In *Proceedings of Oceans '95 MTS/IEEE, San Diego, October, 1995.*

EXPERIMENTS IN AUTOMATIC RETRIEVAL OF UNDERWATER OBJECTS WITH AN AUV

Howard H. Wang * Stephen M. Rock †
Stanford University Aerospace Robotics Laboratory
Durand Building 250
Stanford, California 94305

Michael J. Lee ‡
Monterey Bay Aquarium Research Institute
160 Central Avenue
Pacific Grove, California 93950

ABSTRACT

Automated retrieval of instrumentation packages from the sea floor is a task well-suited for autonomous underwater vehicles. By constraining the generic object-manipulation problem to one in which the object is specifically designed to be picked up by a particular robotic system, both sensing and manipulation can be simplified. Readily available vision processing hardware along with well-established vision algorithms can be used to locate the package.

The strategy for automated object retrieval by an underwater vehicle is encoded using a finite-state machine (FSM). Each phase of the automatic pickup is represented by a state. As monitors detect the successful completion or failure of a phase, events are issued that trigger the FSM to transition to the next phase of the task or to an alternative, corrective action. These events cause the vehicle control system to change control modes or desired setpoints within the activated controller.

The automatic retrieval task will be demonstrated experimentally with the *OTTER* testbed. These results will be presented at the conference.

INTRODUCTION

Autonomous underwater vehicles (AUVs) are on the brink of being fully operational tools for daily use by scientists and engineers for undersea exploration and utilization. Much of the technology needed to build AUVs that are capable of carrying sensors for missions to survey large sectors of the ocean for physical and chemical quantities have been developed and demonstrated. By investigating technologies that can add subsea-intervention capabilities to AUVs, a new dimension of AUV applications will be enabled. These capabilities include autonomous placement and retrieval of sensors on the ocean floor, the servicing of long-term, monitoring stations, and docking with undersea platforms for data exchange and power transfer.

As a part of the joint Aerospace Robotics Laboratory (ARL) and Monterey Bay Aquarium Research Institute (MBARI) underwater robotics research program, technologies for automated intervention are being extended to AUVs. ARL's background in fundamental control theory and robotic systems is combined with MBARI's operational experience in ocean engineering and the practical requirements of its marine scientists to form a solid foundation that has been drawn upon by this research.

Background

Small, untethered AUVs have a niche in the domain of submersibles that are used for subsea exploration and utilization. The tether is a dominating cost in running a remotely-operated vehicle (ROV). The tether-handling equipment places constraints on the size of the support vessel and the number of crew required to manage it. Untethered AUVs should be cheaper to operate since they can be launched from most any ship of opportunity with only minimal support personnel.

However, by removing the tether, the high speed, high-bandwidth communications link between the submersible and the human operator is broken. Direct operator control of the robot's

*Doctoral Candidate, lazarus@sun-valley.Stanford.Edu
†Professor of Aeronautics and Astronautics, rock@sun-valley.Stanford.Edu
‡Senior Research Engineer, lemi@hp850.Mbari.Org

actuators becomes impractical. Instead, there are two other ways in which an untethered, unmanned submersible can be controlled to accomplish tasks underwater. In the first, the robot is completely autonomous and entirely responsible for doing the task itself–implying the use of advanced, "intelligent" planners and schedulers to manage robotic control. Current state-of-the-art technology using artificial intelligence is limited in the types of tasks that totally autonomous robots can perform reliably. The capabilities of AUVs are far from their ROV cousins which are under the control of skilled human operators.

In the second method, human operators can direct robotic actions by issuing commands at an elevated level of abstraction that does not require high-speed or high-bandwidth data-transfer rates. Using acoustic modem technology, a limited communications channel between the human and the AUV is possible for exchanging telemetry and user commands. However, the on-board controller must be augmented to assume some of the responsibilities of controlling the lower levels of robotic functionality. Using on-board sensors, feedback controllers for continuous actions along with discrete-event controllers for logical actions can produce complex robot behavior. Human intelligence is used where computer intelligence is currently lacking such as sensor-data interpretation and on-line planning for anomalous conditions. This type of "intelligent control" is a step towards producing highly capable, totally autonomous underwater vehicles.

At ARL, we have developed a framework called "Task-Level Control" (TLC) for building intelligent controllers. Humans accomplish global objectives by commanding on-board computers to control the robot to do autonomous tasks. The person is concerned with what needs to be done and why, whereas the on-board controller manages the how a task is carried out. A task is defined to be a set of actions that a robot can execute by itself to reach a goal. Commands such as "capture that object", "put this there", "survey an $n \times m$ area at position x, y, z", etc., are issued by the human supervisor to direct a robot to accomplish a task. A sequence of these tasks would constitute a complete mission. A more detailed description of this control philosophy can be found in [8, 11]. A control/software architecture implementing TLC is described in [13, 9, 5].

In the MBARI/ARL program, we have had success implementing various autonomous tasks for underwater robots. Using video cameras as direct sensors for automatic control, we have successfully demonstrated the following tasks: automatic tracking of underwater objects by the robot, automatic station keeping using only on-board sensors, and automatic creation of mosaics of the ocean floor [1, 3, 2]. In addition, we have demonstrated automatic positioning and trajectory following by an AUV. Cooperative arm/vehicle control was accomplished by using models of the hydrodynamic interaction of the arm and vehicle to eliminate the disturbance forces generated during arm motion [4].

Thus, a logical step to increase progressively the autonomous capabilities of underwater robots is to investigate methods that allow AUVs to pick up objects off of the ocean floor by themselves.

Motivation

Using a multi-degree-of-freedom (DOF) manipulator to pick up arbitrary objects is a large research area in itself without undertaking the additional difficulties of doing it underwater with a free-floating submersible. However, there is a subset of the general case that is more straight forward to achieve while remaining useful in many underwater applications.

The generic task can be greatly simplified by replacing a multi-DOF manipulator with a fixed boom and by restricting the types of objects to be picked up to ones with well-known geometry that have been designed specifically for detection and retrieval. Even under these constraints, many types of tasks can still be accomplished. For example, an AUV can be used to place instruments at precise locations on the ocean floor and then later to retrieve them autonomously. AUVs can use similar strategies to connect to an underwater dock in order to transfer data and power for long duration missions.

Thus, we are limiting the scope of the problem for this initial effort. The package to be retrieved will be designed to be easily picked up by an AUV. A lighted beacon is attached to the package that can be sensed by the AUV for location and control during the retrieval operation. Instead of an articulated arm, a fixed boom will be used to pick up the package. Even in this simplified scenario, a host of problems need to be addressed before AUVs can robustly retrieve instruments in the open ocean. For example, strategies to deal with currents creating disturbances near the retrieval site will need be developed. There may be constraints to the approach of an AUV to the package, e.g. the package is sitting next to a wall or in a crevice. Can intelligent controllers be developed such that the pickup can be robust to those situations?

While we are not prepared to address all of these issues at this time, this paper will report on the research done to date. In the following sections, the experimental apparatus will be described and then the algorithmic elements forming the intelligent controller will be discussed. Preliminary experimental results will be presented at the conference.

EXPERIMENTAL SETUP

The *OTTER* underwater robotic testbed is being used to develop and test the automated pickup task through experiments in a test tank environment. The basic testbed is fully discussed in [12]. In this section, we will emphasize the modifications made to *OTTER* to support the pickup task.

OTTER Testbed

Table 1 contains a brief list of the important characteristics of the *OTTER* testbed (seen in Figure 1). A small tether connects the testbed to the user control station at the surface through two video lines and an Ethernet connection. Batteries on board *OTTER* are trickle charged also using the tether.

Figure 1: **The *OTTER* Underwater Robot.**

The operator can monitor the robot's progress and issues commands through a variety of computer workstations. A range of computer-generated environments is available to the user from full 3D, virtual reality to a simple 2D graphical display.

Dimension	2.1 m long, 1 m wide, 0.5 m high
Displacement	150 kg dry (approx. 450 kg wet)
Thrusters	2 x 1 hp main thrusters, 6 x 1/4 hp maneuvering thrusters
Maximum Depth	1000 m estimated
Maximum Speed	4 kts estimated
Power	Nickel-Cadmium batteries, 750 W hrs on board
Control	On-board real-time VME computers, full 6-DOF control
Sensors	stereo video cameras, inclinometers, depth, compass, rate gyros, SHARPS positioning

Table 1: General Characteristics of *OTTER*.

For picking up the "instrument package", a small boom is mounted on the front of *OTTER*. The boom is outfitted with a light that can be detected by the stereo-vision system described below. The object representing a scientific instrument is simply made of PVC with a mounted light that can also be tracked by the vision system.

Vision System

Physically, the stereo-vision system consists of two Pulnix 840N, black and white, CCD video cameras mounted in custom underwater housings. The cameras are connected to a Datacube DIGICOLOR digitizing board with a ROIStore board to store digitized video frames. All of the vision processing after digitization is accomplished in software with a MVME 167 (25 MHz 68040). A diagram of the system is seen in Figure 2. Although this layout has the basic arrangement used for earlier experiments in control from vision completed with *OTTER* (see references [1, 3, 2]), we would like to note that the custom vision-processing hardware boards employed previously are not being used for the experiments described in this paper.

The cameras are mounted in a fixture that allows the divergence, rotation, and tilt of one camera to be adjusted with respect to the other for calibration. The camera mount is attached to an unique pan/tilt mechanism (see Figure 3). While the pan degree-of-freedom is normally configured, tilting of the cameras is achieved by rotating the camera mount around the outside circumference of the front-lateral thruster duct. This configuration was chosen to reduce the space requirement

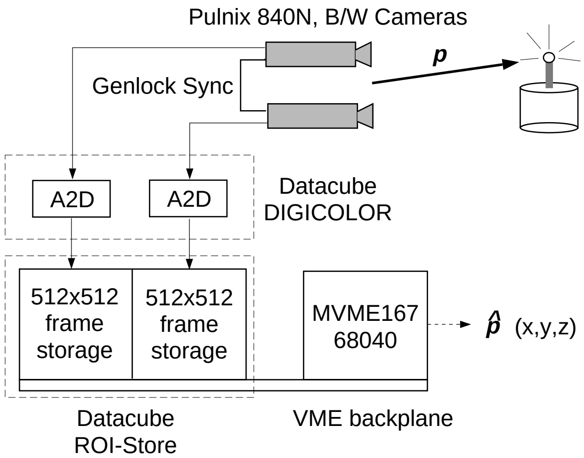

Figure 2: **Schematic of the Vision System.**

of the camera pan/tilt system.

Figure 3: **Stereo cameras mounted on a pan/tilt.**

ALGORITHMS

The fundamental algorithms used in the vision processing and the control of *OTTER* to pick up objects are not new. The theory behind stereo vision and proportional derivative controllers have been around for a long time. What may be unique is how they were applied together with event-based logic to create a higher-level controller to accomplish the pickup task in a real experimental setting.

Stereo Vision

We will first describe what happens during vision processing. To begin, both the arm and package to be picked up are outfitted with lights. Thus, the vision system "sees" the arm and the package as bright dots. Correspondence between dots seen by each camera of the stereo pair must be made to determine which dots were produced by which objects[1]. The geometry for a single object as seen by stereo cameras is diagrammed in Figure 4.[2]

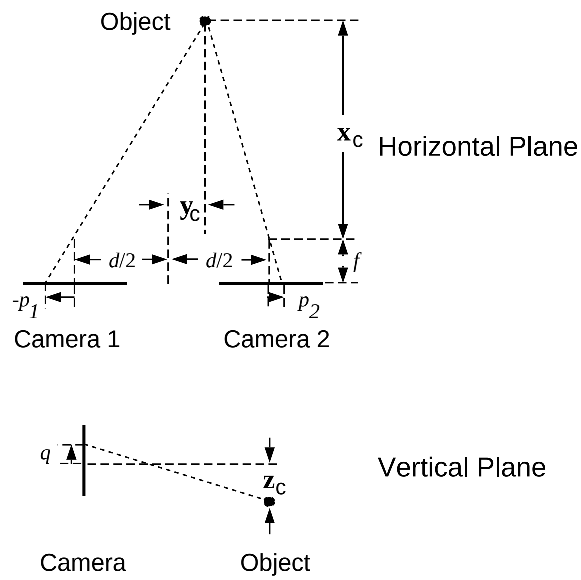

Figure 4: **A single object seen by a stereo pair of cameras.**

The relative location of an object can be calculated using the following equations after identifying the dots representing the object.

$$\mathbf{x_c} = \frac{fd}{p_2 - p_1}$$

$$\mathbf{y_c} = \frac{d}{2}\frac{p_2 + p_1}{p_2 - p_1}$$

$$\mathbf{z_c} = \frac{q\mathbf{x_c}}{f}$$

where,

$(\mathbf{x_c}, \mathbf{y_c}, \mathbf{z_c})$ is the position of the object relative to the stereo cameras.

d is stereo separation between the two cameras.

f is focal length of the lenses (identical for both cameras).

p_1, p_2 are the horizontal positions of the dots produced by the object on the focal plane of each camera.

[1] We will use the generic term "object" to refer to both the arm and the package that is to be picked up.

[2] The cameras are assumed to be identical and simply represented using a pinhole approximation.

q is the vertical position of the dots on the focal plane (assumed to be identical for both cameras).

The inertial, body-fixed, and camera-relative coordinate systems being used are shown in Figure 5.

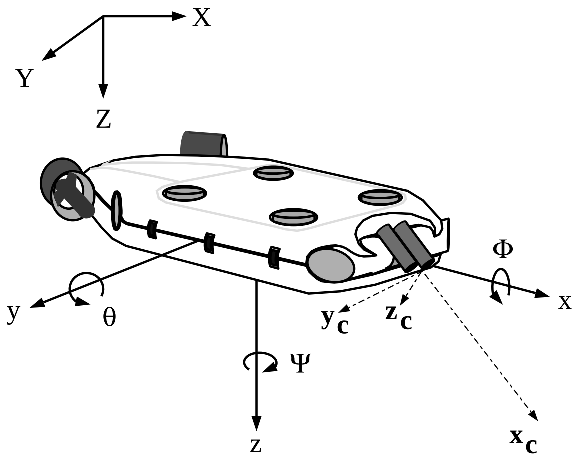

Figure 5: *OTTER*'s **coordinate systems.**

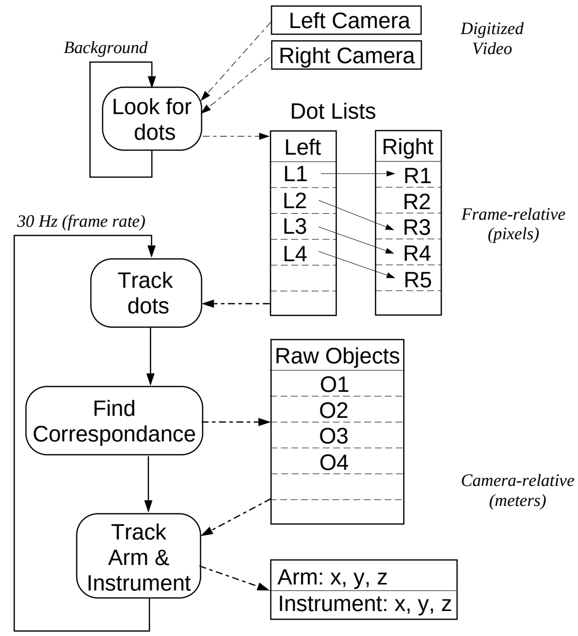

Figure 6: **Vision processing schematic.**

The vision processing strategy is based on a 2D system created in previous research for tracking objects in a plane [6] and now modified to track objects in 3D by stereo cameras. A background process continually searches for new dots in the cameras' field-of-view. If a new dot is found, it is placed on a list of known dots for each camera. For every newly-acquired camera frame, each known dot is tracked by looking for it in a region to where the dot is expected to have moved by using an estimated velocity. If a known dot has not been seen, and thus not tracked, after a specified amount of time, the dot is considered lost and removed from the known list. For all dots that are tracked, their positions in the video frame and estimated velocities are updated (See Figure 6).

Then for each dot in the left camera (arbitrarily chosen) that has not already been paired with a dot in the right camera, we look in the right camera for the corresponding dot produced by the same object. Dots are considered to be corresponding only if they lie on the same horizontal line in each camera. This algorithm only works if new objects do not appear on the same horizontal plane at the same time. For two or more objects lying in a plane and with only a single point of light representing each object, the correspondence problem cannot be resolved without additional information since it is ambiguous as to which dot corresponds to which object between the cameras.

After correspondence is made, the camera relative (x_c, y_c, z_c) positions of objects represented by each pair of dots are calculated using the equations previously noted, and a list of objects that are tracked is produced. Simple heuristics are used to identify the arm and the "instrument" to be picked up. The output of the vision system is then used in a feedback controller to guide the AUV to first find and then approach and pick up the instrument package.

In vision sensing experiments, the precision of the vision system was determined to be within a centimeter with the object at a meter from the cameras. The precision degraded with the square of the range of the object as theoretically predicted. The accuracy and precision of the vision system are highly dependent on the physical alignment and calibration of the cameras. However, since the position of the arm and object are directly measured by the same sensor, if the error vector between the arm and object is driven to zero, then the arm will be at the object. In this way, problems with poor calibration show up mainly as unknown gains in the system.

Strategy for Automatic Retrieval

Automatic object retrieval is one of many complex tasks that an underwater robot can be expected to perform. In the *OTTER* system, the robot is commanded by the user at a task level. It is best first to define what constitutes a task within our control paradigm. Then, we will describe the encoding method that we use to program a computer system to control a robot to accomplish the task. Finally, the task of automatically retrieving underwater packages will be described within the programming framework.

Characteristics of tasks A task consists of a desired goal and a set of actions that a robot can execute to reach the final state. The most basic actions are ones that low-level controllers manage to drive robotic motion in a stable and predictable manner. The main feature of a task is the knowledge of how to coordinate these actions to accomplish the specific objective. A task may be composed of many sub-goals that must be achieved first before reaching the final goal. Each of these sub-goals may, in turn, be tasks themselves. Thus our tasks are recursively defined.

In the creation of the task, the programmer is encoding knowledge and prediction of how the environment should change when the commanded actions are executed. When certain conditions arise indicating the completion of a sub-goal, different pre-programmed actions are triggered to carry the system into the next phase of the task.

The programming of tasks incorporates a priori knowledge of actions and possible outcomes, and thus, makes many assumptions. During actual task execution, these assumptions may proved to be false, and provisional actions may have to be taken. Many alternate execution paths may be encoded for each task, but there will always be singular events which were not anticipated. The recognition and resolution of these anomalous conditions must be relegated to a higher level of control, for example, a human supervisor or AI planner.

Task encoding method Many different methods can be used to encode tasks in a program to control computers to monitor and direct robotic actions at a high, logical level. Most of these methods appear in the form of textural languages. These languages include rule-based production systems, Prolog with its backward-chaining search engine, synchronous languages such as Esterel, and even low-level languages such as C.

For various reasons, we have chosen to use a graphical language based on the finite-state machine (FSM) paradigm to encode tasks. First, it is fairly natural to decompose a task into different phases or states of completion, and then, to link the states together by events that trigger new actions to advance the robotic system towards the completion of the task. The generic definition of FSMs gives them the exact structure necessary to encode a task that is decomposable into states, events and transitions. Second, finite-state machines can be graphically represented and programmed. We believe that the paradigm of graphical programming languages is more natural and intuitive for the human programmer to formulate and inscribe high-level strategies to accomplish tasks.

We are using RTI's *ControlShell* real-time, control framework to program both low-level controllers and high-level tasks for the *OTTER* system [10, 7]. ControlShell provides an FSM facility to create, manage, and execute state-transition graphs that represent event-driven processes, i.e. tasks. The transition between states are specified by a boolean relation between stimuli (user-defined events) that initiate the transition, an action (user-defined function) to execute when the transition is taken, and alternative states to where the transition finishes that are dependent on the result of the action. The task of automatic object retrieval is a concrete example that is described fully in the next section and seen in Figure 7.

The automatic retrieval task The nominal strategy for automatic object retrieval is straight forward and not complicated. Upon receiving the command to retrieve an instrument, *OTTER* transits to and begins to search the area around the coordinates where the instrument is thought to be located. After the vision system "sees" the lighted beacon of the instrument, it begins to follow an approach trajectory toward the instrument. This approach trajectory moves *OTTER* to a position where the boom can be guided on a final trajectory for pick up. After the instrument is picked up, *OTTER* is sent home.

This strategy has been graphically encoded in an FSM and can be seen in Figure 7. Alternate courses of action have been specified for contingencies such as losing track of the object or missing the object during the pick up phase of operation. At any time, an "ABORT_TASK" command can be sent that causes *OTTER* to stop what it is doing and return home.

To understand in depth how the FSM controls the robot, we will look closely at what happens when the robot is in the *Approaching* state of the task. In this phase, *OTTER* is following a global trajectory (in our case, using the SHARPS positioning system) to move toward the instrument. If the vision system loses track of the instrument, perhaps due to occlusion of the instrument by a

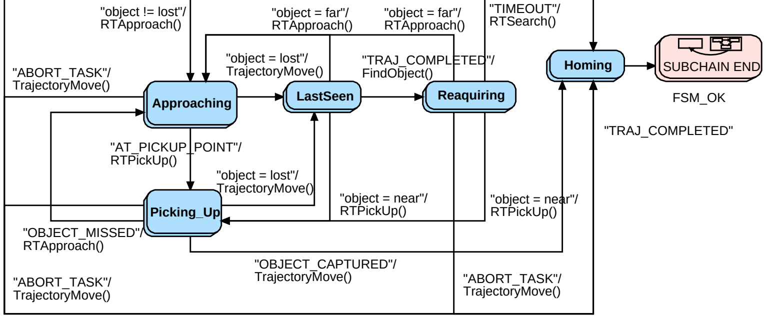

Figure 7: **The finite-state machine controller.**

passing school of fish, *OTTER* is commanded to hold station and search the last known position of the instrument with the vision system. If the instrument has not been found after a certain period of time, *OTTER* is commanded to begin following a global search pattern. However, if the vision system reaquires tracking of the instrument, the distance that the instrument is away from the submersible will determine whether or not to enter the final pickup phase or to resume the approach trajectory towards the instrument. When *OTTER* has reached the end of the approach trajectory and is close enough to begin the pickup phase, *OTTER* switches from a global-control mode to a local-control mode using the error vector from the boom to the instrument, produced by the vision system, in guiding it during the final pickup maneuver.

At the time this paper is being written, we are in the process of beginning in-water experiments with *OTTER* and a pseudo-instrument for pickup. It is expected that this strategy and its implementation in the FSM will be modified as warranted by experimental results.

CONCLUSIONS

Underwater intervention is an ability required by science and industry for exploring and exploiting undersea resources. Automatic retrieval of instrument packages is a step towards providing intervention capabilities on AUVs. We are working on developing both the hardware and software technologies to make automated, underwater intervention possible. A set of hardware and software facilities described in this paper is being used in experiments in automatic retrieval. Preliminary results are expected to be presented at the conference.

References

[1] R. L. Marks, S. M. Rock, and M. J. Lee. Automatic object tracking for an unmanned underwater vehicle using real-time image filtering and correlation. In *Proceedings of IEEE Systems, Man, and Cybernetics*, France, October 1993. IEEE.

[2] R. L. Marks, S. M. Rock, and M. J. Lee. Real-time video mosaicking of the ocean floor. In *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, Cambridge, MA, July 1994. IEEE.

[3] R. L. Marks, H. H. Wang, M. J. Lee, and S. M. Rock. Automatic visual station keeping of an underwater robot. In *Proceedings of IEEE Oceans 94 Osates*, Brest, France, September 1994. IEEE.

[4] T. W. McLain, S. M. Rock, and M. J. Lee. Experiments in Hydrodynamic Coupling Compensation for an Underwater Arm/Vehicle System . In *Proceedings of IEEE Oceans 95*, San Diego CA, October 1995. Accepted for publication.

[5] G. Pardo-Castellote and S. A. Schneider. The Network Data Delivery Service: Real-Time Data Connectivity for Distributed Control Applications. In *Proceedings of the International Con-*

ference on Robotics and Automation, San Diego, CA, May 1994. IEEE, IEEE Computer Society.

[6] S. Schneider. *Experiments in the Dynamic and Strategic Control of Cooperating Manipulators*. PhD thesis, Stanford University, Stanford, CA 94305, September 1989. Also published as SU-DAAR 586.

[7] S. A. Schneider. Real-time innovations, inc. 954 Aster, Sunnyvale, CA. email: info@rti.com.

[8] S. A. Schneider and R. H. Cannon. Experimental Object-Level Strategic Control With Cooperating Manipulators. *The International Journal of Robotics Research*, 12(4):338–350, August 1993.

[9] S. A. Schneider, V. Chen, and G. Pardo-Castellote. Object-Oriented Framework for Real-Time System Development. In *Proceedings of the International Conference on Robotics and Automation*, Nagoya, Japan, May 1995. IEEE, IEEE Computer Society.

[10] S. A. Schneider, V. W. Chen, and G. Pardo-Castellote. ControlShell: A Real-Time Software Framework. In *Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service and Space*, volume II, pages 870–7, Houston, TX, March 1994. AIAA, AIAA.

[11] M. A. Ullman, R. H. Cannon, and S. M. Rock. A Modular System Architecture for Multi-Manipulator, Free-Flying Space Robots. In *Proceedings of the International Symposium on Intelligent Control*, pages 90–97, Chicago, IL, August 1993.

[12] H. H. Wang, R. L. Marks, T. W. McLain, S. D. Fleischer, D. W. Miles, G. A. Sapilewski, S. M. Rock, M. J. Lee, and R. C. Burton. OTTER: a testbed submersible for robotics research. In *Proceedings of the ANS 6th Topical Meeting on Robotics and Remote Systems*. American Nuclear Society, February 1995.

[13] H. H. Wang, R. L. Marks, S. M. Rock, and M. J. Lee. Task-Based Control Architecture for an Untethered, Unmanned Submersible. In *Proceedings of the 8th Annual Symposium of Unmanned Untethered Submersible Technology*, pages 137–147. Marine Systems Engineering Laboratory, Northeastern University, September 1993.