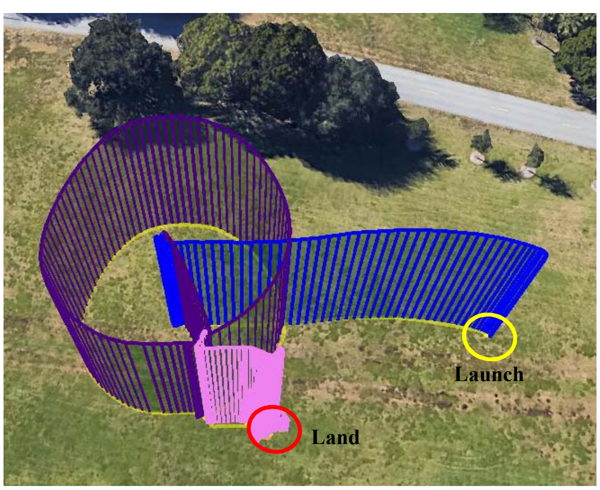

Fig. 10. Actual UAV autonomous mission test

E. Automated Launch and Recovery Operations for an AUV

A proposed design to address the mechanism and operation of an autonomous launch and recovery system (LARS) for an AUV (REMUS 100), using a USV as a host, has been developed and successfully demonstrated [26]. A novel approach where the AUV is autonomously launched and recovered using a taut line attached in the upper end to the LARS (onboard the USV), and the lower end to a depressor wing that produces negative lift (towards the seafloor) is considered. The negative lift produces a desired level of tension and angular displacement on the taut line to enable the coupling action between the AUV and the LARS, as depicted in Fig. 11.

The effort involved development and implementation of the requisite hardware, establishing an appropriate communication protocol during relative acoustic positioning between the AUV and the USV, development and implementation of a low-level controller to facilitate a constant speed and heading during launch and recovery actions, and setting up a navigation algorithm to facilitate being at particular locations during different stages of the operation along the trajectory of the USV. The launch and recovery (L&R) operation was modeled using OrcaFlex, which enables dynamic analysis and simulation of off-

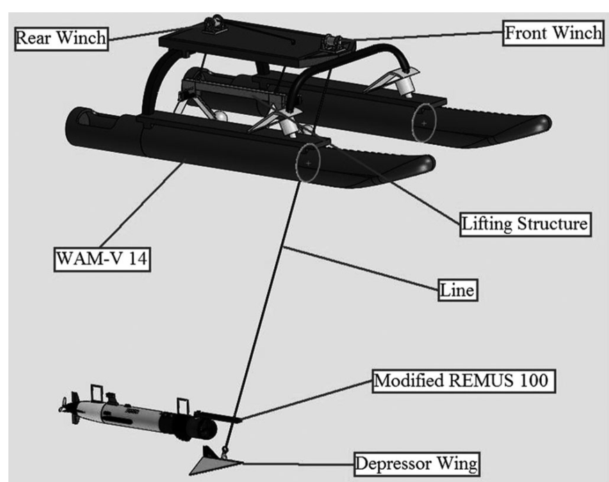

Fig. 11. Schematics of the LARS

shore marine systems [27]. The model includes LARS components and both vehicles, simulated under various conditions in order to monitor the response of the parameters involved, including position, orientation, and velocity of the vehicles as well as changes in tension of the taut line during the operation. The implementation of LARS enables deployment of an AUV in remote locations from onboard the USV for conducting local autonomous survey missions while expanding the possibilities of multi-domain autonomy, providing under-water information of the world.

IV. Ongoing Research And Further Work

Work is underway in the following topics in support of meeting the objectives in enhancing the USV capabilities.

A. Low-level control system

The current low-level Pixhawk controller is being refined through development of controllers involving analysis of vehicle motion characteristics data obtained during traversing over straight line, zigzag and circular pattern paths and physics-based models that include CFD ship-wave interaction studies underway.

B. Perception System

Observations from suites of perception sensors on the vehicles are collated and utilized to provide comprehensive information of the external world in each of the three operational domains: water surface, air and eventually, under-water, in a later stage, when the AUV is considered. A suite of complementary sensors for perception is being implemented on the USV in support of localization, mapping, obstacle avoidance, object classification and scene segmentation, target tracking and general mission-based intelligent decision-making. The sensors include RADAR, LiDAR and a custom-made stereovision system.

The RADAR is reliable under most adverse environments including heavy rain, splashing sea waves, dust and extreme weather conditions for mid-range perception. It is suitable for mapping the surroundings as well as capturing dynamics of moving objects, including orientation and speed. The LiDAR provides greater ranging accuracy and resolution to mid-range perception; it is particularly suitable for 3-D mapping of the surroundings in cluttered environments. The stereo vision sensor provides a dense cloud of ranging data as well as color and texture information, which makes it ideal for object detection and feature point extraction (convenient when performing SLAM) tasks among others; The ranging distance capacity achieved depends on the image resolution and the distance between the cameras on the stereo rig; a balanced approach between sufficient image resolution and associated computational cost is required in determining the ranging capability.

C. World Modeling

The aim of the proposed effort is to have a decentralized distributed system in which each vehicle (in a cooperative autonomous mission involving heterogeneous unmanned vehicles) has a complete and redundant world model, built through perception of its surroundings, exchange of broadcasted messages among the vehicles, and available a priori information of the region and local conditions; the model would be independently maintained and constantly updated by each vehicle. The advantages of such a decentralized system are that it is resilient with no central point of failure, allowing the vehicles to operate in isolation if necessary, and the vehicles do not need to constantly query a central information source. This reduces communication bandwidth requirements, and gives the vehicles more independence, and consequently more autonomy (Carlone, et al. 2010).

Development and maintenance of a constantly updated world map of an unknown environment requires that both vehicle position estimation and mapping problems be solved concurrently. This is typically accomplished using simultaneous localization and mapping (SLAM). Several algorithms, based on stochastic approaches, are available for solving SLAM, including use of particle and extended Kalman filters. The performance of SLAM in providing the best estimate for a world map and the state of the USV relies heavily on the accuracy associated with the ranging sensors as well as an accurate model of the motion of the vehicle (USV), which considers all forces involved, including hydrodynamic forces due to waves and currents, wind and the thrust of the actuators themselves.

D. High-level Planning

Once a model of the world has been constructed, and a confident belief of the position of the vehicle has been estimated, the USV plans a path across the environment. The main objective is to avoid any collision with potential obstacles, while considering typical navigation regulations, such as COLREGS, as well as the priorities associated with a particular task within the context of cooperative multi-domain autonomous missions. The most suitable path-planning techniques are being considered in order to develop a custom high-level planner that provides safe navigation of the USV taking account of various aspects of the operational environment.

E. Endurance Enhancement

Work is in progress in installing a Honda EU2000i gas generator onboard the USV, in support of increasing vehicle endurance and range. The USV uses two batteries in parallel. One charger is connected to each battery. The chargers will be connected to the generator when the latter is integrated into the hybrid system. The chargers will constantly charge the batteries whenever the generator runs. An electrical starting system is being installed into a portable Honda EU2000i gas generator to facilitate automated operations.

V. Conclusion

USV-based surface autonomy involving a UAV and AUV is considered. Capabilities for launch and recovery of the UAV and AUV from a WAM-V USV are being developed. The USV is a catamaran with a stable center tray that serves as a platform for various instruments. Vision-based navigation has been successfully developed on the USV and implementation of RADAR and LiDAR capabilities is underway. Use of a common open-source Pixhawk / Ardupilot framework for both surface and aerial vehicles enables an accelerated approach in developing a significant level of autonomy, including world modeling reconstruction, avoidance of of static and dynamic obstacles (such as other moving boats), while characterizing the type of obstacles the USV encounters in the environment (boats, shoreline, etc). The multi-vehicle autonomous approach considered expands the USV's perception of the world in a way that will enable operations over extended regions at remote sites under harsh conditions. The USV, UAV and AUV triad can constitute an autonomous unit, and the goal would be to expand to a number of such units, cooperating with each other, in order to cover larger areas in less time.

ACKNOWLEDGMENT

The work is supported by the Office of Naval Research under Grants N000141512724 (Program Manager: Kelly Cooper).

REFERENCES

[1] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research,* vol. 25, no. 5-6, pp. 403-429, 2006.

[2] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard and D. Nardi, "Fast and accurate SLAM with Rao–Blackwellized particle filters," *Robotics and Autonomous Systems,* vol. 55, no. 1, pp. 30-38, 2007.

[3] L. Andersson and J. Nygards, "C-SAM: Multi-robot SLAM using square root information smoothing," in *IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, USA, 2008.

[4] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The Int. Journal of Robotics Research,* vol. 25, no. 12, p. 1243–1256, 2006.

[5] X. S. Zhou and S. Roumeliotis, "Multi-robot slam with unknown initial correspondence: The robot rendesvous case," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1785-1792, 2006.

[6] L. Carlone, M. Ng, J. Du, B. Bona and M. Indri, "Rao-blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication," in *IEEE Int. Conf. on Robotics and Automation*, Anchorage, Alaska, USA, 2010.

[7] A. Cunningham, V. Indelman and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2013.

[8] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos and G. Grisetti, "Multi-robot SLAM using condensed measurements," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013..

[9] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numer. Math,* vol. 1, pp. 269-271, 1959.

[10] 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 SSC4,* vol. 4, no. 2, pp. 100-107, 1968.

[11] L. Kavraki, P. Svestka, J. Latombe and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation,* vol. 12, no. 4, pp. 566-580, 1996.

[12] S. La Valle and J. Kuffner, "Randomized kinodynamic planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[13] S. La Valle and J. Kuffner, "Rapidly-exploring Random Trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2001, pp. 293-308.

[14] I. R. Bertaska, "Intelligent supervisory switching control of unmanned surface vehicles (Doctoral Dissertation)," ProQuest, Boca Raton, FL, 2016.

[15] WAM-V Marine Advanced Research, Inc., WAM-V Marine Advanced Research, Inc., 06 May 2015. [Online]. Available: http://www.wam-v.com/.

[16] px4 autopilot, "px4 autopilot," 3D Robotics, 2016. [Online]. Available: https://pixhawk.org/.

[17] ArduPilot, "ArduPilot," ArduPilot, 2016. [Online]. Available: http://ardupilot.org/.

[18] A. Tridgell, "MAVProxy," [Online]. Available: http://ardupilot.github.io/MAVProxy/html/index.html.

[19] M. Oborne, "Installing Mission Planner," Ardupilot, [Online]. Available: http://ardupilot.org/planner/docs/common-install-mission-planner.html.

[20] ArduPilot Dev Team, "Telemetry (landing page)," ArduPilot Dev Team, 2016. [Online]. Available: http://ardupilot.org/copter/docs/common-telemetry-landingpage.html#common-telemetry-landingpage.

[21] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 34, no. 7, pp. 1409-1422, 2012.

[22] A. J. Sinisterra, M. R. Dhanak and K. von Ellenrieder, "Stereovision-Based Target Tracking System for USV Operations," *Ocean Engineering,* vol. 133, pp. 197-214, 2017.

[23] FLIR, "Bumblebee2 1394a," FLIR Integrated Imaging Solutions, Inc, 2017. [Online].

[24] M. Bohnenkamp, "Development of an Autonomous Vision-Based Landing System for an UAV to Ensure Highly Accurate Landing on an USV," B.Sc thesis, Gelsenkirchen, 2015.

[25] 3DR Robotics, "X8+ Operation Manual," 2014. [Online]. Available: https://3dr.com/wp-content/uploads/2014/11/X8+-Operation-Manual-vA.pdf.

[26] E. Sarda and M. Dhanak, "A USV-Based automated launch and recovery system for AUVs," *IEEE Journal of Oceanic Engineering,* vol. 42, no. 1, pp. 37-55, 2017.

[27] M. Wu, J. -F. Saint-Marcoux, R. D. Blevins and P. Quiggin, "Innovative 3-D implementation of riser wake interface assessment," in *Proc. Int. Soc. Offshore Polar Eng.*, 2008.