A Hierarchical Multi-ASV Control System Framework for Adversarial ASV Detainment

Mingi Jeong*, Julien Blanchet*, Joseph Gatto, and Alberto Quattrini Li

***Abstract*— In this work, we examine a multi-robot autonomous surface vehicle detainment problem, in which a heterogeneous fleet of naval drones attempts to intercept and entrap adversarial vessels. This problem presents interesting challenges relating to 1) behavior modeling and intent-recognition of the adversary; 2) multi-robot allocation and control for minimum-time, minimum risk detainment of multiple ships; 3) hybrid human-in-the-loop decision-making in a distributed, fast-evolving scenario. We've developed an initial approach in simulation that uses a hierarchical command structure whereby a human issues directives to autonomous vessel groups which then independently engage with their objective. Looking forward, we intend to expand our simulation to further address the above challenges and test our work in a real world scenario.**

I. Introduction

The use of autonomous multi-robot systems to perform complex tasks has become prominent in robotics research [1]. Multi-agent systems provide various advantages over single-agent systems including increased performance, efficiency, and occasionally safety. For example, in 2014, the US navy demonstrated Control Architecture for Robotic Autonomous Command and Sensing (CARCaS) with multiple unmanned vehicles and remote personnel monitoring the tasks such as maritime blockage/arrest, and target strikes [2]. In a defense setting, multi-robot systems are useful when the task at hand is complex yet too dangerous for a team of human operators to carry out. This motivates the need for a multi-agent robotic framework that allows for a human operator to instruct a team of robots to carry out complex dangerous missions from a safe, remote location.

The aim of this work is to develop solutions in the context of a *multi-robot autonomous surface vehicle (ASV) detainment problem*. The objective is for a robotic naval fleet consisting of heterogeneous ships to surround and detain (potentially) adversarial enemy ships in the minimum amount of time – see Fig. 1. Achieving and sustaining detainment demands coordinated behavior among friendly vessels, while the multiple, adversarial nature of the enemy ships present challenges to communication, resource allocation, and prioritization. Here we present preliminary work developing and simulating a hierarchical human-in-the-loop dispatch-and-detain system that uses a combination of flocking and vortex behaviors to achieve detainment.

First, we provide a formalization of the problem in Sec. II, followed by a brief overview of related work in Sec. III. We then describe the design of our system in Sec. IV and report preliminary simulation results in Sec. V. Lastly, we discuss insights and research questions arising from our progress in Sec. VI and provide a roadmap for next steps in Sec. VII.

*These authors contributed equally to this work. The authors are with Department of Computer Science, Dartmouth College, Hanover, NH USA {mingi.jeong.gr, julien.b.blanchet.gr, joseph.m.gatto.gr, alberto.quattrini.li}@dartmouth.edu

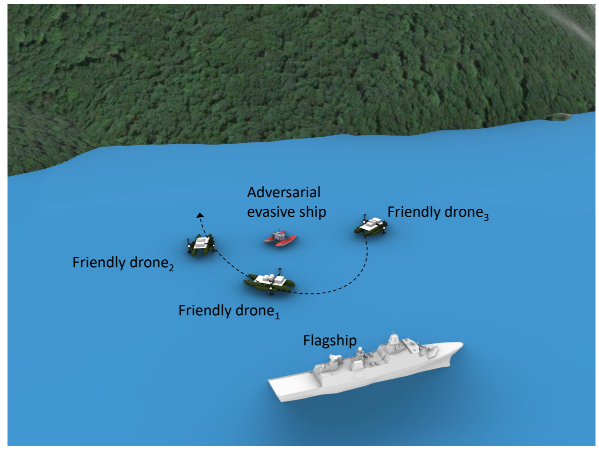

Fig. 1. Envisioned multi-Robot ASV detainment scenario: a fleet of friendly drones are capturing the adversarial evasive ship while coordinating with the affiliated flagship.

II. Problem Statement

The *multi-robot ASV detainment problem* involves the use of a fleet of autonomous surface vehicles to surround and block further motion of enemy ASV(s). The problem can be formulated as follows (see Fig. 1): On our friendly side, we have a human operator H in the loop, robotic flagship set $F = \{f_1, f_2, f_3, \ldots, f_i\}$, and sub-member drone set $D_i = \{d_i^1, d_i^2, d_i^3, \ldots, d_i^j\}$ for each flagship. Our fleet attempts to capture enemy ships in the set $E = \{e_1, e_2, e_3, \ldots, e_k\}$. Communication and control follows a heterogeneous and hierarchical structure $H > F > D$, representing a typical chain of command in real world, whereby the human operation H may issue directives to autonomous flagships F which then control and direct their fleet of drones. Such ships F, D, E are deployed in an environment with obstacles $\mathcal{E} \subset \mathbb{R}^2$.

The *objective of the system* is to establish a control scheme $H > F > D$ such that D achieves a detainment of E in the *minimum* amount of time T_E, where $T_E = \sum_{m=1}^{k} t_{e_m}$ and t_{e_m} is an individual detaining time of an enemy m being captured ($1 \leq m \leq l$). An enemy is defined as being detained when it is unable to move a distance of δ without coming within ϵ of a member of D (or some point along the shoreline). We refer to δ as the *detainment radius* and ϵ as the *capture distance*.

III. Related Work

A. Pursuit and Evasion

Our detainment problem is a form of a pursuit-evasion (PE) game, as we have one group attempting to pursue and capture an escaping adversary. There are tomes of research on search problems in general and on pursuit-evasion games specifically. [3] provides a survey of a subset of these works relating to mobile robotic applications, broken down into two key lines of research: (1) pursuit-evasion games taking place on graphs or polygonal environments and (2) probabilistic search of independently moving targets.

Pursuit evasion can be viewed as a specific problem within the space of autonomous search. Within autonomous search, the parameters of PE search problems yield large search spaces - for example, our naval pursuit evasion application consists of multiple heterogeneous searchers with constrained motion, possibly imperfect detection of evaders, within a finite or polygonal environment, in which there are possibly multiple adversarial evaders with bounded speed and turning angle.

B. Coordinated Control of ASVs

Getting a fleet of semi-autonomous (*flagship*) and fully autonomous (*drone*) ships to perform a capture operation on an escaping enemy ship is an exercise of multi-robot coordinated control. Peng *et al.* [4] provide a review of this body of work as it relates to ASVs. Peng *et al.* mention several design priorities for multi-ASV systems that are relevant in our case, including meeting communication constraints, avoiding collision with team members and obstacles, and adhering to maritime traffic rules (COLREGs). Ihle *et al.* [5] describe a system for formation path-following in which individual robots' path variables are synchronized. Almeida *et al.* [6] offer a formation path-following approach that uses Lyapunov-based techniques and graph theory to explicitly account for vehicle dynamics and inter-vehicle communication networks and implicitly compensate for ocean currents. Our approach uses techniques inspired by these works as building blocks to implement a more complex "surround and detain" behavior.

C. Multiple ASVs: Defense Applications

There are works in the literature that attempt to implement complex behaviors and address problems, similar to the *multi-robot ASV detainment*. Antonelli *et al.* [7] developed a naval-defense system whereby a team of ASVs is assigned to intercepting an enemy based on prior detection and identification as an assumption. The approach considered two costs functions – interception distance and interception time – and heuristically assigned a member of ASVs to intercept the intruder via online optimization. In the end, this system only dispatches one ASV for interception purposes, whereas our problem calls for the use of multiple drones to surround and prevent further movement of the enemy. Jiang *et al.* [8] describe a method for encircling a target with an ASV using a line-of-sight controller. This method is developed given an unknown target velocity and path, while it is designed to be robust in the presence of ocean currents. Our paper also attempts to navigate to and around a moving enemy, but unlike this work, we pursue the enemy with multiple robots and surround with the objective of blocking further movement.

D. Human-In-The-Loop Command of Robotic Fleets

Human-guided robotic fleets have been used in a variety of multi-robot systems. Authors have investigated various modalities of robot fleet control. From issuing commands via voice, motion gestures [9], or with multimodal commands [10], integrating human knowledge into the multi-robot pipeline is a common practice when the task at hand is highly complex [11]. Authors have also investigated varying degrees of human input. Elfes *et al.* [12] created a multi-robot exploration software architecture that oversees a fleet of extended deployment autonomous surface vehicles. Their system allows a single human operator to effectively supervise multiple robots through the use of a "sliding autonomy control architecture", which allows for robots to operate at varying levels of autonomy ranging from fully autonomous to fully human-teleoperated. In general, scenarios that assume human command of a multirobot fleet must contend with human operators' finite attention capacity and provide an interface that gives the human operators the capabilities they need for the task at hand while preventing the cognitive load of fleet control from being overwhelming. As we extend our work to more demanding scenarios with multiple capable adversaries and a larger number of friendly boats to coordinate, we foresee a need to carefully consider the attentive limits of the human operator and design an information and control interface suitable for our application, using methods described in [13].

IV. Preliminary System Architecture

The robots belonging to F, D, E have different behavior characteristics. Fig. 2 illustrates the high-level overview of their characteristics. On the friendly side, F is based on the hierarchical control architecture: (1) via H's voice control, F relays autonomous orders to D – capture a ship, follow a ship, and navigate to a point, and (2) via H's action input, F can be remotely controlled, i.e., go to a desired location. Then, D conducts fully autonomous mission supports as per the different parametrization by F – flocking, formation, and capture assist. These behaviors are running on top of an always-on obstacle-avoidance routine and one behavior can be active at a time. On the enemy side, E has different levels of capabilities: (1) static motion, (2) dynamic motion with random wandering, and (3) dynamic motion with an adversarial behavior, i.e., an evasive action from local approaches by D.

V. Preliminary Results and Evaluation

A. Experimental Setup

Based on the proposed system configuration, we first tested a prototype system on a lightweight 2-D simulator [14]. This enables fast and lightweight validation with a large

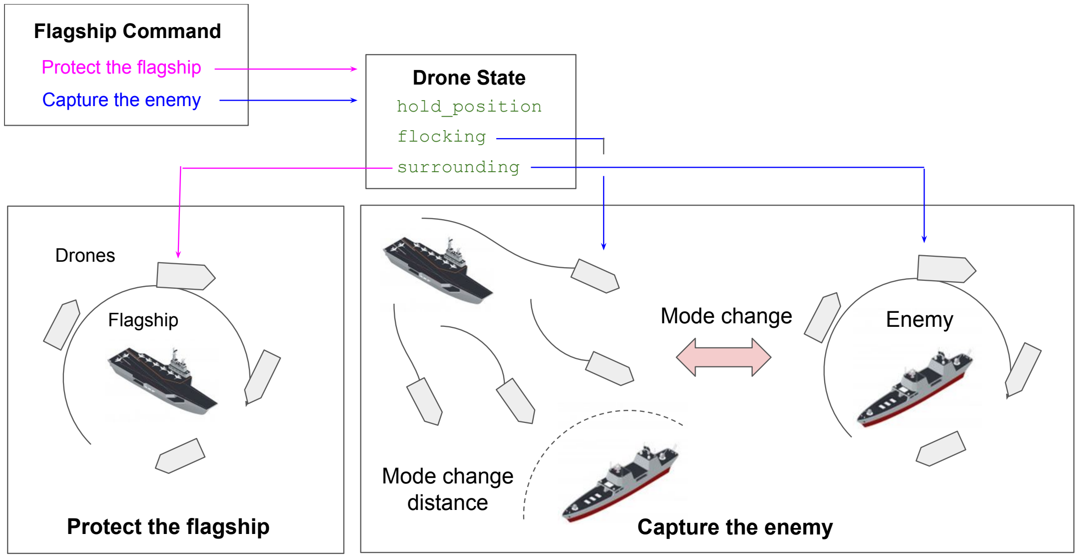

Fig. 2. Overview of drone states and behavior modules.

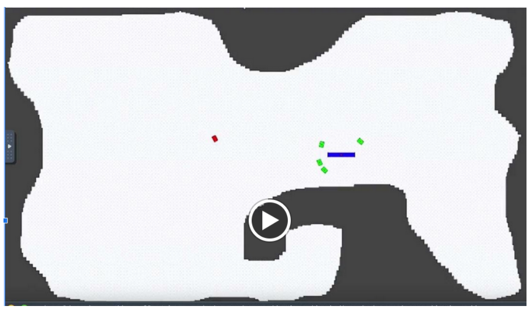

Fig. 3. Simulated environment in Stage ROS with flagship (blue), friendly drones (green), and the enemy ship (red).

number of agents under no complex external conditions such as wind, current, and wave. We used a simple custom-built marine environment with $320\,\text{m} * 180\,\text{m}$, as shown in Fig. 3. For these preliminary experiments, we assume the following fundamental components:

- We set the simulation to use one flagship and one enemy ship (H, E is set as $l = 1$).
- Fleet members (H, F, D) on the friendly side can communicate via a maritime satellite channel (Inmarsat-C [15]) in real-time, which is simulated by ROS [16] communication mechanisms;
- F, D, E are equipped with GPS for positioning, AIS for data exchange [17], and RADAR for ranging purpose. Hence, the friendly side can obtain the information of the enemy side or vice versa;
- Fleet members both on the friendly and enemy side have *a-priori* information of the surrounding environment, i.e., *nautical chart*; and
- All the robot models in F, D, E have kinematics based on a differential drive using twin motors.

B. Evaluation Criteria

Under the experimental setup of the *multi-robot ASV detainment problem*, we evaluate the task performance using the following criteria:

- **C1: Detainment Success (DS):** whether the system successfully captures the enemy or not. Specifically, E is surrounded by D and unable to move beyond the detainment radius δ without coming within capture distance ϵ of some $d \in D$. Additionally, we require the ships to maintain a successful detainment for a minimum time threshold, i.e., 5 seconds.
- **C2: Detainment Time (DT):** how long it takes for the system to successfully detain the enemy. Specifically, this metric refers to the time for H to issue an order and receive the completion feedback along the chain of the command.

More formally, we propose a quantitative metric for each criterion. In order to define a function for **C1**, we first define the *Furthest Ship Distance* (FSD) which returns the distance of the friendly ship closest to the enemy:

$$FSD(D, E) = \max(\text{dist}(E, d)) \ \forall d \in D$$

where $\text{dist}(\cdot, \cdot)$ is Euclidean distance in $\mathbb{R}^2$. We also define *Max Perimeter Edge Length* (MPEL), which returns the maximum distance between any two ships along the perimeter of the convex hull which encompasses E. Let $D_{hull} = \{d_1, d_2, ..., d_n\}$ for the n drones surrounding the enemy:

$$MPEL(D) = \max(\text{dist}(d_1, d_2)) \forall d_1, d_2 \in D_{hull}$$

Then, we define *DS* as follows:

$$DS(D,E)=\begin{cases}1 & \begin{aligned}&\mathit{if}\, P(E) \in ConvexHull(D)\\ &\wedge FSD(D,E) < \delta\\ &\wedge MPEL(D) < 2*\epsilon\end{aligned}\\ 0 & \mathit{else}\end{cases} \quad (1)$$

where $P(E)$ is the coordinate of enemy drone(s) in $\mathbb{R}^2$, and distances δ, ϵ are as defined in Section II. Eq. 1 evaluates whether or not the capture has been completed by computing the convex hull of the vertices of each friendly ship, and testing to see if the enemy ship is located inside the resulting polygon. The enemy ship will be deemed detained if it is both inside the convex hull and meets the distance constraints imposed by δ and ϵ.

For **C2**, we define *DT* as follows:

$$DT = T(H_{\text{feedback}}) - T(H_{\text{issue}}) \quad (2)$$

where $T(H_{\text{issue}})$ is the time when the human operator issues a command, e.g. 'capture the enemy' and $T(H_{\text{feedback}})$ is the time when the human operator receives confirmation feedback, e.g. 'task complete', from the system. The interval DT covers the performance of capturing E, affected by all system components: (1) hierarchical chain of command ($H - F - D$), (2) communication constraint, (3) motion constraint, (4) fleet composition, (5) adversarial level of E.

C. Preliminary Results

Our system can successfully detain enemy ships in a reasonable time with no collisions. Overall, the system achieved a 97.08% rate of detainment success. Table I lists success metrics by experimental conditions. In particular, the *wander* enemy behavior and the *k=3* drone-count conditions were associated with higher rates of detainment failure. As expected, starting distance was associated with detainment time. Variations in enemy behavior were only slightly associated with detainment time. The $k = 6$ drone condition exhibited similar mean detainment time as the $k = 3$ condition, but with higher variance.

TABLE I

RESULT SUMMARY: DETAINMENT TIME AND SUCCESS RATE BY EXPERIMENTAL CONDITION

	Mean DT	Std Deviation	Successes	Tests
hold position	49.59 sec	23.26 sec	54	54
wander	56.06 sec	24.09 sec	49	54
evade	59.67 sec	18.80 sec	54	54
3 drones	52.11 sec	18.37 sec	76	81
6 drones	57.86 sec	25.36 sec	81	81
near (50m)	41.46 sec	20.28 sec	53	54
medium (100m)	46.42 sec	11.62 sec	52	54
far (over 200m)	77.60 sec	13.77 sec	52	54

There are a few surprises exhibited by the data in the test runs. It is expected that the *wander* enemy behavior was more challenging to detain than the *hold position* behavior, as measured by both detainment success rate and mean detainment time. However, it was surprising for the *evade* behavior to have a higher detainment success rate than *wander* (see Table I). This hints that the enemy evasive AI may be counterproductive in its effort to escape capture, which is something we observed while monitoring the tests – while attempting to avoid capturing ships, the enemy often steered itself into corners.

As shown in Table I, increasing the number of drones from 3 to 6 yielded a better detainment success rate but did not yield an improvement in detainment times. This is surprising, as one might expect additional drones to enable the fleet to corner an enemy ship faster, or at least as fast. This outcome hints that drones may interfere with each other with the implicit coordination methods we used (flocking & surrounding) and illustrates the potential for explicitly-communicated, plan-based coordinated behavior to improve upon our approach.

VI. Follow-up Research Questions

Based on the preliminary results in the simulations, we propose the following research questions to be answered:

A. R1: How can the system on the friendly side reliably predict and estimate the intent of the enemy ASVs?

Although we assumed the use of AIS and GPS directly for the simulated environment, an essential improvement for the real-world scenarios is to enhance the intelligence on perception, including tracking, classifying, and identifying. The tracking system for targets, as one of the most important criteria, will include the following state estimation: speed, heading, position of the enemies at current time and a predicted time. Sensor fusion, as one feasible option, can deliver a more complete picture of the situation for human or automatic confirmation [18]. A key challenge will be to make the tracking system robust with less computational complexity, which ensures real-time computation. Another challenge is to model & track the behavior of enemies such that ships with suspicious behavior can be detected and flagged for detainment / search.

B. R2: How can we coordinate a decision-making process between the human and the autonomous agents?

In a practical application of the detainment problem there are cases where human inputs could be not only inevitable, but desirable, even with progress on automatic intent recognition. For example, consider a port-security application, where the "enemy" vessels in question could be innocent civilian ships, and where port-congestion may demand a prioritization of limited enforcement resources. In such a scenario, the system would be enhanced with a well-designed human-control interface (e.g., [19]) that provides the operator with the context and focused information needed to make critical high-level decisions and to communicate with detained or flagged-suspicious vessels.

C. R3: How can the system on the friendly side distribute the ASVs and allocate tasks in case there are multiple enemies to be captured?

Our preliminary work limits the scenario to a single flagship grouping and a single enemy vessel. Scaling this system to multiple vessel groups, whereby drones can be assigned to different flagship groupings, each chasing a different adversarial ship, presents a question of resource allocation. This question becomes yet more interesting when operating under uncertainty and when considering dynamic behavior. For example, given a limited number of drones, could the drones coordinate to herd adversarial ships into proximal locations for simultaneous detainment, or could the system reallocate drones on the fly to pursue targets that prove more evasive than initially calculated?

VII. Expectation and Next Steps

Fig. 4. Conceptual system in a real-world scenario: a fleet of our custom-made ASVs is in action on the field and a human operator gives commands while aboard a flagship.

We anticipate two thrusts of effort in continuation of our preliminary work: system development and real robot implementation.

From a control standpoint, we aim to extend our preliminary work to the multi-flagship, multi-enemy scenario. Doing so entails the development of a more sophisticated A.I. capable of intent recognition and resource allocation. We seek to add this intelligence to the autonomous flagships, which can serve as the coordinators and controllers of the swarm of drones, and which can request human operator input in fast-evolving situations.

We also seek to implement our control approach on a real-world system using a fleet of custom built inflatable robotic pontoon boats, as shown in Fig. 4 [20] (length $2.7\,\mathrm{m}$, beam $1.8\,\mathrm{m}$, max speed $2.5\,\mathrm{m/s}$, operating speed $1.0\,\mathrm{m/s}$ with over 3 hours battery life) with two electrical trolling motors, sonar, surface and underwater RGB cameras, laser scanner, and GPS.

Acknowledgement

This work is supported in part by the Burke Research Initiation Award and NSF CNS-1919647, 2144624, OIA1923004.

References

[1] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, Apr. 2021.

[2] G. Wu, T. Xu, Y. Sun, and J. Zhang, "Review of multiple unmanned surface vessels collaborative search and hunting based on swarm intelligence," *International Journal of Advanced Robotic Systems*, vol. 19, no. 2, 2022.

[3] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, 2011.

[4] Z. Peng, J. Wang, D. Wang, and Q.-L. Han, "An overview of recent advances in coordinated control of multiple autonomous surface vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, 2021.

[5] I.-A. F. Ihle, M. Arcak, and T. I. Fossen, "Passivity-based designs for synchronized path-following," *Automatica*, vol. 43, no. 9, 2007.

[6] J. Almeida, C. Silvestre, and A. Pascoal, "Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 14, 2010.

[7] G. Antonelli, F. Arrichiello, G. Casalino, *et al.*, "Harbour protection strategies with multiple autonomous marine vehicles," in *Modelling and Simulation for Autonomous Systems*, Springer International Publishing, 2014.

[8] Y. Jiang, Z. Peng, D. Wang, and C. L. P. Chen, "Line-of-sight target enclosing of an underactuated autonomous surface vehicle with experiment results," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, 2020.

[9] A. Couture-Beil, R. T. Vaughan, and G. Mori, "Selecting and commanding individual robots in a multi-robot system," in *2010 Canadian Conference on Computer and Robot Vision*, 2010.

[10] S. Pourmehr, V. (Monajjemi, S. A. Sadat, *et al.*, ""you are green": A touch-to-name interaction in an integrated multi-modal multi-robot hri system," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '14, Bielefeld, Germany: Association for Computing Machinery, 2014.

[11] D. St-Onge, V. Varadharajan, and G. Beltrame, "Tangible robotic fleet control aamas demonstration," 2019.

[12] A. Elfes, G. W. Podnar, J. M. Dolan, *et al.*, "The telesupervised adaptive ocean sensor fleet (taosf) architecture: Coordination of multiple oceanic robot boats.," 2008.

[13] A. Rule and J. Forlizzi, "Designing interfaces for multi-user, multi-robot systems," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '12, Boston, Massachusetts, USA: Association for Computing Machinery, 2012, ISBN: 9781450310635. DOI: 10.1145/2157689.2157705.

[14] R. T. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, 2008.

[15] *Inmarsat-c*, https://www.inmarsat.com/en/index.html, Accessed: 2022-10-01.

[16] M. Quigley, K. Conley, B. Gerkey, *et al.*, "Ros: An open-source robot operating system," in *ICRA workshop on open source software*, Kobe, Japan, vol. 3, 2009.

[17] International Maritime Organization, *International convention for the safety of life at sea (SOLAS)*, 1974.

[18] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, 2020. DOI: 10.1109/ACCESS.2020.2975643.

[19] A. Rule and J. Forlizzi, "Designing interfaces for multi-user, multi-robot systems," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '12, Boston, Massachusetts, USA: Association for Computing Machinery, 2012, ISBN: 9781450310635. DOI: 10.1145/2157689.2157705.

[20] M. Jeong, M. Roznere, S. Lensgraf, A. Sniffen, D. Balkcom, and A. Quattrini Li, "Catabot: Autonomous surface vehicle with an optimized design for environmental monitoring," in *Proc. OCEANS*, 2020.