In Proceedings of the Symposium on Interactive 3D Graphics, ACM, April 1995

Underwater Vehicle Control from a Virtual Environment Interface

Stephen D. Fleischer and Stephen M. Rock
Stanford Aerospace Robotics Laboratory

Michael J. Lee
Monterey Bay Aquarium Research Institute

Abstract

This paper describes a collaborative research effort initiated by Monterey Bay Aquarium Research Institute (MBARI), Stanford Aerospace Robotics Laboratory (ARL), and NASA Ames Research Center. The goal of this joint effort was to develop an experimental system which demonstrates real-time supervisory control of an underwater vehicle from an interactive, 3-D graphical interface.

Introduction

OTTER (Oceanographic Technologies Testbed for Experimental Research) is a remotely-operated underwater vehicle jointly constructed by MBARI and ARL. Recently, our research focus has been the creation of a 3-D graphical interface to control OTTER. To accomplish this task, the NASA Ames Virtual Environment Vehicle Interface (VEVI) was chosen as a baseline and extensively modified for use with the underwater vehicle.

This research was divided into two separate objectives. The first objective was to extend the capability of the current X Window-based graphical user interface (GUI) for OTTER, by taking advantage of the current virtual reality (VR) technologies available from NASA Ames. This new GUI is capable of providing the user with better visualization of the underwater environment and improved control of the vehicle.

Our second goal was to demonstrate task-level position control of OTTER from the new virtual environment interface. Specifically, the user should be able to control the vehicle along a desired trajectory by specifying a number of via points along the path. This requires successful integration of the graphical interface into the vehicle control system hierarchy.

Stephen D. Fleischer and Stephen M. Rock
Durand Bldg., Stanford University, Stanford, CA 94305
fleisch, rock@sun-valley.Stanford.EDU

Michael J. Lee
MBARI, 160 Central Avenue, Pacific Grove, CA 93950
lemi@mbari.org

VEVI Structure

In the past, VEVI has been used to control other robotic vehicles of all types with great success. Some examples include the NASA Ames TROV underwater vehicle, which has explored the waters of the Antarctic; the ARL space robots, which simulate the zero-gravity of space in two dimensions; and most recently, the CMU Dante II eight-legged walking vehicle, which explored the Mt. Spurr volcanic crater in Alaska. [1]

Figure 1 shows the current structure of the VEVI software, as implemented in the OTTER control architecture. The core of VEVI is the Renderer, which was written on top of the WorldToolKit (WTK) world simulation library. The WorldToolKit library, developed by the Sense8 Corporation, enables programmers to graphically simulate an environment, including the universe model and any movable objects within that universe. The Renderer has the capability to interact with novel virtual reality devices, such as stereo or head-mounted displays, flying mice, 6-DOF spaceballs, and head-trackers.

In order to communicate with the connected vehicle, VEVI transfers data through shared memory to the VehicleNode, which then communicates to the vehicle over a network. The NDDS network protocol, developed by students in ARL at Stanford University, [2] provided the communications interface between VEVI and OTTER.

The VeviNode provides VEVI with the ability to support multiple users across a network. The Renderer talks to the VeviNode through shared memory, which then broadcasts information to other copies of VEVI

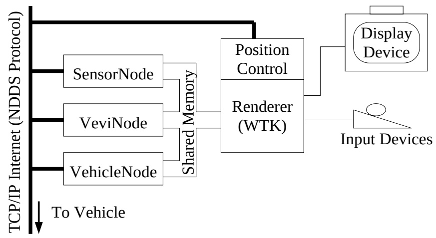

Figure 1: **VEVI Structure**

running on the network.

The SensorNode is only used if there are specialized sensors which cannot be accessed through the VehicleNode. All sensors on OTTER, including the thrusters, cameras, and acoustic positioning system, are accessed directly through the VehicleNode.

Implementation on OTTER

In our attempt to achieve position control from the virtual environment interface, we were able to take advantage of the OTTER Task-Level Control architecture. [3] As seen in Figure 2, this paradigm divides the vehicle control system into three levels, which can each be implemented on separate computers. The lowest (servo) level includes the real-time control loops, which are implemented in the computers on-board the underwater vehicle. Task commands are sent from the organizational level, which are then decomposed into smaller tasks by the middle (task) level and sent to the servo level for execution.

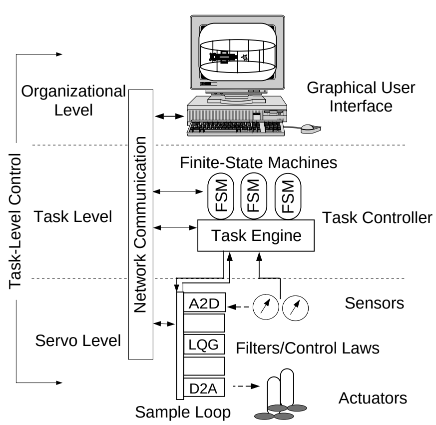

Figure 2: **OTTER Task-Level Control Structure**

This methodology isolates the graphical user interface, which encompasses the entire organizational level, from the lower levels of the control hierarchy. Conceptually, task-level control enables the user to perform complex functions (e.g. driving a vehicle along a desired path) by combining lower-level tasks which can be performed autonomously by the vehicle. After implementing a simple point-to-point transect as a task for the vehicle, we added a position control module to VEVI. With this module, the user is able to graphically specify a series of via points along a path by dragging around a ghost image of OTTER in the virtual environment. These via points are then translated into task commands which are sent to the vehicle.

Conclusions

We have performed several demonstrations of task-level position control from the virtual environment interface in the Naval Postgraduate School (NPS) fresh-water test tank. Currently, VEVI runs in single-user mode, with a standard SGI color monitor for display and a standard 2-DOF mouse for user input.

By developing this operational platform for experimental research, we plan to pursue fundamental research in the development of interactive, 3-D virtual environment interfaces to control complex, real-time robotic systems. In terms of the OTTER project, we believe that continued research will encourage teams of marine research scientists to work with the underwater vehicle remotely, while enabling them to visualize the environment and relevant data in real-time.

In essence, we hope to enable the end-user to become more effective in performing a variety of tasks by maintaining a simple, intuitive interface to an inherently complex system.

References

[1] Fong, T. A Computational Architecture for Semi-autonomous Robotic Vehicles. In *Proceedings of AIAA Computing in Aerospace 9 Conference* (San Diego, CA, October 1993), AIAA.

[2] Pardo-Castellote, G., and Schneider, S. A. The Network Data Delivery Service: Real-Time Data Connectivity for Distributed Control Applications. In *Proceedings of the International Conference on Robotics and Automation* (San Diego, CA, May 1994), IEEE, IEEE Computer Society.

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