Abstract—We present a study on the use of cooperative robots to execute a caging mission on the water’s surface. In particular, we consider the problem of using two robotic boats (under-actuated autonomous surface vessels) connected with a floating rope, to ‘capture’ a floating object from a known location on the water’s surface and ‘shepherd’ it to a designated position. This paper focuses on the cooperative control strategy of the two vessels. Each vessel’s behavior is governed by a supervisor software module that handles the communication with the other vessel and controls all elementary tasks that compose the overall mission. The elementary tasks, specifically developed for under-actuated vessels, are arranged by priority, and merged using a behavior-based approach, namely the Null-Space based Behavioral control. The proposed technique is validated by field experiments with two autonomous robotic boats on the surface of a lake.

I. INTRODUCTION

Marine robotics has fascinated many researchers due to its wide application domain.

Here, we focus on marine surface robots, in particular Autonomous Surface Vessels (ASVs). A wide body of literature covers motion control problems for a single vessel. The reference book [13] surveys the main automatic control issues for an ASV, including tracking control and dynamic positioning. For under-actuated vessels, i.e., those unable to apply forces in all directions, tracking control approaches have been more recently investigated in [17], [2] and [8], while dynamic positioning techniques have been investigated in [19], [20] and [3].

Several studies have been presented on the use of fleets of marine surface robots; most focus on the formation control problem. The work in [10] deals with the formation control problem for under-actuated vessels with communication constraints, and solves this via non-linear techniques. The work in [15] uses Lagrange multipliers to solve the formation control problem in the presence of slowly varying environmental disturbances and measurement noise.

Despite the interest in these topics, the applications with real vessels in the field are very limited. Notable exceptions include [11], [22], [1] and [9].

In this paper, we present a study on the use of cooperative ASVs to execute a caging mission on the water’s surface. In particular, we consider the problem of requiring two under-actuated ASVs, connected by a floating rope, to capture a floating object from a known location, and to bring it to another designated position. This problem has applications involving object deployment (e.g., buoys or marine sensor network nodes), automatic rescue and recovery maneuvers. Moreover, using a fishing net instead of the rope, similar missions can be performed to clean the water’s surface or collect material for biological investigation. Caging and box pushing have been widely studied for wheeled multi-robot systems, and a large amount of literature covers most of the main aspects of the problem [12], [14], [16] and [23]. However, to the best of our knowledge, this is the first paper focusing on caging in an active medium, such as the water. It is also one of the few applications of cooperative ASVs in the field.

We focus on the coordination control strategy for a team of under-actuated surface vessels, and we present results of preliminary experiments in the field. The control architecture is organized in layers working at different frequencies. At the highest level, a supervisor, using local sensor data and receiving information from the other vessel, dynamically defines the active tasks and their priority order. At an intermediate level, the Null-Space based Behavioral (NSB) control is used to merge the multiple tasks organized in priority and to define the motion directives. Finally, a low-level controller, specifically designed for the available vessels, generates the thruster and rudder commands to realize the motion directives received from the NSB.

An early use of the two boats in a cooperative mission is described in [7], wherein two ASVs were required to cooperatively reach several locations spread in the environment while respecting a communication constraint. Compared to [7], this paper is novel since it presents a different application that, given the high number of constraints to be simultaneously taken into consideration, has an increased complexity. For the mission considered here, new task functions were built to solve the specific problem of the caging mission with under-actuated vessels, and a new coordination control strategy is proposed. Moreover, the internal software architecture of the robots has been completely revised and now uses the Robot Operating System (ROS) framework [21].

The experimental results validate the coordination control strategy, and this represents the first step to achieve the caging mission in the aquatic setting. The final objective, indeed, is to equip the two boats with a floating rope and use them to recover a floating object from a known position. With this goal in mind, we aim at investigating the effects of the rope and of the carried object on the dynamics of the vessels to design a better low-level controller. Moreover, we
will develop an active target that can report its position to the vessels via wireless communication. This will be used during the course of the mission to verify that the target has been caged, or if it gets lost during transportation.

The rest of the paper is organized as follows. Section II introduces the caging mission in the aquatic environment, and the set of elementary tasks that will be used to achieve the mission. Section III presents the multi-layer control architecture on-board each vessel. Section IV gives a mathematical description of the elementary tasks composing the mission. Sections V and VI respectively present the experimental set-up used for the experiments and the results of the experiments in the field. Finally, in Section VII we derive some conclusions and explain future work on the topic.

II. MISSION

In the proposed mission, we want to use two under-actuated ASVs to cage a floating object and to bring it to a desired position. The vessels are notionally equipped with a floating flexible rope, thus their motions have to be coordinated in such a way that the target is captured and it does not get lost during the transportation, see the sketch in Figure 1. To accomplish the mission, multiple parameters and constraints have to be simultaneously considered.

- First, the positions of the vessels, as a formation, have to be controlled. We control the mean position of the two vessels (i.e., their barycenter) and we make it move toward either the target or the designated position.
- Assuming that one vessel is the port and the other one the starboard, we can define the angle and the advancing direction of the formation. We control this angle to ensure that the vessels approach the target with the correct orientation. Moreover, to avoid losing the object during the transportation, backward motion are prohibited.
- The distance between the vessels must be less than the rope length.
- The vessels must not cross each other (to avoid twisting the rope).
- The vessels must avoid collisions’ between themselves. Avoidance of external obstacles will be added in future research.

Based on the importance of the tasks, we prioritize them in the following order:

1) collision avoidance: to ensure the integrity of the vessels;
2) distance between the vessels: to ensure the integrity of the rope;
3) cross avoidance: to ensure the proper execution of the mission and avoid the rope getting stuck in the boats’ propellers or rudder;
4) orientation of the formation: to ensure that the approach toward the object is from the proper direction, and to avoid losing the object during the transportation;
5) barycenter position: this has the lowest priority since the formation has to only move when the other tasks are properly achieved.

Fig. 1. Cooperative caging with surface vessels.

III. CONTROL ARCHITECTURE

The vessels use a layered control architecture (see the sketch in Figure 2) where each layer operates at a different frequency. At the highest level a supervisor, using sensor data and receiving information from the other vessel, dynamically defines the active tasks and their reference values. At an intermediate level, the Null-Space based Behavioral control merges the multiple tasks organized by priority and defines the motion directives to the vessel. Finally, a low-level controller, specifically designed for the test-bed vehicles, generates the thruster and rudder commands to make the vessels follow the motion directives received from the NSB.

A. Supervisor

The supervisor is in charge of selecting the tasks to be activated and defining their desired values based on the system state and the information exchanged with the other vessel. It works in strict correlation with the NSB and it also defines when a task, in specific conditions, has to subsume a lower priority one (by outputting a null-space matrix having only zeros as entries). For the caging mission, the supervisor is organized as a Finite State Machine (FSM) with the following states:

1) initialize the system and wait until the other vessel has been initialized;
2) approach the object with a proper angle assigning the desired parameters of the task functions (e.g., desired position of the barycenter, formation desired angle and allowed maximum distance between the vessels);
3) overtake the object to ensure that the caging has been properly executed; eventually, decrease the inter-vessel distance to reduce the object escape possibility;
4) move toward the designated position, avoiding maneuvers that may cause the object to be lost.
5) release the target.

The transition from one state to another depends either on threshold values associated with task errors or the supervisor state on the other vessel. This functionality has been added since the vessel communicates asynchronously and the measured task errors can be slightly different between the two vessels. Thus, as soon as one of the supervisors reads that the desired threshold values have been reached, both the supervisors have to switch to the respective next states.
B. NSB

The single tasks that compose the overall mission are managed and composed using the NSB. This is a behavior based approach, developed by some of the authors of this paper, to manage multiple tasks organized in priority order. The main idea of the approach is to define elementary task functions for each individual task, and to use a projection mechanism (based on the null-space projection matrix) to compose the tasks following their priority order. Indeed, the tasks are combined so that the lower priority tasks do not effect the higher priority ones. The null-space projection matrix of a task, in fact, filters out the velocity component that would conflict with the higher priority ones. The null-space projection is the task error defined as

\[ \tilde{\sigma}_i = \sigma_i - \sigma_i^d, \]

where \( \sigma_i \in \mathbb{R}^m \) is the task variable to be controlled, \( m \) is the task function dimension, and \( p = [p_p \ p_s] \in \mathbb{R}^3 \) is the vector containing the positions of the port and starboard robots (respectively \( p_p \) and \( p_s \)). The velocity reference to solve the \( i^{th} \)-task is calculated as

\[ v_i = J_i^\dagger (\tilde{\sigma}_{i,d} + \Lambda_i \tilde{\sigma}_i), \]

where \( J_i^\dagger \) is the pseudo-inverse of the task function Jacobian matrix, \( \Lambda_i \) is a constant positive-definite matrix of gains and \( \tilde{\sigma}_i \) is the task error defined as \( \tilde{\sigma}_i = \sigma_{i,d} - \sigma_i \) (\( \sigma_{i,d} \) and \( \sigma_i \) are the desired value of the task function and its derivative).

When the mission is composed of multiple tasks, the overall velocity vector is obtained by merging the outputs of the single tasks respecting their priority order; that is, before adding a contribution of a single task to the overall vessel velocity, a lower-priority task is projected onto the null space of the immediately higher-priority task so as to remove those velocity components that would conflict with it. If the subscript \( i \) also denotes the priority of the task with, e.g., Task 1 being the highest-priority, in a three task mission the overall vessel velocity is given by:

\[ v_{NSB} = v_1 + N_{1,1} v_2 + N_{1,2} v_3, \]

where \( N_{1,k} \) is the projection matrix into the null-space of the tasks from 1 to \( k \). In particular, defining \( J_{1,k} \) as

\[ J_{1,k} = \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_k \end{bmatrix}, \]

the null-space projection matrix \( N_{1,k} \) is elaborated as

\[ N_{1,k} = \left( I - J_{1,k}^\dagger J_{1,k} \right). \]

C. Low-level controller

The low-level controller is aimed at steering the vessel along a desired path and moving it with a desired velocity [13]. Receiving motion reference commands from the NSB, the maneuvering controller has to generate the actuators commands. Basing on the model for ASVs in [13], and considering the under-actuated propulsion system of the vessel (see Figure 3), a maneuvering controller for the ASVs used herein is designed following the approach proposed in [18].

![Fig. 2. Control architecture for a team of two under-actuated vessels.](image-url)

![Fig. 3. The vessel propulsion system and of the velocity reference angles.](image-url)

The ASVs used for experiments have two independent thrusters that can be used to apply a force in the surge
direction and a torque to change the vessel yaw; moreover, the vessels have a rudder that facilitate high-speed turns. The low-level controller is expressed as the sum of a heading autopilot and a surge control aimed at causing the vessel to follow the velocity reference commands given by the NSB. Indeed, following the control architecture of Figure 2 and based on the sketch in Figure 3, the output of the NSB for a single robot is a velocity vector $\mathbf{U}_{NSB}$ that can be geometrically represented through its norm $\|\mathbf{U}_{NSB}\|$ and its direction $\chi_{NSB}$. These are given to the low-level controller as desired surge and heading/advancing direction. The heading autopilot is aimed at controlling the heading of the vessel to make it move in the desired direction $\chi_{NSB}$; it regulates the propulsion torque and the rudder angle to correct the orientation of the vessel. The surge control has to make the norm velocity of the vessel to track the $\|\mathbf{U}_{NSB}\|$ value generated by the NSB; however, the vessel moves at full speed only when the orientation error is null. Thus, the surge control works as a PI controller regulating the advancing direction multiplied by a scaling factor depending on the orientation error.

IV. TASKS

In this section we present a brief description of the elementary tasks which make up the caging mission.

![Diagram of reference angles for the task functions.](image)

**A. Barycenter**

The barycenter of the team is the mean position of the two vessels. Thus, the task function is expressed by:

$$\sigma_b = f_b(p_p, p_s) = \frac{(p_p + p_s)}{2},$$

where $p_p, p_s$ are the positions of port and starboard vessels, respectively.

The output of the task function is:

$$v_b = J_b^\top(\hat{\sigma}_{b,d} + \Lambda_c\hat{\sigma}_o),$$

where $J_b \in \mathbb{R}^{2\times4}$ is the Jacobian matrix defined as

$$J_b = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix},$$

the psuedo-inverse of the Jacobian matrix is $J_b^\dagger = J_b^\top \left(J_bJ_b^\top\right)^{-1}$, and $\hat{\sigma}_o = \sigma_{b,des} - \sigma_b$ is the error task function. The desired value of the task function ($\sigma_{b,des}$ and $\hat{\sigma}_{b,des}$) expresses the desired trajectory of the barycenter.

**B. Orientation**

This task controls the angle of the formation. Referring to Figure 4, it has to control the $\phi$ angle that represents the advancing direction of the formation. Thus, the task function is defined as:

$$\phi = \text{atan2}(\Delta Y, \Delta X) + \frac{\pi}{2},$$

where $\Delta Y, \Delta X$ are the projections along $Y$ and $X$ of the vector $p_s - p_p$.

The Jacobian matrix is calculated as

$$J_o = \frac{1}{\|p_s - p_p\|^2} \begin{bmatrix} p_s[1] - p_p[1] \\
- \left(p_s[0] - p_p[0]\right) \\
p_s[0] - p_p[0]\end{bmatrix},$$

and the output is elaborated as:

$$v_o = J_o^\top\left(\hat{\sigma}_{o,d} + \lambda_o\hat{\sigma}_o\right).$$

**C. Collision-avoidance**

The task is implemented individually on the two vessels. In the following we describe the task function for the vessel on the 'starboard' (the description for the other vessel is analogous). In the presence of the port vessel (considered as a static obstacle) in the advancing direction, the aim of the task is to keep the starboard vessel at a safe distance from the port one. Thus, its implementation produces a velocity in the $p_s - p_p$ direction aimed at controlling the inter-vessel distance. Formally, the task function is

$$\sigma_c = \|p_s - p_p\| \in \mathbb{R},$$

where $J_r = \frac{v}{\|p_s - p_p\|} \in \mathbb{R}^{1\times2}$ is the task Jacobian where $\frac{v}{\|p_s - p_p\|}$ is the unit vector aligned with the $p_s - p_p$ direction. Defining $\sigma_{c,d} = d$ as the desired distance, the task output is

$$v_c = J_r^\top\Lambda_c(d - \|p_s - p_p\|).$$

Possible motions in this task null-space are all the motions that do not change the distance between the vessels. Thus, the null-space projector matrix projects the velocity commands of the lower-priority tasks along the tangential direction of a circle centered in the port vessel and passing through the starboard vessel.

**D. Distance**

This task has to ensure that the rope does not get broken by controlling the distance between the vessels. The task is formally equivalent to Collision-avoidance, however it is activated under a different condition (i.e., when the distance is greater than a given threshold). It also has different desired values and task gains.
E. Cross avoidance

This task is designed to prevent the vessels from crossing each other with the goal of avoiding to twist the rope (see Figure 5). To this aim, referring to Figure 4, the task output velocity vector for the starboard vessel is computed as a function of the angle $\theta$ (computed positive in the counter-clockwise direction). The task function and null-space projection matrix are built geometrically. For example, considering the position, in polar coordinates, of the starboard vessel in the reference frame of the port one, the task function outputs a velocity in the opposite direction to the $Y_1$ axis when $0 < \theta < \pi/2$ and when $\pi/2 < \theta < \pi$. The velocity in the $X_1$ direction in generally not controlled and this degree of freedom is used to build the null-space projection matrix. However, when the starboard robot is close to the positive $Y_1$ semi-axis, the $X_1$ direction is controlled in order to avoid twisting of the vessels. For the port vessel an analogous procedure is used. If the rope becomes twisted, an emergency procedure is activated.

![Port Boat Starboard Boat](image)

**Fig. 5.** Sketch of twisted condition.

V. EXPERIMENTAL SET-UP

The robotic platform (Figure 6) is composed of two ASVs designed by the University of Southern California’s Robotic Embedded Systems Lab (RESL). Each ASV is an OceanScience QBoat-I hull with a length of 2.1 m and a width of 0.7 m at the widest section. Each boat weighs 48 kg with instrumentation and batteries. The on-board computing package consists of a Mini-ITX 2 GHz dual-core computer for primary computation, and a Gumstix 400 MHz single-board computer for auxiliary operations related to the science package. A 28 Ah sealed lead acid (SLA) battery is used to power the computer and all sensors, and a 32 Ah AGM battery is used for the drive motors and the rudder. The ASVs have been observed to have a nominal runtime of 6 hours.

The sensor suite on the ASVs consists of a navigation package and a science package. The navigation package of the two vessels are slightly different. Both the vessels have a uBlox EKF-5H GPS that provides global position updates at 2 Hz. One vessel is equipped with a 3DM-GX3-25 IMU with integrated compass sampled at 100 Hz; the other is equipped with an ISIS IMU working at 100 Hz, and a PNI TCM-2 Compass sampled at 5 Hz. The science package on-board both the vessels consists of a TWI wind speed and direction sensor, an Imaginex profiling sonar, and a Hydrolab MiniSonde MS-5 water chemistry sensor. The science package is used to gather weather, bathymetry, and water chemistry data from lakes and marinas.

The ASVs are controlled by software built using the open-source framework Robot Operating System (ROS) [21]. The framework provides a structured communications layer on top of the running operating system allowing intercommunicating nodes and services to be developed easily. ROS also includes a handful of tools to aid in experimental robotics. The different nodes of the ASVs manage specific portions of the system or of the control architecture, and they generally run at different frequencies as reported in Figure 7.

![fig6.jpg](image)

**Fig. 6.** The USC RESL Autonomous Surface Vessels connected with a floating rope.

![image](image)

**Fig. 7.** Software architecture on-board each vessel.

VI. EXPERIMENTAL RESULTS

We performed preliminary experiments in the field with the two under-actuated vessels. The experiments were executed in the Echo Park Lake in Los Angeles (lat: 34°4’22.06"N, lon: 118°15’38.74"W). They focused on testing the coordination control strategy of the vessels to accomplish the caging mission. To this aim, a target and a designated position were passed to the vessels assigning their GPS coordinates. The parameters of the task functions are selected as in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barycenter</td>
<td>$A_b = 0.1 * I_2$</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\lambda_0 = 1.0$</td>
</tr>
<tr>
<td>Distance</td>
<td>$\lambda_d = 1.0$</td>
</tr>
<tr>
<td>Cross</td>
<td>$\lambda_c = 0.2$</td>
</tr>
</tbody>
</table>

![image](image)

**Table.** Parameters of the task functions.

Figure 8 shows the paths of the vessels during the overall mission. At the beginning of the mission, the vessel supervisors synchronize between themselves to make the vessels start moving at the same time; thus, the vessels start to move toward the target (the green dot in Figure 8) keeping a maximum distance of 20 m. The desired orientation angle of the formation is given by the barycenter-target direction. Once the barycenter task function error is lower than 3 m.
the supervisors switch to the respective next states, and they command the vessels to move along their actual advancing direction for a few meters; this functionality is added to make the vessels overtake the target and to ensure it has been properly caged. Then, the supervisors command the vessels to move toward the designated position (the blue dot in Figure 8). To avoid losing the target, during the motion the vessels are not allowed to move backwards with respect to the angle of the formation. The vessels stop when the distance from the designated position is less than 3 m.

Figure 9 shows the error of the barycenter task function during the complete experiment. From the plot, the three different states of the supervisors can be recognized. In the first 40 s the vessels move toward the target reducing the task function error. When the supervisor state changes, a new position in the vessels’ advancing direction is commanded to ensure that the target is properly caged (this causes a step in the barycenter error); the vessels move in this new direction until the distance from the barycenter and the initial position of target is greater than 5 m (around 55 s). Then, the supervisors switch again and assign the designated position as the desired value of the barycenter task function (causing a new step to the barycenter error). The mission ends when the vessels’ barycenter reaches the designated position.

Figure 10 shows the orientation error during the overall mission. Comparing Figure 10 with Figure 9, it is worth noting the rapid convergence to zero of the orientation error, with respect to the barycenter error. This shows the basic functionality of the NSB method that gives priority to the highest priority tasks. Figure 11 shows the distance between the vessels during the experiment; the distance task function is activated when the distance is greater than 14 m; the obstacle avoidance is activated when the distance is less than 8 m.

The video attached to the paper shows a reconstruction from the experimental data of the vessels’ paths where the position of the vessels is given by the GPS readings, while their heading is given by the compass values. From the video, it is possible to observe the effect of the current and wind making the vessels consistently drift during their motion.

Since the vessels have a maximum velocity of 1.5 m/s, the saturation management technique presented in [?] has been used to manage the actuator saturation respecting the NSB task-priority.

We have also performed preliminary tests connecting the boats with a 20 m floating rope. We have executed the caging mission in remote control and have performed preliminary tests of autonomous navigation with the presented control architecture. Despite the fact that the mission is achievable with the realized set-up, the presence of the rope strongly...
influences the system dynamics and strongly degrades the low-level controller performances. Thus, our next steps will be focused on including the effects of the external forces generated by the rope into the vessel dynamic model and to re-derive the low-level controller to compensate for these disturbances. Moreover, an active target will be developed to properly achieve the caging mission; that is, we will develop a floating object with a GPS and wireless communication capabilities so that we can integrate it with the two boat system. The active target will be used to send its instantaneous position to the vessels; moreover, it will be used both to control whether the caging has been properly executed, and to eventually detect if the target get lost during the transport.

VII. CONCLUSIONS

In this paper we have introduced the problem of caging floating objects on the water’s surface using cooperative marine ASVs. In particular, we focused on the coordination control strategy to make a team of two under-actuated vessels achieve the assigned mission, and we tested the performance of the coordination strategy in the field using two ASVs. The experimental results show the effectiveness of the proposed technique. To accomplish the complete caging mission in the field, future research will focus on the development of a low-level controller to take into consideration the forces generated by the floating rope and by the caged object that strongly effect the vessels’ dynamics. Moreover, we will develop an active floating target that will send the vessel its GPS location.

ACKNOWLEDGMENTS

This work was supported in part by the NOAA MERHAB program under grant NA05NOS4781228 and by NSF as part of the Center for Embedded Network Sensing (CENS) under grant CCR-0120778, by NSF grants CNS-0520305 and CNS-0540420, by the ONR MURI program (grants N00014-09-1-1031 and N00014-08-1-0693) by the ONR SoA program and a gift from the Okawa Foundation. The research leading to these results has received funding from the European Community’s Seventh Framework Programme under grant agreement n. 231378 (STREP project Co3-AUVs - Cognitive Cooperative Control for Autonomous Underwater Vehicles).

REFERENCES