Design and Development of Command and Control System for Autonomous Underwater Vehicles

Tan Yew Teck, Mandar Anil Chitre, Prahlad Vadakkepat, Shiraz Shahabudeen

***Abstract*—Command and control for AUVs has been an area of active research over the past years. Inspired by the command structure in real ships and submarines, we have developed a command and control system which operates through the interaction of multiple software agents. The software agents take on roles such as Captain, Executive Officer, Navigator, Pilot, etc. to achieve specified missions. The command and control system has been tested in simulation and in field tests on the STARFISH AUV developed at the National University of Singapore. In this paper, we will present the command and control architecture and some field test results.**

***Index Terms*—Autonomous Underwater Vehicle (AUV), Command and Control System, Software Architecture, Hybrid Architecture, Modularity.**

I. Introduction

DESPITE substantial progress in Autonomous Underwater Vehicle (AUV) technologies over the last few years, the Command and Control (C2) system continues to challenge researchers. To carry out a mission, the C2 system must be robust, adaptive, and able to cope with the changes in dynamic and uncertain environments. The C2 system is a highly complex and critical software in a mission-based AUV. At a higher level, it is in charge of breaking down a mission into tasks, interpreting mission commands from the operator, making decisions, taking appropriate actions if a problem is encountered and ensuring the safety of the AUV throughout the mission. At a lower-level, the C2 system is capable of interpreting raw data coming from the AUV's sensors and commanding different actuators or low level control systems to generate the desired behavior in order to fulfil each mission task.

C2 system for AUV projects have been evolving over the years. During the early stages, C2 systems fell into one of the categories: reactive or deliberative, centralized or distributed, top-down or bottom-up. However, as AUV technology advances and the need for better functionality and capability arises in the AUV's working environment, a C2 system adopting only one of the architecture mentioned above is no longer able to handle complicated tasks in partially unknown environments. In order to solve this problem, majority of the C2 systems nowadays use a hybrid architecture. Hybrid architectures are constructed by the combination and/or integration of two or more different architectures to become a single system that takes advantages of each architecture while minimizing their individual weaknesses.

In this work, we have developed a C2 system with the hybrid architecture for a single modular AUV and it can be naturally extended to be used for a team of collaborative AUVs. The C2 system is flexible enough for changes to be made and extra functionalities to be added depending on the mission tasks assigned and the configuration of each individual AUV in the team. The C2 system is currently being used for AUV missions in a System-In-The-Loop simulator and tested on the STARFISH AUV. The STARFISH project is an initiative at the Acoustic Research Laboratory (ARL) of the National University of Singapore (NUS) to study collaborative missions carried out by a team of low-cost, modular AUVs. Simulation results showed good performance and reactivity on simple navigational tasks while preliminary lake test results of the first prototype AUV further validated the functionality of the developed C2 system.

The remainder of this paper is organized as follows. Section II illustrates the architectural overview of the component based C2 system. Section III and IV present the simulation results from the System-In-The-Loop simulator and the lake experiment. Finally, section V concludes the paper and discusses future work.

II. Architectural Overview

The review of literature revealed various control architectures implemented by different researchers in the field in which hybrid architecture is the most popular. A hybrid architecture is constructed by the combination of both deliberative and reactive architectures.

In the STARFISH project, we have developed a novel C2 system based on a hybrid hierarchical model as shown in Fig. 1. It adopts a deliberative-reactive architecture and consists of a set of interacting agent components arranged in hierarchical order to depict different level of command responsibilities. As proposed in literature [1], [3], [4], our architecture consist of three levels: Supervisory level, Mission level and Vehicle level. The Supervisory level is in charge of monitoring the high level mission and vehicle status as well as corresponding and sending the information to the operator/mothership. The Mission level is responsible for mission/tasks planning and finally, the Vehicle level carry out the mission tasks and perform obstacle avoidance by utilizing different Sentuators (sensors and actuators) to generate the desired maneuvering behaviors. A communication component (Signaling_Officer) also is designed to provide a communication link with the mothership/operator or with another AUV. Chart Room is the database where a map of the mission areas are stored while Mission Script consists of different mission files identified by their mission numbers.

Each agent component has its private data and implements its own algorithms depending on the assigned tasks. The vehicle's C2 tasks are achieved via the interaction and cooperation among the involved agent components. An

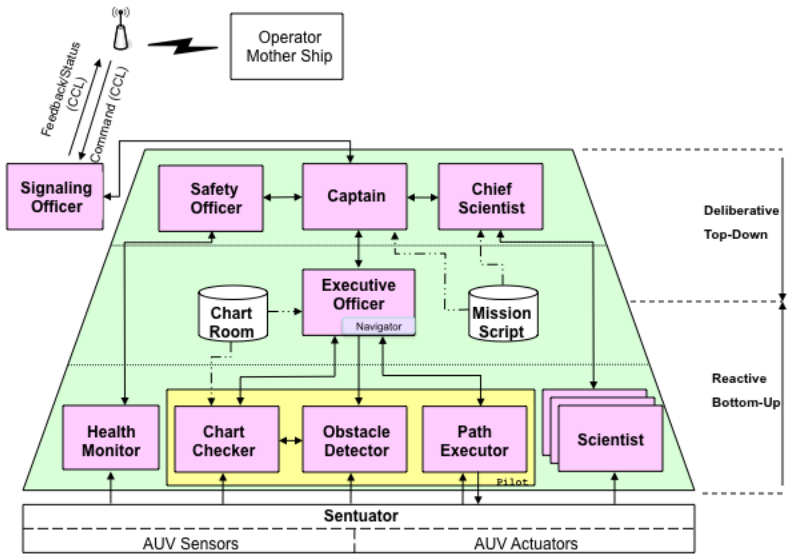


Fig. 1. Hybrid Control Architecture for the AUV.

agent component's internal activity is governed by a finite state machine which processes its tasks continuously depending on the current state of the component.

The following paragraphs give a detailed description of the responsibilities and tasks of different agent components:

A. ***Supervisory Level: Captain, Chief_Scientist and Safety_Officer***

There are three components under the Supervisory level: the Captain, the Chief_Scientist and the Safety_Officer. Components in this level carry out the main decision making with respect to the mission and vehicle safety. Any one of these components has the right to modify or abort a mission if found necessary.

The Captain component is in charge of the high level supervisory tasks. It starts, coordinates, oversees and controls the execution of all other components while keeping track of mission progress. In situations where the AUV encounters problems caused by software errors or hardware failures, the Captain determines the source of the problem and attempts to solve it. However, if the problem continues to exist, the current mission is aborted and the operator is notified. The decisions are made based on inputs from components within the C2 system and a simple rule-based system with knowledge represented as IF-THEN rules. For missions that involve multiple-AUVs, the Captain is also involved in cooperation and coordination among the AUVs.

The STARFISH AUV can have different payload sections added or exchanged to meet the requirements of mission or to perform underwater scientific experiments. The Chief_Scientist is responsible for command and control of payload sections. When the AUV is in the mission area, the Chief_Scientist enables the corresponding Scientist components and starts analyzing the obtained information. When necessary, the Chief_Scientist informs the Captain to modify its navigational plan, or to abort the current mission if it fails to perform the assigned tasks.

It is important to ensure an AUV's safety throughout mission execution. For an autonomous mission, the AUV must be able to detect any abnormality that might arise and take necessary steps to make sure that it is not lost during the mission. The Safety_Officer polls data regarding the health conditions of all the devices (sensors and actuators) from Health_Monitor components. It then analyzes the health condition of each device. Besides that, Safety_Officer also looks at sensor data to determine unsafe conditions and perform emergency abort if necessary. This provides a safeguard against agent component malfunction. However, whenever critical events such as a leak develop, the Safety_Officer will cut off the entire AUV's power without consulting the Captain and drop the ballast to prevent the hardware from being damaged.

B. ***Mission Level: Executive_Officer***

At the Mission level, Executive_Officer converts mission points to tasks, plans the task sequence and outputs the task commands as well as mission path for the mission execution. Whenever a mission number and START command are received from the Captain, the Executive_Officer reads the mission file to retrieve the task sequence, mission parameters as well as mission points. The retrieved mission points are then fed to the Navigator for mission path planning. If a feasible path is found between the start and target mission point, the resultant tasks are passed to vehicle level for navigation while the mission points and task sequence are reported back to Captain for mission monitoring. However, if the Navigator fail to find a path, the Captain is informed the operator is notified via Signaling_Officer. When there are changes in the mission points during mission execution, the mission path will be re-planned.

C. *Vehicle Level: Path_Executor, Obstacle_Detector, Chart_Checker, Scientist, Health_Monitor*

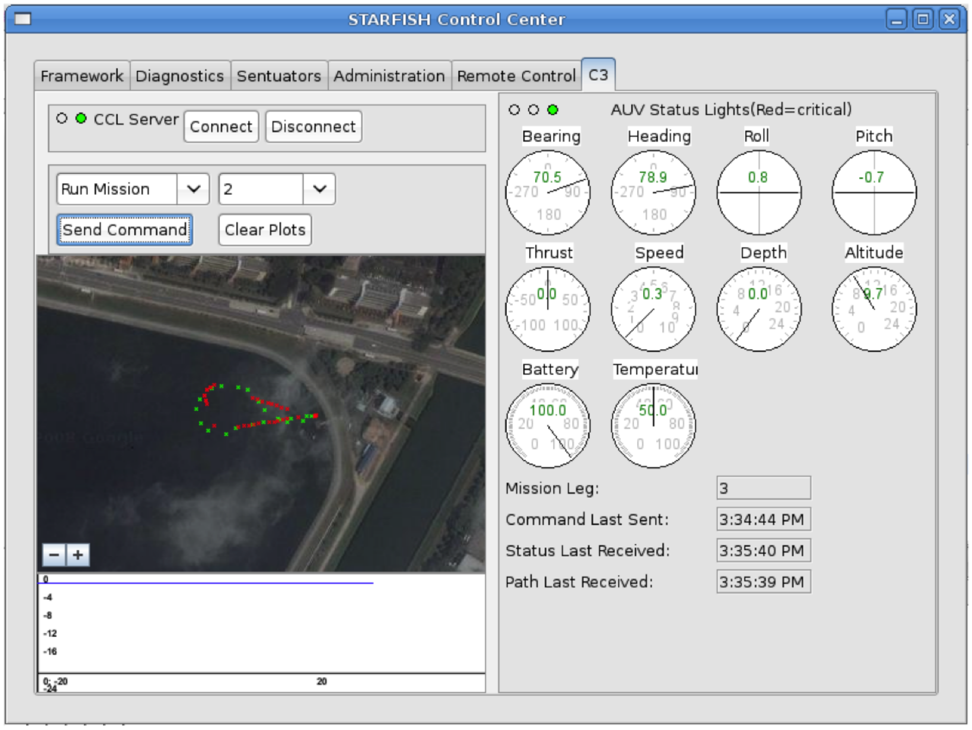
The Vehicle level consist of five components: Path_Executor, Obstacle_Detector, Chart_Checker, Scientist and Health_ Monitor. They are the reactive components that interact with the vehicle's sensory and actuator level - the Sentuator level. The control and processing at the vehicle level is distributed among its components. Every component has its own set of responsibilities and operates asynchronously based on the commands from the higher level of control hierarchy. Among the components at this level, Path_Executor, Obstacle_Detector and Chart_Checker together play the role of a pilot. They handle tasks ranging from translation of mission tasks into control signal, depth and position keeping, obstacle avoidance and mission chart updating. The Scientist component is responsible to interact with devices in payload section while the Health_Monitor component keeps track of the overall AUV's health condition.

1. Obstacle_Detector and Chart_Checker
   During mission execution, floating obstacles and sea floor are threat to the AUV's safety. Collision with any threats may jeopardize the mission as well as the AUV. Early detection of unknown obstacles lying along the AUV's path is crucial to make sure it has enough time and space to perform the avoidant maneuver. The Obstacle_Detector reads the data from Forward Looking Sonar (FLS) to determine the location of objects that exist along the AUV's mission path.
   The obstacle data are sent to the Chart_Checker which in turns checks for existence of the obstacles in the Chart Room. ChartRoom is the central storage place for the maps of the mission area where the AUVs are operating. Obstacles that are known in the Chart Room will be taken into account when planning mission path. Obstacles that do not exist in the Chart Room are marked and the corresponding location and depth are updated in the maps. Collision checking is then be performed by the Chart_Checker along the mission path. If any of the newly found obstacles lie in the mission path, the Chart_Checker modifies the mission path to make sure the AUV would not collide with the obstacle.
2. Path_Executor
   The Path_Executor is responsible for translating the high level mission tasks into vehicle's low level maneuver control. This component implements a library of basic functions that the vehicle can use to generate the desired maneuvers. One or more basic functions can be invoked concurrently to achieve a high level mission task. This is bottom-up approach where distributed simple vehicle behaviors can be merged to form complex maneuvers.
3. Scientist
   Scientist component is responsible for processing and analyzing the data obtained from payload sensors. To allow exchangeable payload sections, one Scientist component is built per payload section and loaded by the Chief_Scientist depending on the final setup of the AUV. More than one Scientist components can exist in a single AUV if there are several payload sections attached. They are coordinated and controlled by Chief_Scientist component throughout mission. Two payload sections are currently built for STARFISH project. They are the Advanced Navigational payload and Side Scan Sonar payload.
4. Health_Monitor
   Health_Monitor component keeps track of the health conditions for all the devices in the AUV including the optional payload section. This is particularly important to make sure the AUV is in its optimum working condition. Every hardware interfacing component (Sentuator component) in the AUV implements an internal health monitoring method which checks the health status of the hardware it attached to. This information is collected periodically by the Health_Monitor component for vehicle health status updating and forwarded to Safety_Officer for further action.

D. *External Communication: Signalling_Officer*

Communications with the mission operator/mothership or other AUVs is supported by the Signaling_Officer through a message-passing mechanism. Signaling_Officer acts as the AUV's external communication node, and is represented outside the overall control hierarchy. This component encodes and decodes the messages among AUVs or between AUV and operator. Besides that, Signaling_Officer is also responsible for updating the operator with the current mission and AUV status periodically. For the STARFISH AUV, the Common Control Language (CCL) is adopted as the message format for acoustic communication [5]. CCL is developed to establish a standard for communication between different agents during operation and provides support for interaction between machines and operators. A CCL message is a 32-byte data packet. The first Byte is used to specify the message mode (type) followed by optional data values.

During mission execution, different CCL messages are sent periodically to the operator for display on the GUI interface. Fig 2 shows a sample CCL message string with their corresponding representations and the screen capture of Command and Control GUI module during operation.



#C,1,0,1,0,1,5ffffc000000003238ffb500570061000364326403000000000000000000000000

Fig. 2. Screen capture of the C2 GUI interface and a sample CCL message.

More CCL messages will be defined in the future for multi-AUV scenarios.

E. *Benefits from the architectural design*

The resulting C2 system's architectural design offers many benefits. The hybrid modular-hierarchical control architecture adopts both top-down and bottom-up approach in its control structure. This allows deliberative high level mission control while decouples the low level reactive vehicle control. Moreover, the breaking down of C2 tasks into individual agent components presents an explicit view of the clearly defined control responsibilities at different level of control hierarchy.

The implementation of state machine in the component facilitates controllability and observability [2] in the control architecture. Both controllability and observability allow monitoring and controlling of the internal structure and behavior of agent components. This is particularly important in a C2 system where supervisory components at the high level control architecture can monitor and command the behavior of low level components.

The component-based design of the C2 system on top of the DSAAV [6] makes exchangeable components possible and provides flexibility in terms of software implementation and alternation. Instead of modifying the existing software components, new components with identical interfaces but different algorithms can be built and loaded when necessary. Besides that, the Scientist component can be configured to adapt to the AUV's final payload setup without affecting the overall control structure. This can be done easily by changing the entries in the configuration file.

Since the components are self-contained and the inter-component communication is carried out through message passing, the internal operation of the components do not interfere with each other. This provides fault tolerance if errors occur in one component, as they do not cause the whole C2 system to malfunction.

III. Simulation

The outlined C2 system has been developed and tested on STARFISH simulator - a 3D System-In-The-Loop (SITL) simulator that uses Open Dynamic Engine (ODE) [8]. SITL simulation has advantages in terms of system testing and deployment. The same source code or programs that are running in the SITL simulator can be directly used on the hardware platform without any alteration. This results in rapid and simplified system development. Besides that, the simulator also allows different underwater conditions like sea current to be simulated for testing.

The objective for simulation trials was to validate the C2 system architecture before final deployment on the AUV. During the simulation trials, the AUV was given a mission to dive, navigate though a few mission points at certain depths and finally, surface at the end point. The purpose of this mission is to 1)test the functionality of the components in the C2 system and 2)verify the overall C2 system performance in carrying out an assigned autonomous mission.

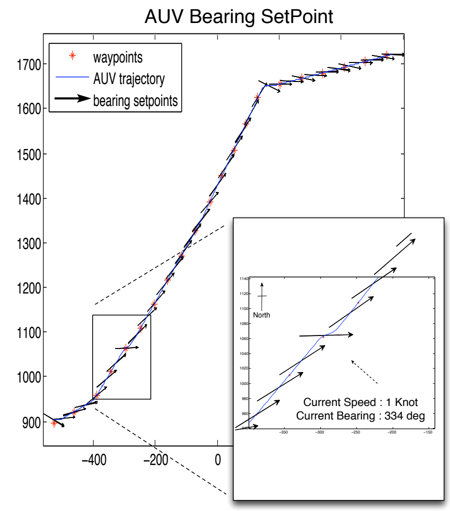


Fig. 3. Plot of AUV's trajectory and bearing setpoints.

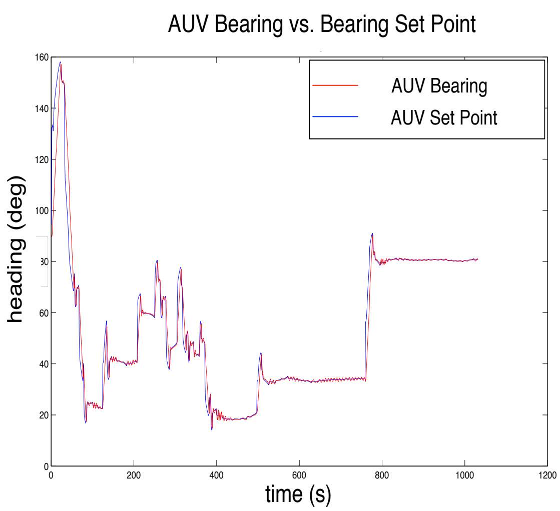


Fig. 4. Plot of reference and actual vehicle bearing during mission.

A. Simulation results

A mission has been carried out using simulated ocean terrain obtained from the Singapore coastal charts [7]. The simulator is subjected with sea current of 1 knot heading at 334° (North = 0°, clockwise) throughout the simulation. Figures 3 and 4 show the resulted AUV's trajectory and bearing of the AUV throughout the mission execution. The AUV traveled a distance of approximately 1000 meters, with depth ranging from 0 to 15 meters and average bearing error less than 5 degrees. As can be seen in the zoomed in area of Fig. 3, the bearing setpoints are made to compensate the sea current so that the vehicle exhibits path following behavior. The resulting mission path took into account of vehicle's maximum turning and pitching angle. All the waypoints were visited when the mission was completed. Although these are preliminary results obtained with a simulator with a simple mission, it is sufficient to verify the basic functionality of each component while confirming the integrity of the overall C2 system.

B. Field Trial

Several trials have been carried out at Pandan Reservoir (*Latitude* = 1.3171°, *Longitude* = 103.7482°, Fig. 5), Singapore. We present data from one surface trial. The trials were conducted using our test AUV. In the trial, the AUV is given a mission file to navigate through the mission points and stop once the mission is completed. During the surface mission (depth = 0), the AUV is subject to wind disturbance as well as north-pushing current caused by a water pump near the start location. The expected behavior is for the deliberative layer to receive operator's command and plan a path through the mission points, while for the reactive layer to maneuver the vehicle based on the mission path and abort the mission if any abnormity happen.

Fig. 5. (a) Plot of AUV's path for surface mission. The coordinates are based on raw GPS data whenever it is available, and rely on data from positioning system (PosSys) when GPS is not available. The AUV started from the floating platform and navigated through all mission points.

Fig. 5 shows the resulted trajectory (green circles) executed by the AUV. The AUV positioning is based on raw GPS data or approximated by the AUV's positioning system (PosSys) if GPS is not available. The velocity and heading is obtained from the DVL. Although the GPS lost its fix a few times during the surface mission, the AUV's positioning system managed to provide acceptable position data to complete the mission successfully. Fig. 6 shows path followed by AUV during the surface mission. Black arrows and blue line show the AUV's bearing set points and path executed. Red circles are the waypoint radius, the waypoint is considered reached when the AUV is within this area. From the surface mission, we observe that the AUV bearing is set slightly towards the southwest direction nearby the start location to compensate the water pump's induced velocity. Once it is out of the pump

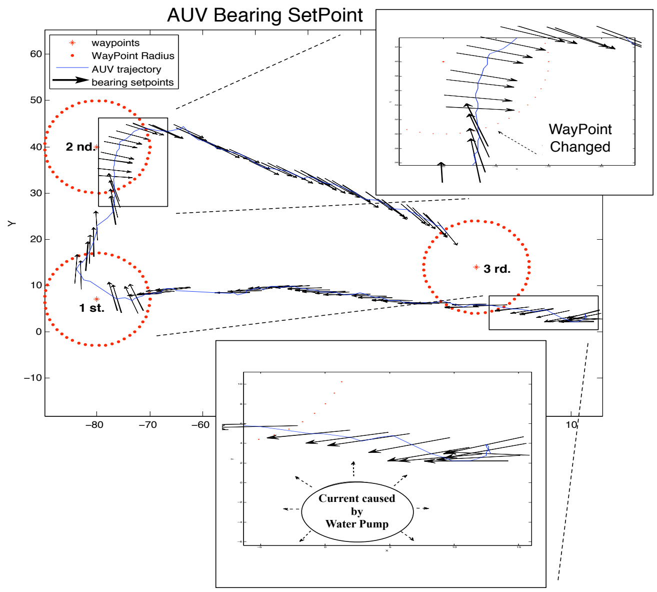


Fig. 6. Plot shows the AUV's bearing and bearing setpoints through the mission execution.

area, the set points were pointed directly towards the first mission point. This demonstrated the path following behavior implemented in the reactive layer. Also, it can be seen that the bearing setpoints changed to point to the next waypoint whenever the AUV is within the waypoint radius. Despite the simplified navigation missions, the C2 system has shown expect behavior. We expect to further validate the overall C2 system functionality in our future trials.

IV. Conclusion

A novel AUV command and control system architecture has been developed. The focus has been to develop a generic control and software architecture for a single AUV and later, expendable to multi-AUV operations.

The design, testing and validation of the architecture has been described. The proposed control architecture has a hybrid structure. It contains a set of interacting agent components and are grouped into three hierarchical control layers that enables the mission supervisory and command to be executed at higher layer while decouples the vehicle and navigational control from the lower layer. It also provides capabilities for real time mission status updates and vehicle or mission error detection. The control architecture also allows multiple Cheif_Scientist components to be added to adapt to the AUV's final setup without affecting the overall control structure.

The C2 system has been developed and tested in a software simulator. It is also currently operational in a STARFISH AUV and has been tested in initial field trials.

Future work includes refining the current C2 system through further field tests and implementing better localization and mapping algorithm in Chief_Scientist component when Side Scan Sonar is available. When the second STARFISH AUV is ready, the C2 system will be expanded to handle multi-AUV missions.

References

[1] H. Yavuz, A.B., *A New Conceptual Approach to the Design of Hybrid Control Architecture for Autonomous Mobile Robots.* Journal of Intelligent and Robotic System, 2002. 34: p. 1-26.

[2] Antonio C. Dominguez-Brito, Daniel h.S., Jose Isern-Gonzalez, Jorge Cabrera-Gamez., *CoolBOT: A Component Model and Software Infrastructure for Robotics. Software Engineering for Experimental Robotics.* Vol. 30/2007: Springer Berlin.

[3] M.Carreras, N. Palomeras, P. Ridao and D. Ribas, *Design of a mission control system for an AUV.* International Journal of Control. July 2007. Vol. 80. No. 7. p. 993-1007

[4] J. Evans, P. Patron, B. Smith and D.M. Lane. *Design and evaluation fo a reactive and deliberative collision avoidance and escape architecture for autonomous robots.* Springer.

[5] Eugene Eberbach, Christine N. Duarte, Christine Buzzell, Gerald R. Martel, *A Portable Language for Control of Multiple Autonomous Vehicles and Distributed Problem Solving*, Proc. of the 2nd Intern. Conf. on Computational Intelligence, Robotics and Autonomous System, CIRA'03, Singapore, Dec. 15-18, 2003.

[6] Mandar Chitre. *DSAAV - A Distributed Software Architecture for Autonomous Vehicles.* (Presented at OCEANS 2008. MTS/IEEE Quebee.)

[7] *Charts for Small Craft . Singapore Strait and Adjacent Waterways.* Maritime and Port Authority of Singapore (MPA), 2004.

[8] "Object Dynamic Engine," available at: http://www.ode.org/. Accessed 1 August 2008.