Coordinated ASV-UAV control for marine collision-free navigation

Leonardo Garberoglio
Grupo de Estudio de Sistemas de Control
Facultad Regional San Nicolás
Universidad Tecnológica Nacional
Buenos Aires, Argentina
Email: lgarberoglio@frsn.utn.edu.ar

Ignacio Mas
CONICET
Centro de Sistemas y Control
Instituto Tecnológico de Buenos Aires
Buenos Aires, Argentina
Email: imas@itba.edu.ar

Juan I. Giribet
Grupo de Procesamiento de Señales,
Identificación y Control
Facultad de Ingeniería
Universidad de Buenos Aires
CONICET
Email: jgiribet@fi.uba.ar

***Abstract*—There is an increasing interest in replacing a unique, complex, and expensive vehicle equipped with several sensors with a group of small vehicles, each of them carrying fewer sensors. There are several advantages in these segmented architectures, such as cost, flexibility, redundancy, and robustness, among others. The advantage of segmented architectures is even more noticeable if the vehicles carrying those sensors have different characteristics or environments of operation, e.g. aerial, terrestrial or marine vehicles. This work proposes a multi-robot system where an autonomous marine vehicle avoids obstacles relying on aerial images provided by an autonomous flying vehicle. Both robots navigate in a coordinated fashion increasing the detection area and allowing to adjust the obstacle detection horizon. In order to validate the control scheme two simulation scenarios are presented.**

***Index Terms*—Mobile robotics, unmanned aerial vehicles, autonomous surface vehicle, cluster space control**

I. Introduction

In recent years, great progress has been made on the use of autonomous surface vehicles (ASV) in water domain applications, such as water sampling [1], oil skimming [2] and surveillance [3]. Advances on electronic sensors, miniaturization of electric motors and advances on lithium-ion batteries, allowed designers to think of small vehicles as a feasible option for a variety of applications. 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], [5], and [6].

At the same time, Unmanned Aerial Vehicles (UAV) development has grown exponentially. They have the advantage of being able to see a wide panorama from the sky, making them useful for several applications such as monitoring tasks and surveillance. In [7] this property has been used to help an ASV navigate in an unstructured environment. This aerial perspective is beneficial to complement the navigation information acquired by the ASV, which usually relies on local information.

UAV and ASV collaboration has been proven to be effective for several tasks, in particular for environmental monitoring applications, such as characterization of highly dynamic coastal regions [8], oil spill mitigation [9], among others.

In [7] the UAV serves as an additional sensor, providing information to the ASV. Due to weight restrictions, the ASV navigation computer has more computational power than the UAV embedded computer. This is the reason why all the information is sent to the ASV and the marine vehicle runs the guidance algorithm. However, in some scenarios, a more reactive strategy should be adopted. In this work it is assumed that, instead of sending navigation information to the ASV to be processed afterwards, information from the UAV is used to generate appropriate commands to navigate the environment in execution time.

For systems of multiple mobile robots, one of the key technical considerations is the technique used to coordinate the motions of the individual vehicles. A wide variety of techniques have been and continue to be explored, drawing on work in control theory, robotics, and biology [2] and applicable for robotic applications throughout land, sea, air, and space. Notable work in this area includes the use of leader-follower techniques, in which follower robots control their position relative to a designated leader [10], [11], [12]. A variant of this is leader-follower chains, in which follower robots control their position relative to one or more local leaders, which, in turn, are following other local leaders in a network that ultimately is led by a designated leader [13] Researchers have demonstrated the use of automated tugboat fleets and swarm navigation techniques in order to move other ships [14] or the use of two ASVs for tele-supervised sensing of aquatic phenomena such as harmful algae blooms [15]. These solutions had been implemented with robots in the same domain (water or air or ground). 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. The cluster space [16] approach is a formation control method that promotes the simple specification and monitoring of the motion of a mobile multi-robot system. The strategy conceptualizes the n-robot system as a single entity, a cluster, and desired motions are specified as a function of cluster attributes, such as position, orientation, and geometry.

This work is particularly focused on the ability to navigate an autonomous marine surface vehicle avoiding collisions.

Such collisions may be due to obstacles on the water or the coastlines of the water course being traversed. Usually, obstacle detection relies on onboard sensors, such as lidars [17], cameras [18], or sonars [19]. This work proposes a distributed system where a camera is situated onboard an unmanned aerial vehicle flying above the ASV. This allows to improve the detection area and adjust the obstacle detection horizon.

A multi-robot formation definition is proposed to properly specify and monitor the parameters of a collision avoidance algorithm, such as imaged area and relative camera pose with respect to the marine vehicle. Based on the image-processing algorithm, a controller is defined to guide the formation through a collision-free course. Numerical simulations in different situations are presented to validate the approach.

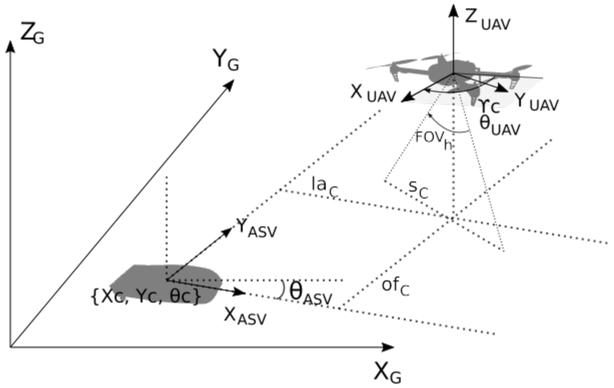

Fig. 1. Cluster Space Variables and frames

II. Multi-robot formation approach

This work implements the cluster space control method to specify the task at hand. This method allows for the selection of a set of independent system state variables suitable for specification, control, and monitoring tailored to the problem to be solved. These state variables form the system's cluster space. Cluster space state variables may be related to robot-specific state variables, actuator state variables, etc. through a formal set of kinematic transforms. These transforms allow cluster commands to be converted to robot specific commands, and for sensed robot-specific state data to be converted to cluster space state data. As a result, a supervisory operator or real-time pilot can specify and monitor system motion from the cluster perspective. The hypothesis is that such interaction enhances usability by offering a level of control abstraction above the robot and actuator-specific implementation details [20]–[23].

In this work we present a cluster space control scheme applied to an ASV-UAV formation. The UAV is equiped with a RGB Camera pointing downwards, with the main goal of detecting the ASV, obstacles and the coast-line for navigation purposes. With this setup, an operator can control the position of the cluster, the imaged area or swath width, and orientation for the camera, as well as the forward/backward or the left/right offsets of the UAV with respect to the ASV. While the operator or supervisor algorithm can specify values for these parameter, the cluster space control method will control the individual vehicles' positions to meet such specifications.

The first step in the implementation is the selection of an appropriate set of cluster space state variables. To do this, we introduce a cluster reference frame and select a set of state variables that capture key pose and geometry elements of the cluster.

A. Cluster Space State Variables

Figure 1 depicts the relevant reference frames for the UAV-USV formation and the selected cluster space parameters.

We have chosen to locate the cluster frame $\{C\}$ coincident with the ASV position and orientation. Based on this, the seven robot space state variables corresponding to the 3-DOF ASV pose and the 4-DOF UAV pose $\vec{r} = (x_{ASV}, y_{ASV}, \theta_{ASV}, x_{UAV}, y_{UAV}, z_{UAV}, \theta_{UAV})^T$ are mapped into seven cluster space variables $\vec{c} = (x_c, y_c, \theta_c, s_c, la_c, of_c, \gamma_c)^T$, where (x_c, y_c, θ_c) defines the cluster frame, the swath or imaged area width is defined by s_c, the backward/forward offset between ASV and UAV is defined by la_c, the left/right offset is defined by of_c, and UAV (i.e., camera) orientation is defined by γ_c. Controlling the last four parameters, an operator can change the image taken by the UAV's onboard camera.

B. Kinematic Equations

The forward position kinematics are the set of equations that allow the transformation from vehicles variables $\vec{r}$ to cluster variables $\vec{c}$.

$$\vec{c} = KIN(^G\vec{r}) \tag{1}$$

where

$$x_c = x_{ASV} \tag{2}$$
$$y_c = y_{ASV} \tag{3}$$
$$\theta_c = \theta_{ASV} \tag{4}$$
$$s_c = 2z_{UAV}\tan\left(\frac{FOV_h}{2}\right) \tag{5}$$
$$la_c = \cos(\theta_{UAV})\Delta_x + \sin(\theta_{UAV})\Delta_y \tag{6}$$
$$of_c = -\sin(\theta_{UAV})\Delta_x + \cos(\theta_{UAV})\Delta_y \tag{7}$$
$$\gamma_c = \theta_{UAV} \tag{8}$$

with: $\Delta_x = x_{UAV} - x_{ASV}$ and $\Delta_y = y_{UAV} - y_{ASV}$, and FOV_h is the horizontal field of view of the on-board camera.

The inverse kinematics allow the cluster variables to be transformed back into robot variables.

$$^G\vec{r} = KIN^{-1}(\vec{c}) \tag{9}$$

where

$$x_{ASV} = x_c \tag{10}$$

$$y_{ASV} = y_c \tag{11}$$

$$\theta_{ASV} = \theta_c \tag{12}$$

$$x_{UAV} = x_c + la_c \cos(\gamma_c) - of_c \sin(\gamma_c) \tag{13}$$

$$y_{UAV} = y_c + la_c \sin(\gamma_c) + of_c \cos(\gamma_c) \tag{14}$$

$$z_{UAV} = \frac{s_c}{2tan(\frac{FOV_h}{2})} \tag{15}$$

$$\theta_{UAV} = \gamma_c \tag{16}$$

By differentiating the forward and inverse position kinematics, the forward and inverse velocity kinematics can easily be derived, obtaining the jacobian ${}^{G}J({}^{G}\vec{r})$ and inverse jacobian $J^{-1}(c)$ matrices relating parameter velocities in both spaces:

$$\dot{\vec{c}} =^{G} J({}^{G}\vec{r})^{G}\dot{\vec{r}} \tag{17}$$

$$^{G}\vec{r} = J^{-1}(c)\dot{\vec{c}} \tag{18}$$

C. Cluster Space Controller

Figure 2 shows the navigation reference generator system. It takes cluster velocity (V_c), orientation (θ_c), and UAV-ASV offsets (la_c, of_c) from an operator and an image from UAV's on-board camera and calculates cluster parameter references. Figure 3 presents the control architecture for trajectory based cluster space control of an ASV-UAV system. A cluster level PID controller compares cluster position and velocity with desired trajectory values and outputs cluster commanded velocities, which are translated into individual robot velocities through the inverse jacobian. Data from the robots are converted to cluster space information through the forward kinematics and jacobian and fed back into the controller. The non-holonomic constraint given by the differential drive motion of the ASV effectively reduces its mobility from three degrees of freedom down to two. Due to this, the UAV-ASV cluster becomes a six-DOF system. As a consequence, an inner-loop ASV-level heading control is needed and the cluster space controller does not regulate the cluster parameter corresponding to the yaw orientation of the ASV relative to the cluster.

D. Obstacle Avoidance Algorithm

The UAV is equipped with an RGB camera, pointing downwards in order to detect the ASV and obstacles. The acquisition process is made by an OpenCV-to-ROS bridge. The image obtained is subjected to a cascade of image processing filters. The first step is the conversion from RGB to gray-scale image. The second stage applies a threshold filter in order to obtain a segmented version of the original image. After this, the white part of the image represents the water and the black part represents the ASV and the obstacles. In order to remove some water artifacts, a dilatation filter followed by an erosion filter are applied. The object detection happens on the top 100 pixels of the image. On that region a momentum calculation is performed and its X coordinate is evaluated. Without obstacles, the X value is $horizontal\_pixels/2$. With an object in the right (left) side of the image, this value decreases (increases), giving an indication of the obstacle location.

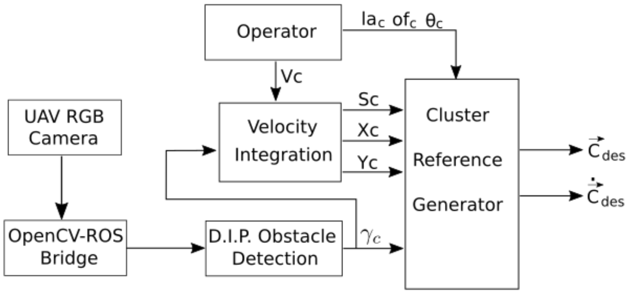

Fig. 2. Navigation reference generator diagram

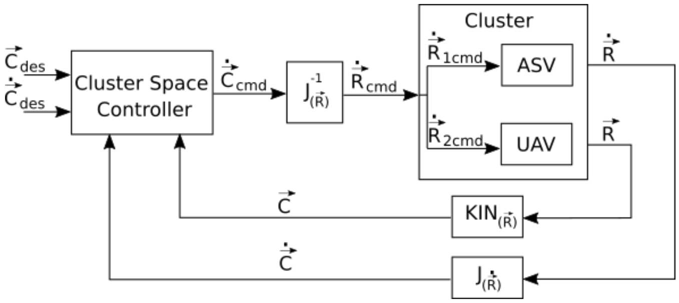

Fig. 3. Cluster space control architecture for an ASV-UAV system

III. Numeric validation

In this section, the simulation setup used to numerically validate the control scheme presented in the previous section is described.

The simulation was implemented on a desktop PC running Ubuntu 16.04 x86_64, with 8 GB of RAM and an Intel's i7 core. The simulation runs in a Robot Operating System (ROS)—Kinetic—and Gazebo 7.0. environment. The UAV was modeled using a Gazebo plugin from the Autonomous System Lab of ETH Zürich University [24] for the 3D-Robotics' IRIS quadcopter and the PX4 firmware. The communication with the model was done through a MAVROS interface. An 800×800-pixel RGB camera plugin is used to get an image from the UAV. With this setup and controlling the flight altitude of the UAV, objects with dimensions from a couple of centimeters to some meters can be detecteds. The world models used were generated with a tool developed by Clear Path Robotics [25]. The ASV model, from the same authors, consists of a differential-drive boat with two thrusters at the stern, commanded with forward speed and angular velocity around the z axis. The cluster space control logic is programmed as *ROS nodes* using *python* language. These nodes are connected to the simulated vehicles through *ROS* *topics*. Fig. 4 shows the simulated vehicles within the Gazebo simulator.

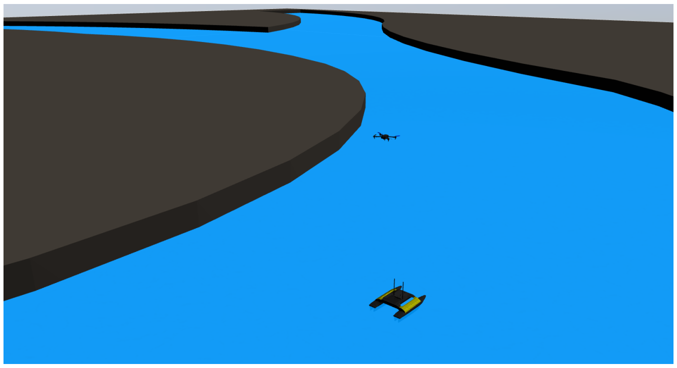

Fig. 4. ASV and UAV in the Gazebo environment simulator

The goal of this setup is to perform autonomous navigation over a course of water while performing object avoidance.

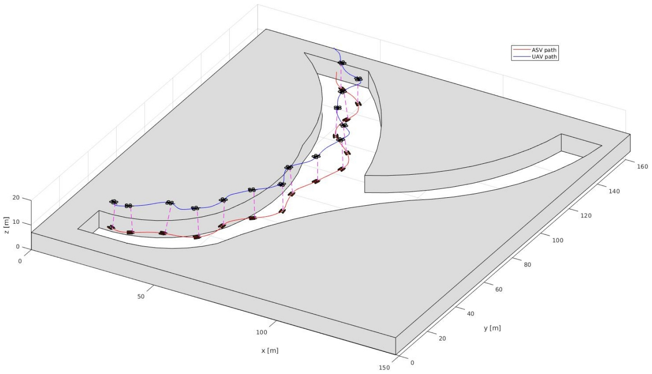

Fig. 5. Path of the ASV and UAV over a water course

IV. Result

In order to validate our approach, two simulation scenarios are evaluated.

A. Navigation over a course of water

In this scenario, initial cluster state parameters are $X_c = 0, Y_c = 0, \theta_c = 0, s_c = 10[m], la_s = 0, of_s = 0, \gamma_c = 0$. A forward cluster velocity V_c is commanded and the formation starts moving. This velocity provides $(V_{Xc}, V_{Yc}) = (V_c cos(\gamma_c), V_c sin(\gamma_c))$, where γ_c is set by the obstacle avoidance algorithm. Velocities are then integrated in order to produce new (X_c, Y_c) references. When *D.I.P. Obstacle Detection* block (see fig. 2) detects a coastline, the position (left/right) of the detected coast is used by the *Cluster Reference Generator* block in order to change the cluster state parameter γ_c and so to change the direction of the forward velocity V_c. The formation continues moving with its new orientation while *Cluster Reference Generator* produces new forward position references. In the presence of a new detected coast, the previous steps are repeated.

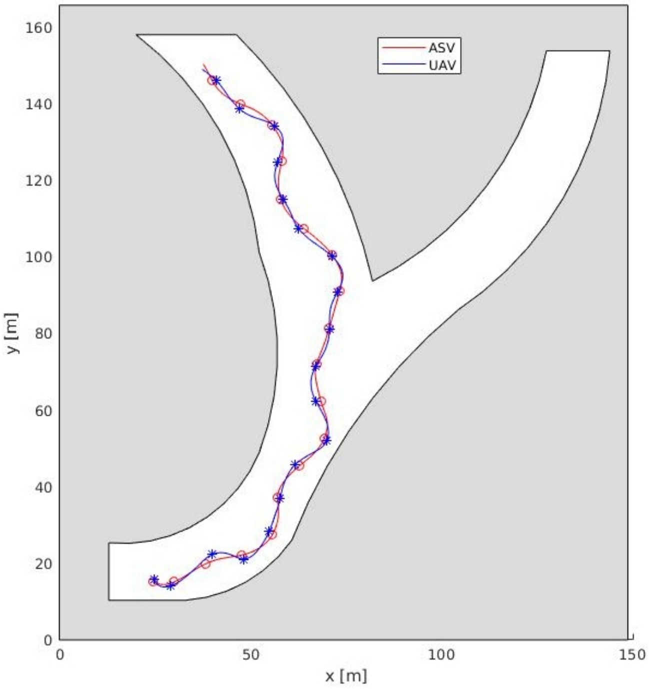

Fig. 6. Top view of Fig. 5

As seen in Fig. 5 and 6, the formation autonomously navigates the water course, detecting the coast-line and making corrections to the position and orientation reference values.

B. Object avoidance

In this scenario, the main goal is to avoid objects in the formation path while performing autonomous navigation. For this purpose, navigation starts with a constant cluster forward velocity V_c. The UAV flies at 7m above the ASV. When an object is detected in the path of the ASV, an avoidance maneuver is applied to the formation by changing the γ_c reference angle with a sign according to which side of the ASV the obstacle is on. Once the formation negotiates the object, γ_c is set back to its initial value.

Figures 7 and 8 show a *3D* and top view of the second simulation scenario. It can be seen that the formation is maintained while the obstacles encountered along the way are avoided by the ASV.

Figure 9 shows a sequence of four images from the UAV camera. The red dot is the calculated top region momentum after image binarization. In the first image the red dot is at the center of the frame, indicating that there are no obstacles in the formation path. In the second image, the red dot is at the right part of the image indicating that there is an obstacle at the left of the navigation path. The Cluster Reference Generator block commands a positive value for γ_c and the formation turns in clock-wise direction. Once the obstacle was avoided the resulting momentum is at the center of the image and a $\gamma_c = 0$ is commanded (image 3). Finally, in the fourth image, the formation continues navigating until the next obstacle is detected.

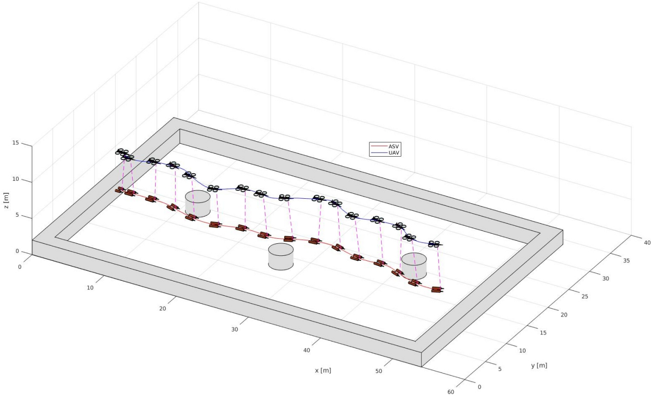

Fig. 7. 3d view of the object avoidance implementation

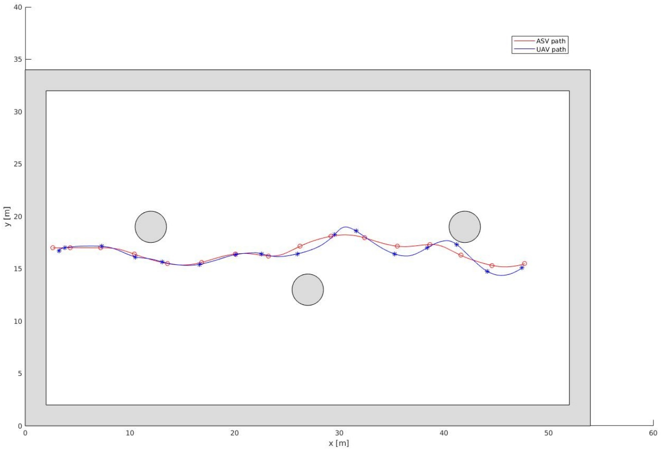

Fig. 8. Top view of the object avoidance implementation of Fig. 7.

V. Conclusion

In this work, an ASV-UAV formation that works in a coordinated fashion using the cluster space control technique is presented. Each robot operates in a different domain and therefore has a different sensing capabilities of the surrounding environment. The formation definition allows to define in a simple way the characteristics of the group that are relevant to a vision-based obstacle avoidance algorithm. The UAV equipped with an RGB camera identifies distant objects and avoidance maneuvers can be executed accordingly. The concept of sensors spatially distributed over a formation of autonomous robots allowed to solve the task of collision-free navigation in a simple way.

Two scenarios were presented in order to validate the proposed approach. Both show a good performance of the implemented system. The concept of using a formation of robots working in different domains to achieve collision-free navigation, was numerically validated. Future work will focus on obtaining experimental results and on analysing how a supervisory control can vary formation parameters to dynamically adjust the obstacle avoidance algorithm behavior to different environment characteristics.

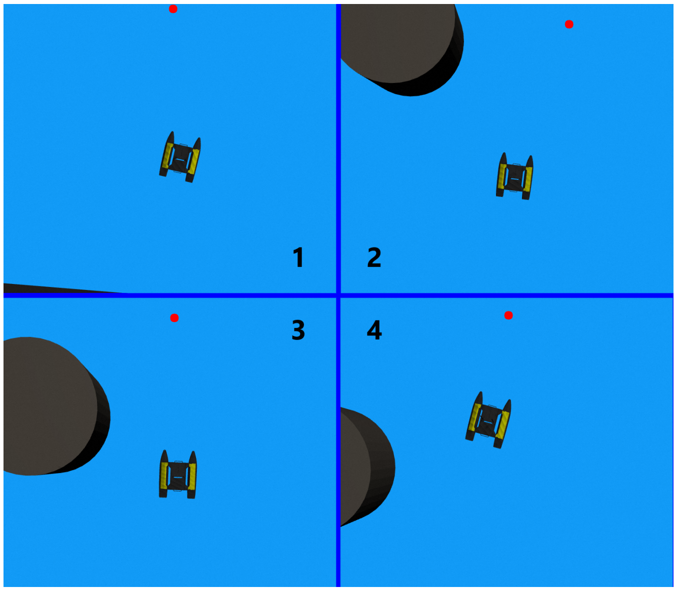

Fig. 9. Sequence of images from the UAV onboard camera

Acknowledgment

This work has been sponsored through the UBA-PDE-18-2019 grant from Universidad de Buenos Aires, Argentina, AMUTISN0005369TC grant from Universidad Tecnológica Nacional, Argentina, and PICT 2016-2016 grant from Agencia Nacional de Promoción Científica y Tecnológica (Argentina).

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] P. Kimball, J. Bailey, S. Das, R. Geyer, T. Harrison, C. Kunz, K. Manganini, K. Mankoff, K. Samuelson, T. Sayre-McCord *et al.*, "The whoi jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters," in *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2014, pp. 1–7.

[5] L. Garberoglio, P. Moreno, I. Mas, and J. I. Giribet, "Autonomous vehicles for outdoor multidomain mapping," in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, June 2018, pp. 1–8.

[6] 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.

[7] E. Pinto, F. Marques, R. Mendonça, A. Lourenço, P. Santana, and J. Barata, "An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Dec 2014, pp. 443–450.

[8] T. Marques, K. Lima, M. Ribeiro, A. S. Ferreira, J. B. Sousa, and R. Mendes, "Characterization of highly dynamic coastal environments, employing teams of heterogeneous vehicles: A holistic case study," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–8.

[9] D. Pedrosa, A. Dias, A. Martins, J. Almeida, and E. Silva, "Control-law for oil spill mitigation with an autonomous surface vehicle," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–6.

[10] Z. Wang and D. Gu, "A local sensor based leader-follower flocking system," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3790–3795.

[11] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur *et al.*, "A framework and architecture for multi-robot coordination," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 977–995, 2002.

[12] B. Smith, A. Howard, J.-M. McNew, J. Wang, and M. Egerstedt, "Multi-robot deployment and coordination with embedded graph grammars," *Autonomous Robots*, vol. 26, no. 1, pp. 79–98, 2009.

[13] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 813–825, 2002.

[14] J. Esposito, M. Feemster, and E. Smith, "Cooperative manipulation on the water using a swarm of autonomous tugboats," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1501–1506.

[15] J. Dolan, G. Podnar, S. Stancliff, K. Low, A. Elfes, J. Higinbotham, J. Hosler, T. Moisan, and J. Moisan, "Cooperative aquatic sensing using the telesupervised ocean sensor fleet," *Proceedings of Remote Sensing of the Ocean, Sea Ice, and Large Water Regions*, vol. 7473, pp. 1–12, 2009.

[16] C. A. Kitts and I. Mas, "Cluster space specification and control of mobile multirobot systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 207–218, 2009.

[17] J. Woo, J. Lee, and N. Kim, "Obstacle avoidance and target search of an autonomous surface vehicle for 2016 maritime robotx challenge," in *2017 IEEE Underwater Technology (UT)*, Feb 2017, pp. 1–5.

[18] F. D. Snyder, D. D. Morris, P. H. Haley, R. T. Collins, and A. M. Okerholm, "Autonomous river navigation," in *Mobile robots XVII*, vol. 5609. International Society for Optics and Photonics, 2004, pp. 221–233.

[19] S. A. Villar, F. J. Solari, B. V. Menna, and G. G. Acosta, "Obstacle detection system design for an autonomous surface vehicle using a mechanical scanning sonar," in *2017 XVII Workshop on Information Processing and Control (RPIC)*, Sep. 2017, pp. 1–6.

[20] T. Adamek, C. A. Kitts, and I. Mas, "Gradient-based cluster space navigation for autonomous surface vessels," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 506–518, 2015.

[21] I. Mas and C. Kitts, "Cooperative tasks using teams of mobile robots," in *IAENG Transactions on Engineering Technologies*, ser. Lecture Notes in Electrical Engineering, H. K. Kim, S.-I. Ao, M. A. Amouzegar, and B. B. Rieger, Eds. Springer Netherlands, 2014, vol. 247.

[22] P. Mahacek, C. Kitts, and I. Mas, "Dynamic guarding of marine assets through cluster control of automated surface vessel fleets," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 1, pp. 65 –75, Feb 2012.

[23] I. Mas and C. Kitts, "Object manipulation using cooperative mobile multi-robot systems," *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2012, WCECS 2012*, pp. 324–329, October 2012.

[24] 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

[25] Clear Path Robotics. Heron usv. [Online]. Available: https://www.clearpathrobotics.com/heron-unmanned-surface-vessel/