Autonomous Vehicles for Outdoor Multidomain Mapping

Leonardo Garberoglio*[1], Patricio Moreno†‡[2], Ignacio Mas§[4] and Juan I. Giribet†‡[3]

*Grupo de Estudio de Sistemas de Control, Facultad Regional San Nicolás,Universidad Tecnológica Nacional
San Nicolás, Buenos Aires, Argentina
[1]lgarberoglio@frsn.utn.edu.ar
†GPSIC – Facultad de Ingeniería, Universidad de Buenos Aires
Ciudad Autónoma de Buenos Aires, Argentina
‡Instituto Argentino de Matemática – CONICET
Ciudad Autónoma de Buenos Aires, Argentina
[2]pamoreno@fi.uba.ar
[3]jgiribet@fi.uba.ar
§Instituto Tecnológico de Buenos Aires – CONICET
Ciudad Autónoma de Buenos Aires, Argentina
[4]imas@itba.edu.ar

***Abstract*—In the last years, progress has been made attempting to replace a unique, complex and expensive vehicle equipped with several sensors such as LIDAR, RGB cameras, thermal sensor, etc. with a group of small vehicles, each of them carrying one sensor. There are several advantages of these segmented architectures, for instance this allows a reduction in the cost of the vehicles (several small vehicles can be less expensive than one big vehicle), the flexibility to choose for a mission only those vehicles with the appropriate sensors, the robustness of the system since it can acquire information even if one vehicle fails, among others. The advantage of segmented architectures is even more noticeable if the vehicles carrying those different sensors, have different characteristics or environments for operations, e.g. aerial, terrestrial or aquatic vehicles. In this work, we present the experimental results obtained with an ASV (Autonomous Surface Vehicle) and a UAV (Unmanned Aerial Vehicle) that cooperate to obtain a topographic survey of the terrain. The ASV is equipped with a LIDAR, meanwhile the UAV is equipped with a monocular RGB camera. The data acquired is post-processed in order to obtain a detailed map of the coastline of a creek and the surrounding area.**

***Resumen*— En los últimos años se ha avanzado en el reemplazo de un único vehículo, complejo y de elevado costo, dotado de una variedad de sensores—LIDAR, cámaras RGB, sensores térmicos, etc.—por un grupo de vehículos pequeños, cada uno de ellos cargando un sensor. Se pueden ver varias ventajas en estas arquitecturas segmentadas, entre otras, la reducción en el costo de los vehículos (un grupo de pequeños vehículos puede ser más económico que un único vehículo de mayor tamaño), la flexibilidad de elegir para una misión únicamente los vehículos con los sensores apropiados, una mayor robustez en la arquitectura ya que si un vehículo falla no necesariamente lo haga el sistema. La ventaja en el uso de arquitecturas segmentadas es aún más notable cuando los vehículos que portan los diferentes sensores tienen características o entornos de operación diferentes, por ejmplo, vehículos aéreos, terrestes o acuáticos. En este trabajo, presentamos los resultados experimentales obtenidos con un vehículo autónomo de superficie (ASV, por sus siglas en inglés) y un vehículo aéreo no tripulado (UAV, por sus siglas en inglés) que cooperan para obtener un mapa topográfico del terreno. El ASV es equipado con un LIDAR, mientras que el UAV es equipado con una cámara RGB monocular. Los datos adquiridos son posprocesados para obtener un mapa detallado de la línea costera de un arroyo y la zona lindante.**

I. Introduction

Autonomous surface vehicles (ASV) are drawing attention for several applications, such as water sampling [1], oil skimming [2] and surveillance [3]. The deployment of a big unit, with a large amount of sensors, was historically the preferred choice. Advances on electronic sensors, miniaturization of electric motors, specially brushless motors, and advances on lithium-ion batteries, allowed designers to think of small vehicles as a feasible option. Although small vehicles can carry just a few sensors, they are cheaper and simpler to design, build and deploy, making them an interesting platform, specially for shallow water operations, see for instance [4] and [5].

The coordination of several ASV to perform a common task has been of great interest in the last years, and different control strategies have been proposed to address this problem. In [6] a cluster space control strategy was successfully used to coordinate a fleet of ASV. In [7] a null-space behavioral approach on ASV was demonstrated. Experimental results of coordination of underwater and surface vessel have been reported in [3] and [8].

In this work, the design, construction and deployment of an ASV is presented. The main purpose of this prototype is to serve as a take-off and landing platform for an unmanned aerial vehicle (UAV). These are the first steps for a research project that began in December of 2017, focused on developing a system consisting of UAV and ASV that cooperate to solve a given task. The cost was the main driver for developing the prototype and compromises had to be made, limiting the applicability of the prototype for real applications. However, in calm waters it has been proved to be useful for some applications. In this work, we present experimental results obtained with an ASV and UAV that cooperate to obtain maps of the terrain. Each vehicle is equipped with different sensors which provide data to obtain a detailed map of the coast and surrounding area.

Working with multi-domain vehicles allows users to achieve complex tasks thanks to the flexibility of the system. However, the control strategy has to take into account the constraints and dynamics of each vehicle. Preliminary developments of the coordinated motion control of an ASV and a UAV are also presented here, showing computer simulation results of trajectory following tasks.

II. ASV DESIGN AND COASTAL MAPPING

In this section, a description of the developed autonomous surface vehicle, named Yaguaron ASV, is presented together with its use in a mission performing coastal mapping.

A. Autonomous Surface Vehicle

The design criteria of the Yaguaron ASV are:

- Low cost.
- Catamaran shape, with two hulls, allowing high payload capability and good stability.
- Differential-drive thruster configuration.
- Commercially-available and low cost on-board computer.
- Electric drive propulsion.

The Yaguaron hulls are made of PVC pipe and fiber glass. The hulls are $0.25\,\mathrm{m}$ in diameter and $1.2\,\mathrm{m}$ long, resulting in a volume of $58\,\mathrm{dm}^3$ for each one. To seal the hulls, two fiberglass pieces were molded. The bow is molded with a bullet-shape custom-made wood mold. The stern is made with a flat molded fiberglass piece. Four screws are fastened from the inside to the outside of each pipe and two screws on the stern piece. Both fiberglass pieces are attached to the main PVC pipe using epoxy resin and fiberglass. After the buoyancy test, putty is applied to fill surface imperfections. Finally, after sanding the hull, a UV protective paint is applied.

The chassis is made of $20\,\mathrm{mm} \times 20\,\mathrm{mm}$ extruded aluminum profile. Two $1.2\,\mathrm{m}$ pieces are fastened on each hull using pre-mounted screws. Three $900\,\mathrm{mm}$ pieces are used as a bridge. On the stern there is a vertical frame, which is made of $25\,\mathrm{mm} \times 25\,\mathrm{mm} \times 5\,\mathrm{mm}$ L-shape aluminum profile. This frame supports electronic devices and sensors. The horizontal frame will serve as the ASV take-off and landing platform. Fig. 1 shows an image of the Yaguaron ASV during operation.

The propulsion consists of two BlueRobotics T100 thrusters. They are mounted on each hull providing a differential-drive configuration, which eliminates the need of a rotating rudder, making the construction and control simpler. Each thruster has a pushing force of 2.5 kgf, giving the ASV a top speed of $1\,\mathrm{m/s}$. Two basic electronic speed controller (ESC) from BlueRobotics are used to drive the thrusters. ESC firmware allows backward/forward operation. The thrusters are mounted using a 3D-printed piece and an aluminum profile. For transportation, thruster can be unmounted. Fig. 2 shows a thruster and fig. 3 shows its mechanical support.

To test the ASV, three $12\,\mathrm{V}$ $7\,\mathrm{Ah}$ sealed lead-acid batteries are used. At cruise speed, Yaguaron consumes $10\,\mathrm{A}$, therefore a 2-hour mission is possible. Three more batteries could be added to extend mission time.

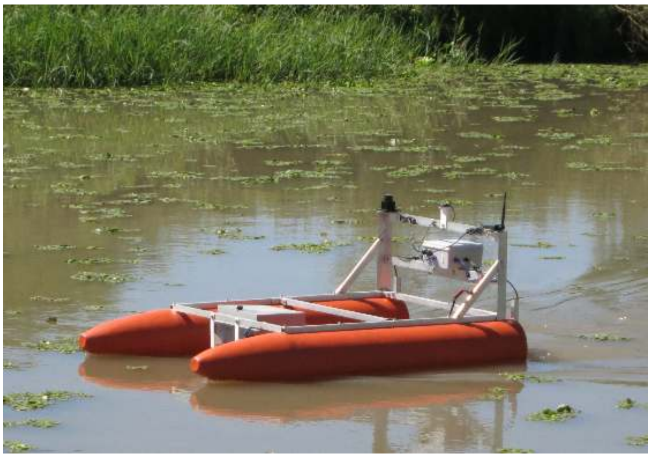


Figure 1: Yaguaron ASV.

Figure 2: BlueRobotics T100 thruster [courtesy BlueRobotics].

Yaguaron uses an off-the-shelf PixHawk Flight Controller (FC). The PixHawk is an open-hardware FC that runs a realtime operating system and autopilot software. The Pixhawk consists of a main FC, a Neo 8m GPS module, an APM power module, a 3DR $433\,\mathrm{MHz}$ radio link, a safety switch and a buzzer. All these items are shown in fig. 4. For this work, Ardurover firmware version 3.2 is flashed into the FC. A 7-channel Radio Control system is used to command the ASV. The left stick commands the throttle (Up/Down) and the right stick commands the steer (Left/Right). A two-position and a three-position switch are used to change Flight Mode (Manual, RTH, Acro, Steer, Guided, Hold and Auto). On a ground-based PC, QGroundControl—a Pixhawk-compatible control and mission planning software—is used to monitor speed, location, battery usage, trajectory and other variables. Communication is established using a 3DR $433\,\mathrm{MHz}$ radio link.

For high-level control, a Raspberry Pi 3 is used. This single-board computer runs Ubuntu Mate and the Robot Operating System (ROS). USB ports on this computer allow the connection an external Wi-Fi dongle and a LIDAR sensor. A $2.4\,\mathrm{GHz}$ Wi-Fi router on the ground provides a network connection between Raspberry Pi on the ASV and the ground-based PC. The FC and Raspberry Pi are linked together through their UART ports. The FC is configured to send/receive Mavlink messages and the Raspberry runs a MAVROS package to interface the Mavlink messages with ROS.

A Hokuyo LIDAR is connected to the Raspberry Pi through a USB port. A ROS package drives the LIDAR and publishes measurement data. This data can be used for several purposes like object avoidance, coast mapping or

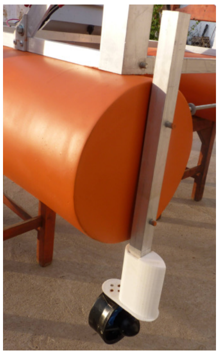


Figure 3: Thruster mounted on the ASV hull.

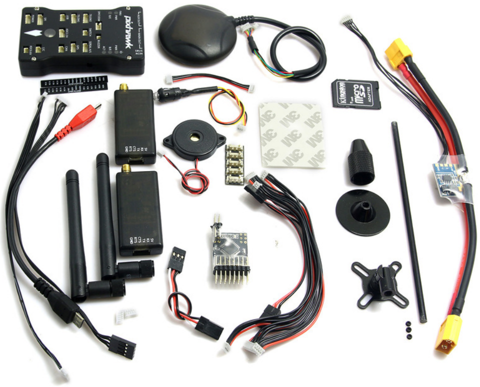


Figure 4: Pixhawk with accessories.

SLAM. The LIDAR is mounted on the vertical frame, shown in fig. 5, using magnets and a 3D-printed support.

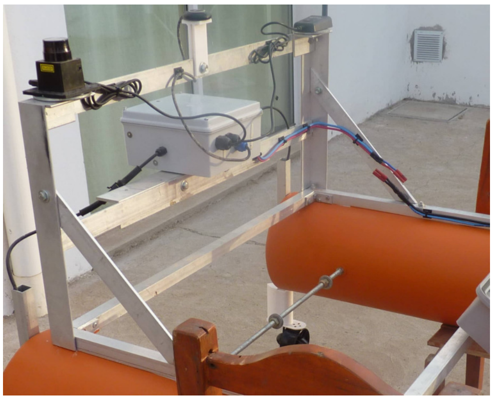
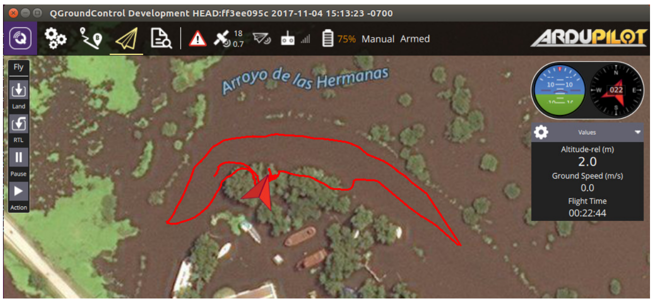


Figure 5: Hokuyo LIDAR mounted on the ASV.

B. LIDAR coast mapping experimental results

A proposed field task to verify the functionality of the Yaguaron ASV is to map a section of the coast of the creek know as "Las Hermanas", in Ramallo city, Buenos Aires province, Argentina. For this task a Hokuyo LIDAR and a ROS based system are used.

The LIDAR is mounted on the vertical frame, pointing to the right. This way the laser beam is always pointing to one coast. For the experiments presented here, the ASV was controlled by an operator, in manual mode, at low speed and close to the coast to get valid LIDAR readings. All measurements are recorded for off-line processing. A ground-based PC running QGroundControl is used to monitor the trajectory. Fig. 6 shows a screenshot of QGroundControl while carrying out the experiment.



QGroundControl Development HEAD:ff3ee095c 2017-11-04 15:13:23 -0700

75% Manual Armed

Altitude-rel (m) 2.0

Ground Speed (m/s) 0.0

Flight Time 00:22:44

Figure 6: Screenshot of trajectory used for coast mapping.

After the recording is complete, LIDAR data is processed on a desktop PC using Hector Mapping ROS package [9]. This package creates a grid map using only LIDAR measurements. In Fig. 7 the raw result of the mapping process is shown. Fig. 8 shows a section of the map overlaid on an aerial mosaic of the area. As Hector Mapping relies only on scan-matching of LIDAR measurements, errors grow over time and mapping long sections of the coast accurately is difficult.

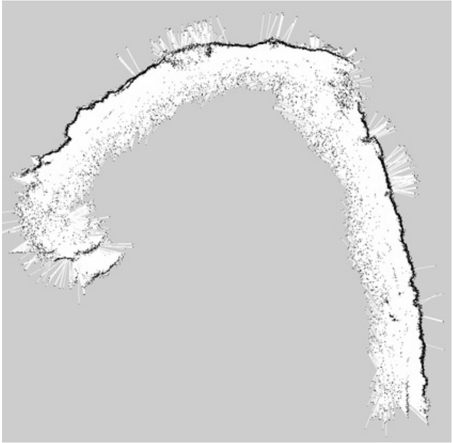


Figure 7: RAW map obtained with Hector Mapping ROS package.

Next, other mapping packages are used to compare the results. In this case, Google Cartographer is used [10]. This package can make use of additional sensor information to improve the accuracy of the results. Data from inertial measurement units (IMU) can be used to estimate the motion of the system and include such information in the generation of the map. A new set of LIDAR data is collected together with inertial measurements from the on-board FC. Fig. 9 shows the results using the Google Cartographer package with IMU information. It can be noticed the importance of adding IMU measurements to solve the SLAM[1] problem with LIDAR. Fig. 10 shows both the resulting map using the Google Cartographer package with and without the IMU information. When no IMU data is used to process LIDAR measurements the attitude drift is not compensated.

III. Aerial survey using a UAV

Another key aspect of the system under development is the capability to use an unmanned aerial vehicle to assist the ASV for navigation purposes, as well as to improve the

[1] Simultaneous Localization and Mapping

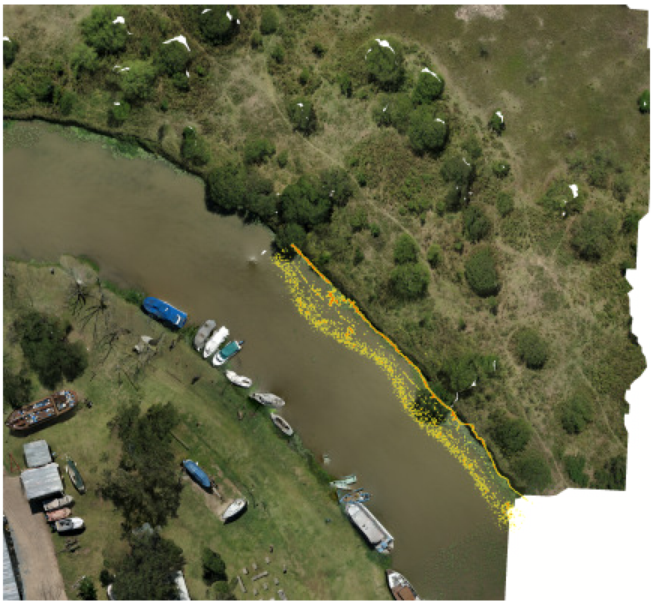


Figure 8: Map obtained with Hector Mapping ROS package overlaid on aerial mosaic.

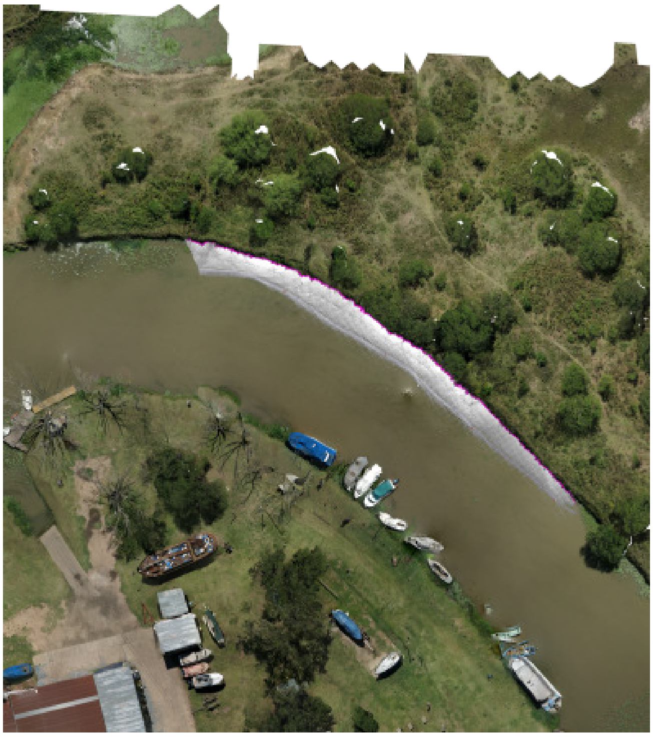


Figure 9: Map obtained with Cartographer Mapping ROS package overlaid on aerial mosaic.

quality of the information gathered based on aerial imagery collected from the UAV.

To evaluate the results of the previous section and to advance towards the objectives described above, a high resolution map of the area of interest was created from still images collected with a six-rotor multicopter with a nadir-pointing camera. The images were processed using the OpenDroneMap free software toolkit to create an orthorectified mosaic of the area. Fig. 11 shows the resulting composed mosaic overlaid on a Google Maps image.

The improvement in the resulting image resolution allows the usage of the generated map as a ground-truth for ASV-based map generation.

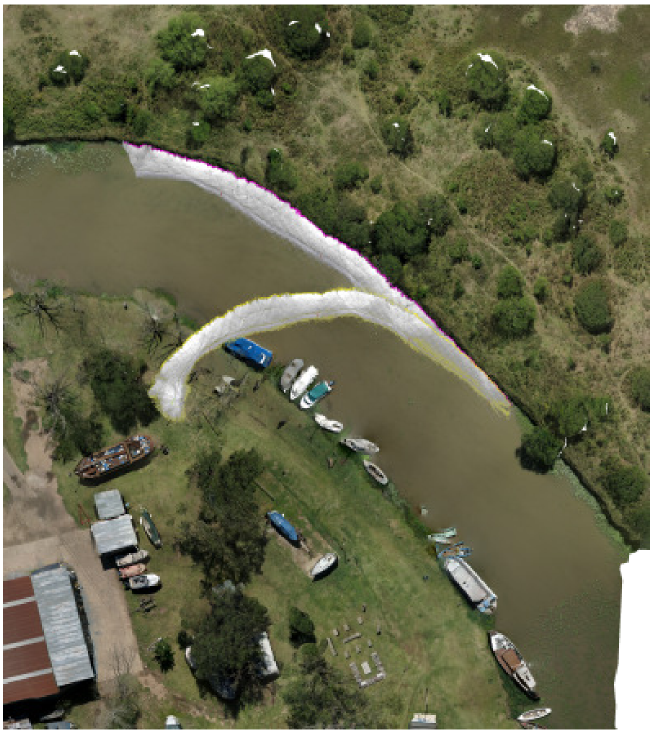


Figure 10: Map obtained with Cartographer (magenta trace using IMU data and yellow trace without using IMU data) overlaid on aerial mosaic.

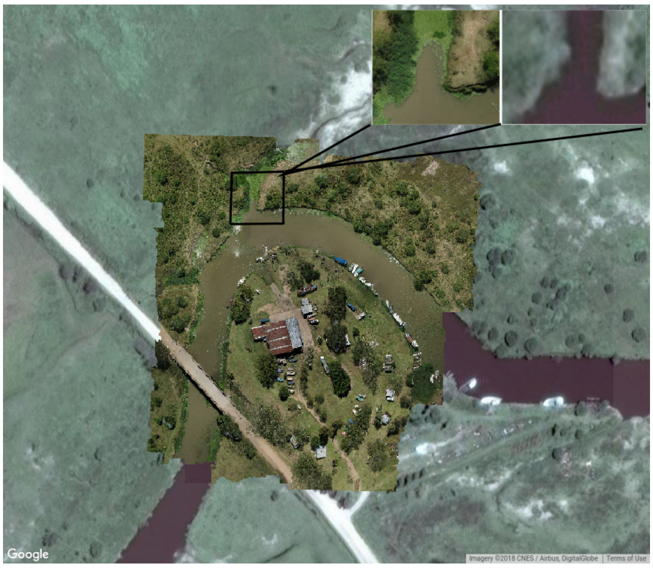


Figure 11: Mosaic composed of aerial photographs with Google Maps image in the background for resolution comparison.

IV. UAV–ASV COOPERATIVE CONTROL

The usage of a UAV and an ASV together extends the capabilities of both vehicles. For example, the UAV can charge its batteries on board the ASV and capture images from above, which enables the ASV to do better trajectory planning. Another advantage of this system is the capability of constructing more accurate maps, using the distributed sensor architecture and integrating information from heterogeneous sensors.

For the vehicles to conduct certain tasks, it may be necessary for them to perform motions in a coordinated fashion. To do so, a formation control approach must be implemented.

A. ASV-UAV formation control definition

In this section, a brief specification of a particular formation definition is given. This specification follows that of [11], and the reader is encouraged to read the details from it and references there in.

Let $\mathbf{p}^a \in \mathbb{R}^3$ represent a vehicle's position in frame a. Let $\mathbb{H}$ be the set of unit-norm quaternions with the standard operations. Each $\overline{\mathbf{q}} \in \mathbb{H}$ can be written as $\overline{\mathbf{q}} = (\mathbf{q}, q_0)$ with $\mathbf{q} \in \mathbb{R}^3$, the vector component, and $q_0 \in \mathbb{R}$ the scalar component.

In this approach, each vehicle's pose is represented using dual quaternions. Each dual quaternion holds the information of the complete pose of a vehicle, namely position and attitude. Given a unit quaternion $\overline{\mathbf{q}} \in \mathbb{H}$ (which represents the attitude) and a quaternion $\overline{\mathbf{p}} \in \mathbb{H}$ with scalar part zero (which represents the position), a dual quaternion is defined as

$$\mathbf{Q} = \overline{\mathbf{q}} + \varepsilon \frac{1}{2}(\overline{\mathbf{p}} \circ \overline{\mathbf{q}})$$

where $\circ$ is the quaternion product, and $\varepsilon \neq 0$ is a dual number which satisfies $\varepsilon^2 = 0$. The dual quaternion $\mathbf{Q}$ is a 4-tuple (the vector and scalar parts of the principal and dual components, respectively) that represents the pose (position and attitude) of a given vehicle with respect to a frame of reference.

The formation is described in a space composed of the pose of the leader –the ASV– and the pose of follower –the UAV– with respect to the leader using dual quaternions. A proportional controller operates in this space and, using the error dynamics described in [11], generates the compensations signals needed to minimize such errors. These signals then translate into velocity commands for each vehicle.

Fig. 12 shows a leader–follower formation definition without using dual quaternions. This definition is relevant, as is easier to understand than looking directly at the dual quaternion parameters. The ASV center is defined in the global frame of reference, as its yaw angle. The follower's position is represented as an azimuth (φ), elevation (ψ) and distance (ρ) from the leader, and the UAV orientation (θ) is defined from that of the leader.

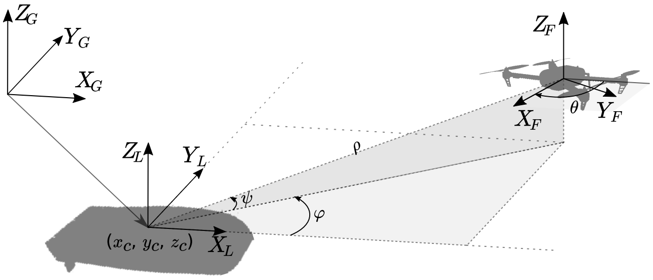


Figure 12: Leader–follower formation definition.

Given a dual quaternion representation of robot i:

$$\mathbf{Q}_i = \overline{\mathbf{q}}_i + \varepsilon \frac{1}{2}(\overline{\mathbf{p}}_i \circ \overline{\mathbf{q}}_i),$$

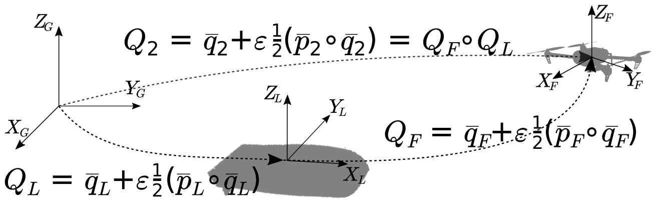
with $i \in \{1, 2\}$ if the representation is in the robot's space and $i \in \{L, F\}$ if the representation is in the formation space, the kinematic transformation from one space to another is given by:

$$\mathbf{Q}_L = \mathbf{Q}_1 \tag{1}$$

$$\mathbf{Q}_F = \mathbf{Q}_2 \circ \mathbf{Q}_1^*. \tag{2}$$

Notice that, the quaternion product $\circ$ can be extended to dual quaternions just considering that $\varepsilon^2 = 0$.

Fig. 13 shows the kinematic transformations that relate the representation in the space of the ASV and UAV poses to the representation in the leader-follower formation space.



Z_G
$Q_2 = \bar{q}_2 + \varepsilon\frac{1}{2}(\bar{p}_2 \circ \bar{q}_2) = Q_F \circ Q_L$
Z_F
Y_G
X_F
Y_F
Z_L
Y_L
X_G
$Q_F = \bar{q}_F + \varepsilon\frac{1}{2}(\bar{p}_F \circ \bar{q}_F)$
X_L
$Q_L = \bar{q}_L + \varepsilon\frac{1}{2}(\bar{p}_L \circ \bar{q}_L)$

Figure 13: Leader-follower formation relations

Let the dual quaternion errors be defined as:

$$\delta\mathbf{Q}_i \quad = \mathbf{Q}_i^* \circ \mathbf{Q}_{i_\text{d}} \quad i \in \{L, F\} \tag{3}$$

with $\mathbf{Q}_{i_\text{d}}$ the desired position and orientation of each vehicle. The following theorem (that can be proven following the ideas of [11, Theorem 1]) gives a control algorithm that makes the position and attitude error converge asymptotically to zero.

Theorem 1: For each $i = \{L, F\}$, assume that the desired dual quaternion $\mathbf{Q}_{i_d}$, angular velocity ω_{i_d} and linear velocity $\mathbf{v}_{i_d}$ are given. Suppose that $\mathbf{K}_{\omega,p_i}, \mathbf{K}_{v,p_i} \in \mathbb{R}^{3\times3}$ are strictly negative definite matrices, and let

$$\delta\mathbf{Q}_i = (\delta\mathbf{q}_i, \delta q_{0_i}, \delta\mathbf{d}_i, \delta d_{0_i})$$

be the tracking error defined in equation (3). If the control commands ($\mathbf{Q}_{i_{cmd}}$) defined for the leader and follower are given by:

$$\dot{\mathbf{Q}}_{i_{cmd}} = \frac{1}{2}\begin{bmatrix} \mathbf{S}(\mathbf{q}_{i_{cmd}}) + \mathbf{I}q_{0_{i_{cmd}}} & 0 \\ -\mathbf{q}_{i_{cmd}}^T & 0 \\ \mathbf{S}(\mathbf{d}_{i_{cmd}}) + \mathbf{I}d_{0_{i_{cmd}}} & -\mathbf{S}(\mathbf{q}_{i_{cmd}}) + \mathbf{I}q_{0_{i_{cmd}}} \\ -\mathbf{d}_{i_{cmd}}^T & -\mathbf{q}_{i_{cmd}}^T \end{bmatrix} \cdot \begin{bmatrix} \omega_{i_{cmd}} \\ \mathbf{v}_{i_{cmd}}^b \end{bmatrix}$$

$$\begin{aligned} \omega_{i_{cmd}} &= \omega_{i_d} + \operatorname{sgn}(\delta q_{0_i})(\mathbf{K}_{\omega,p_i}\delta\mathbf{q}_i), \\ \mathbf{v}_{i_{cmd}} &= \mathbf{v}_{i_d} + \mathbf{R}(\overline{\mathbf{q}}_{i_d})\left(\mathbf{K}_{v,p_i}\mathbf{R}(\overline{\mathbf{q}}_{i_d}^*)\delta\mathbf{p}_i\right), \end{aligned}$$

where $\mathbf{S} : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ is the matrix such that $S(x)y = x \times y$ (vector product) for every $x, y \in \mathbb{R}^3$, and $\mathbf{I}$ is the identity matrix. Then, for $i = \{L, F\}$, it follows that $\lim_{t\to\infty} \delta\mathbf{q}_i = 0$), and $\lim_{t\to\infty} \delta\mathbf{d}_i = 0$.

B. Simulations

In this section, simulation results of the control scheme described in previous section are shown for the proposed system.

The simulation environment was installed on a virtual machine running Xubuntu 16.04 x86_64, with 5 GB of RAM and 4 of the 8 cores of the host machine. The host machine itself was a notebook with 8 GB of RAM and an Intel's i7 core, running Debian 10. The virtual machine had the Robot Operating System (ROS)—Kinetic—and Gazebo 7.0. The UAV was modeled using a Gazebo plugin from the Autonomous System Lab of ETH Zürich University [12] for 3D-Robotics' IRIS and the PX4 Firmware, and the communication with the model was done with a MAVROS interface. The model used for the world was developed by the Laboratório de Sistemas Autônomos – PUCRS [13]. The ASV model, from the same authors, consisted of a differential-drive boat with two thrusters at the stern, commanded with forward speed, ν, and angular velocity around the z axis, ω.

The controller was implemented in MATLAB/Simulink and connected to the ROS environment through the MATLAB Robotics System Toolbox. As the ASV's motion is subject to non-holonomic constraints, due to the differential drive, its orientation is a function of the commanded motion. As the controller does not take into account the non-holonomic restrictions of the ASV's motion, a heading control was implemented to command the ASV. The forward velocity is calculated as $\nu = \sqrt{v_x^2 + v_y^2}$, while the angular speed is computed as $\omega = \operatorname{atan2}(v_x, v_y)$. Throughout the simulations, the controller gains are kept constant at $\mathbf{K}_{\omega,p_i} = \mathbf{K}_{v,p_i} = 0.6\,\mathbf{I}$, where $\mathbf{I}$ is the identity matrix in $\mathbb{R}^{3\times3}$.

Fig. 14 shows the models used in simulation within the simulation environment.

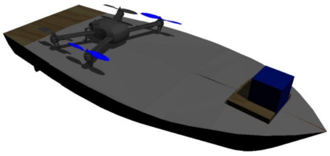
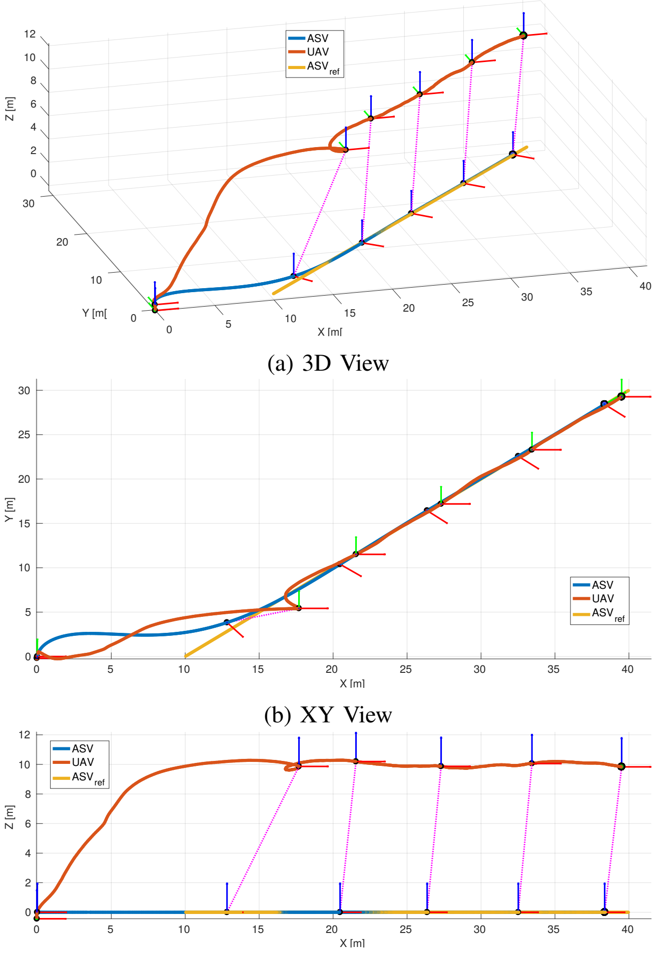


Figure 14: Surface vessel and UAV used in simulations.

1) First simulation scenario: The first simulation consists on the ASV—formation leader—following a rectilinear trajectory with the UAV flying at a constant pose relative to the leader. Following the definitions in Fig. 12, the UAV flies keeping a constant azimuth of $0\,\mathrm{rad}$, an elevation of $\pi/2\,\mathrm{rad}$, a distance of $10\,\mathrm{m}$ and a constant yaw angle of $0\,\mathrm{rad}$.

It can be seen in Fig. 15 that the formation follows the trajectory correctly. Fig. 16 shows the position error of the UAV and the ASV during the simulated experiment. As in [11], the system tracks the reference with a steady state error, as is expected from a pure proportional controller.



(a) 3D View

(b) XY View

(c) XZ View

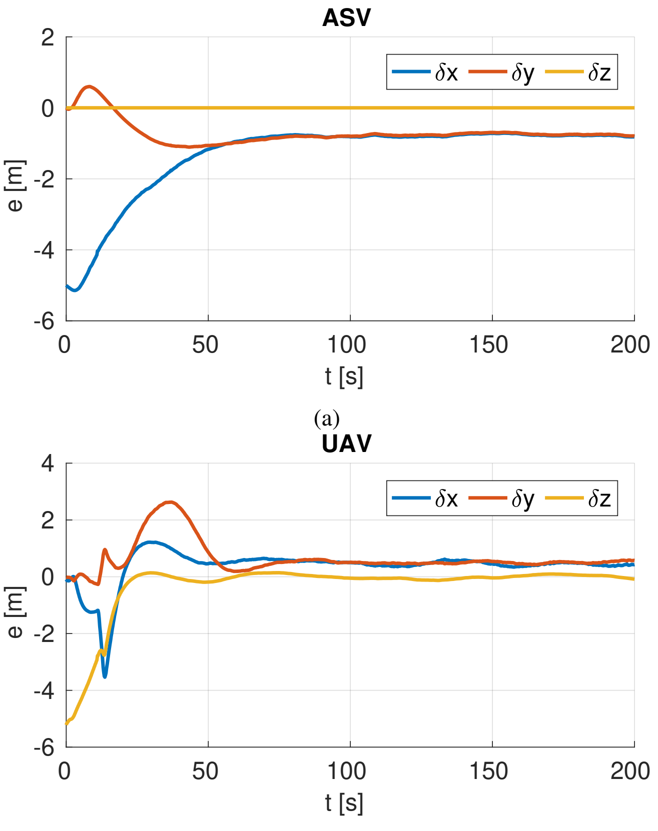
Figure 15: Simulation results of the ASV following a rectilinear trajectory and the UAV flying above it.

2) Second simulation scenario: The next simulation consists on the ASV following a rectangular trajectory with rounded corners with the UAV flying right above the leader. The polygonal trajectory has 2 sides of $14\,\mathrm{m}$, 2 sides of $19\,\mathrm{m}$ and each corner has a turn radius of $2\,\mathrm{m}$. As defined in Fig. 12, the UAV flies keeping a constant azimuth of $0\,\mathrm{rad}$, an elevation of $\pi/2\,\mathrm{rad}$, a distance of $10\,\mathrm{m}$ and a constant yaw angle of $0\,\mathrm{rad}$.

Again, Fig. 18 shows that the formation follows the trajectory. Comparing the errors shown in Fig. 19 with the ones from Fig. 16, it can be seen that the ASV is lagging as the reference is faster than ASV, while the UAV errors are smaller as it goes faster than ASV. This effect can also be noted in Fig. 18a, by watching the dotted lines that match the position of the UAV with the corresponding position of the ASV at a same time instant.

V. Conclusion

This article described the design and construction of an ASV, and some field of applications were discussed. One of them, coastal mapping, was studied and results using different techniques were presented and contrasted using aerial imagery collected by a UAV. Future work, with cooperation between the vehicles, will integrate information from both platforms to improve the quality of the results. In this sense, a formation control method for motion coordination was briefly presented and results on a simulation environment illustrated basic functionality.



(a)

(b)

Figure 16: Simulation results of the ASV following a rectilinear trajectory and the UAV flying above it (corresponding to fig. 15). Position error for each vehicle: $(\delta x, \delta y, \delta x) = \delta \mathbf{p} = \mathbf{p} - \mathbf{p}_d$.

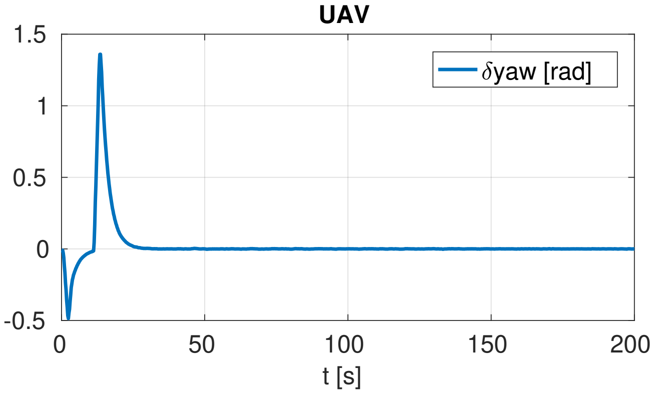
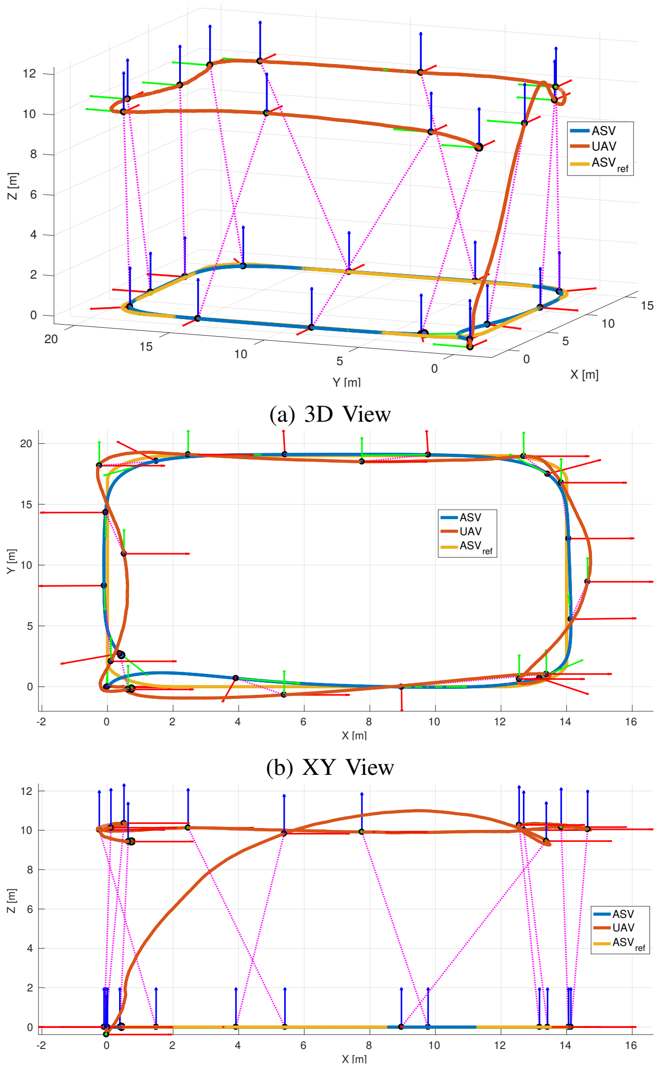


Figure 17: Simulation results of the ASV following a rectilinear trajectory and the UAV flying above it. Yaw error of the UAV.

Acknowledgment

This work has been sponsored through the Agencia Nacional de Promoción Científica y Tecnológica, FONCYT PICT 2016-2016 (Argentina) and UTN - Facultad Regional San Nicolás funds.

The authors would like to thank Ramos Victoria, García Rodrigo, Escalante Francisco, Meraviglia Marcos, Polverigiani Jesús, Riolfo Martín for their assistance during the construction of the ASV and to Nicolás Pontoriero for his support during the experimental results.



(a) 3D View

(b) XY View

(c) XZ View

Figure 18: Simulation results of the ASV following a squared (with rounded corners) trajectory and the UAV flying above it.

References

[1] M. Dunbabin, A. Grinham, and J. Udy, "An autonomous surface vehicle for water quality monitoring," in *Australasian Conference on Robotics and Automation (ACRA 2009)*, S. Scheding, Ed. Sydney, Australia: Australian Robotics and Automation Association, 2009, pp. 1–6.

[2] S. Bhattacharya, H. Heidarsson, G. S. Sukhatme, and V. Kumar, "Cooperative control of autonomous surface vehicles for oil skimming and cleanup," in *Robotics and automation (ICRA), 2011 IEEE international conference on.* IEEE, 2011, pp. 2374–2379.

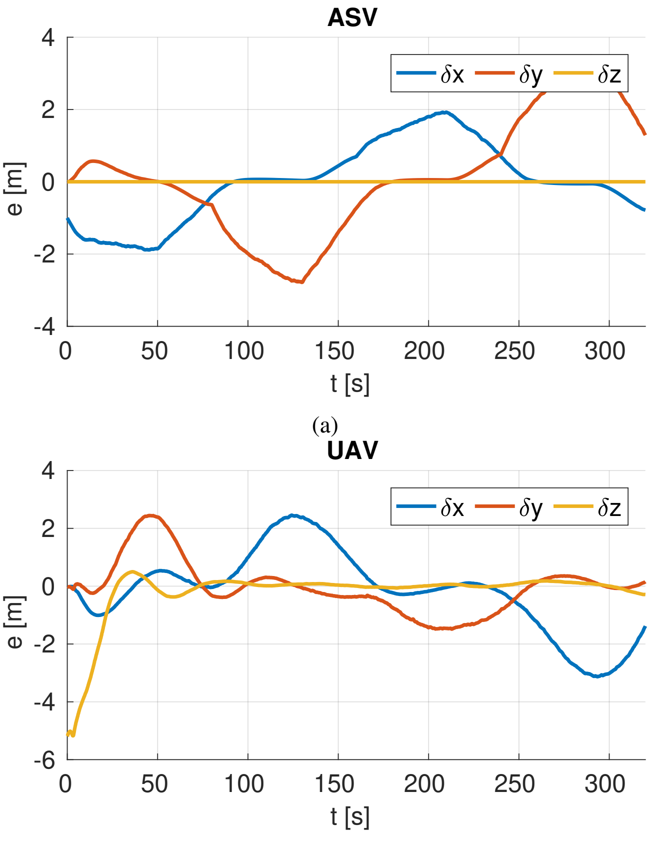
[3] J. Melo and A. Matos, "Guidance and control of an asv in auv tracking operations," in *OCEANS 2008.* IEEE, 2008, pp. 1–7.

[4] S. Bertram, C. Kitts, D. Azevedo, G. D. Vecchio, B. Hopner, G. Wheat, and W. Kirkwood, "A portable asv prototype for shallow-water science operations," in *OCEANS 2016 MTS/IEEE Monterey*, Sept 2016, pp. 1–6.

[5] G. G. Acosta, B. Menna, R. de La Vega, L. Arrien, H. Curti, S. Villar, R. Leegstra, M. D. Paula, I. Carlucho, F. Solari, and A. Rozenfeld, "Macabot: prototipo de vehiculo autonomo de superficie (asv)," in *XI Jornadas Argentinas de Robotica*, Nov. 2017.

[6] P. Mahacek, I. Mas, O. Petrovic, J. Acain, and C. Kitts, "Cluster space control of autonomous surface vessels," *Marine Technology Society Journal*, vol. 43, no. 1, pp. 13–20, 2009.

[7] F. Arrichiello, H. Heidarsson, S. Chiaverini, and G. Sukhatme, "Cooperative caging using autonomous aquatic surface vehicles," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 4763–4769.



(a)

(b)

Figure 19: Simulation results of the ASV following a squared (with rounded corners) trajectory and the UAV flying above it (corresponding to fig. 18). Position error for each vehicle: $(\delta x, \delta y, \delta x) = \delta \mathbf{p} = \mathbf{p} - \mathbf{p}_d$.

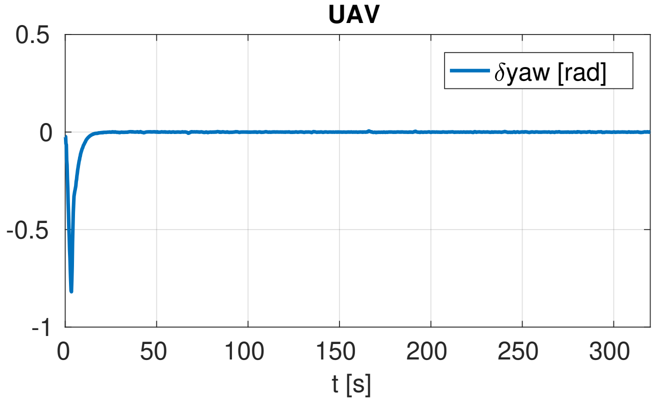


Figure 20: Simulation results of the ASV following a squared (with rounded corners) trajectory and the UAV flying above it. Yaw error of the UAV.

[8] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "Joint asv/auv range-based formation control: Theory and experimental results," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5579–5585.

[9] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[10] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1271–1278.

[11] I. Mas, P. Moreno, J. Giribet, and D. V. Barzi, "Formation control for multi-domain autonomous vehicles based on dual quaternions," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 723–730.

[12] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23

[13] Laboratório de Sistemas Autônomos – PURCS. (2018) Pró-Alertas Disaster Robotics project. [Online]. Available: https://github.com/disaster-robotics-proalertas/usv_sim_lsa