NAVAL POSTGRADUATE SCHOOL
Monterey, California

DISSERTATION

A VIRTUAL WORLD
FOR AN
AUTONOMOUS UNDERWATER VEHICLE

Donald P. Brutzman

December 1994

Dissertation Supervisor: Michael J. Zyda

A VIRTUAL WORLD FOR AN AUTONOMOUS UNDERWATER VEHICLE

Donald P. Brutzman
B.S.E.E., U.S. Naval Academy, 1978
M.S., Naval Postgraduate School, 1992

A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive.

An underwater virtual world can comprehensively model all salient functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot, enabling realistic AUV evaluation and testing in the laboratory. Three-dimensional real-time computer graphics are our window into that virtual world.

Visualization of robot interactions within a virtual world permits sophisticated analyses of robot performance that are otherwise unavailable. Sonar visualization permits researchers to accurately "look over the robot's shoulder" or even "see through the robot's eyes" to intuitively understand sensor-environment interactions. Extending the theoretical derivation of a set of six-degree-of-freedom hydrodynamics equations has provided a fully general physics-based model capable of producing highly non-linear yet experimentally-verifiable response in real time.

Distribution of underwater virtual world components enables scalability and real-time response. The IEEE Distributed Interactive Simulation (DIS) protocol is used for compatible live interaction with other virtual worlds. Network connections allow remote access, demonstrated via Multicast Backbone (MBone) audio and video collaboration with researchers at remote locations. Integrating the World-Wide Web allows rapid access to resources distributed across the Internet.

This dissertation presents the frontier of 3D real-time graphics to support underwater robotics, scientific ocean exploration, sonar visualization and worldwide collaboration.

Doctor of Philosophy in Computer Science
December 1994

Supervisor: Michael J. Zyda
Department of Computer Science

Classification of Dissertation:
UNCLASSIFIED

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY	2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Ph.D. Dissertation

4. TITLE AND SUBTITLE **A VIRTUAL WORLD FOR AN AUTONOMOUS UNDERWATER VEHICLE**	5. FUNDING NUMBERS
6. AUTHOR Donald P. Brutzman	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT

A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive.

An underwater virtual world can comprehensively model all salient functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot, enabling realistic AUV evaluation and testing in the laboratory. Three-dimensional real-time computer graphics are our window into that virtual world.

Visualization of robot interactions within a virtual world permits sophisticated analyses of robot performance that are otherwise unavailable. Sonar visualization permits researchers to accurately "look over the robot's shoulder" or even "see through the robot's eyes" to intuitively understand sensor-environment interactions. Extending the theoretical derivation of a set of six-degree-of-freedom hydrodynamics equations has provided a fully general physics-based model capable of producing highly non-linear yet experimentally-verifiable response in real time.

Distribution of underwater virtual world components enables scalability and real-time response. The IEEE Distributed Interactive Simulation (DIS) protocol is used for compatible live interaction with other virtual worlds. Network connections allow remote access, demonstrated via Multicast Backbone (MBone) audio and video collaboration with researchers at remote locations. Integrating the World-Wide Web allows rapid access to resources distributed across the Internet.

14. SUBJECT TERMS Virtual worlds, autonomous underwater vehicles, robotics, computer graphics, networking, hydrodynamics, real time, artificial intelligence, control systems, sonar, scientific visualization.	15. NUMBER OF PAGES 291
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18

A VIRTUAL WORLD FOR AN AUTONOMOUS UNDERWATER VEHICLE

by

Donald P. Brutzman
B.S.E.E., U.S. Naval Academy, 1978
M.S., Naval Postgraduate School, 1992

Submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in Computer Science

from the

NAVAL POSTGRADUATE SCHOOL
December 1994

Author: ______________________________
Donald P. Brutzman

Approved by: ______________________________
Michael J. Zyda, Professor of Computer Science

____
Robert. B. McGhee
Professor of Computer Science

____
Anthony J. Healey
Professor of Mechanical Engineering

____
Michael P. Bailey, Associate
Professor of Operations Research

____
Man-Tak Shing, Associate
Professor of Computer Science

Approved by: ______________________________
Ted Lewis, Chair, Department of Computer Science

Approved by: ______________________________
Richard S. Elster, Dean of Instruction

ABSTRACT

A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive.

An underwater virtual world can comprehensively model all salient functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot, enabling realistic AUV evaluation and testing in the laboratory. Three-dimensional real-time computer graphics are our window into that virtual world.

Visualization of robot interactions within a virtual world permits sophisticated analyses of robot performance that are otherwise unavailable. Sonar visualization permits researchers to accurately "look over the robot's shoulder" or even "see through the robot's eyes" to intuitively understand sensor-environment interactions. Extending the theoretical derivation of a set of six-degree-of-freedom hydrodynamics equations has provided a fully general physics-based model capable of producing highly non-linear yet experimentally-verifiable response in real time.

Distribution of underwater virtual world components enables scalability and real-time response. The IEEE Distributed Interactive Simulation (DIS) protocol is used for compatible live interaction with other virtual worlds. Network connections allow remote access, demonstrated via Multicast Backbone (MBone) audio and video collaboration with researchers at remote locations. Integrating the World-Wide Web allows rapid access to resources distributed across the Internet.

This dissertation presents the frontier of 3D real-time graphics to support underwater robotics, scientific ocean exploration, sonar visualization and worldwide collaboration.

ACKNOWLEDGEMENTS

Many people helped in this work. Mike Zyda is the best dissertation advisor anyone might hope for. His insight, support and enthusiasm are boundless. Bob McGhee and Tony Healey showed unlimited patience and insight as we explored the frontiers of dynamics modeling. Mike Bailey taught me analytical and discrete event simulation. He and Man-Tak Shing also gave valuable advice on the Ph.D. process. Mike Macedonia's unparalleled understanding of computer networks helped make an entire field intelligible.

Dave Pratt blazed the trail with NPSNET, still the best virtual world around and still gaining on all the others. Dave provided crucial academic advice and also the financial support which made the *SIGGRAPH 94* exhibit at *The Edge* possible. I am indebted to everyone who helped make that weeklong demonstration possible, especially Shirley Isakari Pratt, John Roesli, Frank Tipton, Jim Vaglia, Matt Johnson, Chris Stapleton, Garry Paxinos, Jacki Ford Morie, Theresa-Marie Rhyne, Paul Barham, John Locke, Steve Zeswitz, Rosalie Johnson, Russ and Sue Whalen, Walt Landaker, Dave Marco, Mike Williams, Terry Williams, Hank Hankins and Hollis Berry. I also thank Richard Hamming, Peter Purdue, Gordon Bradley, Jim Eagle, Ted Lewis, Mike McCann, Bruce Gritton, Mike Lee, David Warren, Dave Norman, John Sanders, Dick Blidberg, SeHung Kwak, Ron Byrnes, Drew Bennett, Jim Bales, Jim Bellingham, Alan Beam, Claude Brancart, Rodney Luck, John Gambrino, and Larry Ziomek for their help. Support for this research was provided in part by the National Science Foundation under Grant BCS-9306252 to the Naval Postgraduate School.

This work is dedicated with love and thanks to my wife Terri and our children Hilary, Rebecca, Sarah and Patrick.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. A VIRTUAL WORLD FOR AN AUTONOMOUS UNDERWATER VEHICLE

A. INTRODUCTION

A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive. An underwater virtual world can comprehensively model all necessary functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot, enabling realistic AUV evaluation and testing in the laboratory. 3D real-time graphics are our window into the virtual world. A networked architecture enables multiple world components to operate collectively in real time, and also permits world-wide observation and collaboration with other scientists interested in the robot and virtual world. This architecture was first proposed in (Brutzman 92d).

This dissertation develops and describes the software architecture of an underwater virtual world for an autonomous underwater robot. Multiple component models provide interactive real-time response for robot and human users. Theoretical development stresses a scalable distributed network approach, interoperability between models, physics-based reproduction of real-world response, and compatibility with open systems approaches. Implementation of the underwater virtual world and autonomous underwater robot are documented in a companion software reference (Brutzman 94e).

B. MOTIVATION

Underwater robots are normally called Autonomous Underwater Vehicles (AUVs), not because they are intended to carry people but rather because they are designed to intelligently and independently convey sensors and payloads. AUVs must accomplish complex tasks and diverse missions while maintaining stable physical control with six spatial degrees of freedom. Little or no communication with distant human supervisors is possible. When compared to indoor, ground, airborne or space environments, the underwater domain typically imposes the most restrictive physical control and sensor limitations upon a robot. Underwater robot design requirements therefore motivate this examination. Considerations and conclusions remain pertinent as worst-case examples relative to other environments.

A large gap exists between the projections of theory and the actual practice of underwater robot design. Despite a large number of remotely operated submersibles and a rich field of autonomous robot research results (Iyengar 90a, 90b), few AUVs exist and their capabilities are limited. Cost, inaccessibility and scope of AUV design restrict the number and reach of players involved. Interactions and interdependencies between hardware and software component problems are poorly understood. Testing is difficult, tedious, infrequent and potentially hazardous. Meaningful evaluation of results is hampered by overall problem complexity, sensor inadequacies and human inability to directly observe the robot *in situ*. Potential loss of an autonomous underwater robot is generally intolerable due to tremendous investment in time and resources, likelihood that any failure will become catastrophic and difficulty of recovery.

Underwater robot progress has been slow and painstaking for many reasons. By necessity most research is performed piecemeal and incrementally. For example, a narrow problem might be identified as suitable for solution by a particular artificial intelligence (AI) paradigm and then examined in great detail. Conjectures and theories are used to create an implementation which is tested by building a model or simulation specifically suited to the problem in question. Test success or failure is used to interpret validity of conclusions. Unfortunately, integration of the design process or even final results into a working robot is often difficult or impossible. Lack of integrated testing prevents complete verification of conclusions.

AUV design must provide autonomy, stability and reliability with little tolerance for error. Control systems require particular attention since closed-form solutions for many hydrodynamics control issues are unknown. In addition, AI methodologies are essential for many critical robot software components, but the interaction complexity and emergent behavior of multiple interacting AI processes is poorly understood, rarely tested and impossible to formally specify (Shank 91). Better approaches are needed to support coordinated research, design and implementation of underwater robots.

Despite these many handicaps, the numerous challenges of operating in the underwater environment force designers to build robots that are truly robust, autonomous, mobile and stable. This fits well with a motivating philosophy of Hans Moravec (Moravec 83, 88):

> .. solving the day to day problems of developing a mobile organism steers one in the direction of general intelligence... Mobile robotics may or may not be the fastest way to arrive at general human competence in machines, but I believe it is one of the surest roads. (Moravec 83)

C. OBJECTIVES

This dissertation addresses the following research questions:

- What is the software architecture required to build an underwater virtual world for an autonomous underwater vehicle?
- How can an underwater robot be connected to a virtual world so seamlessly that operation in the real world or a virtual world is transparent to the robot?
- What previous work in robotics, simulation, 3D interactive computer graphics, hydrodynamics, networking and sonar visualization are pertinent to construction of an underwater virtual world?
- What are the functional specifications of a prototypical AUV, and what are the functional specifications of robot interactions with the surrounding environment?
- How can 3D real-time interactive computer graphics support wide-scale general access to virtual worlds? Specifically, how can computer graphics be used to build windows into an underwater virtual world that are responsive, accurate, distributable, represent objects in openly standardized formats, and provide portability to multiple computer architectures?
- What is the structure and derivation for an accurate six degree-of-freedom underwater rigid body hydrodynamics model? The model must precisely reproduce vehicle physical response in real time, while responding to modeled ocean currents and control orders from the vehicle itself. The hydrodynamics model must be general, verifiable, parameterizable for other vehicles, and suitable for distributed simulation. Such a model is highly complex due to multiple interacting effects coupled between all six degrees of freedom.
- What are the principal network software components needed to build a virtual world that can scale up to very large numbers of interacting models, datasets, information streams and users? How can these network components provide interactive real-time response for multiple low- and high-bandwidth information streams over local and global communications networks?
- Sonar is the most effective detection sensor used by underwater vehicles. Sonar parameters pertinent to visualization and rendering include sound speed profile (SSP), highly-variable sound wave path propagation, and sound pressure level (SPL) attenuation. How can a general sonar model be networked to provide real-time response despite high computational complexity? How can scientific visualization techniques be applied to outputs of the sonar model to render numerous interacting physical effects varying in three spatial dimensions and time?
- How can these concepts be implemented in a working system?

D. DISSERTATION ORGANIZATION

The real world is a big place. Virtual worlds must also be comprehensive and diverse if they are to permit credible reproductions of real world behavior. A variety of architectural components are described in this dissertation. Ways to scale up and arbitrarily extend the underwater virtual world to include very large numbers of users, models and information resources are included throughout.

Chapter II reviews related work in underwater robotics, robotics simulation, underwater vehicle hydrodynamics, robot simulation, computer networking, and scientific visualization of sonar models. Chapter III provides precise problem statements and solution overviews, both for the general dissertation topic as well as individual virtual world components. Chapter IV presents the functional characteristics of the NPS AUV, the underwater robot which has been networked with the underwater virtual world. Chapter V describes the requirements and design decisions made in building an object-oriented real-time interactive 3D computer graphics viewer. Chapter VI derives novel extensions to an underwater vehicle hydrodynamics model which permit real-time networked response, standardized nomenclature, suitability for parameterized use by other underwater vehicles, and correctness both in cruise and hover modes. Chapter VII identifies and examines the four network capabilities necessary for scalable and globally distributable virtual worlds. Network considerations include both tight and loose temporal coupling, low-bandwidth and high-bandwidth information streams, audio, video, graphics, multimedia, posture updates using the Distributed Interactive Simulation (DIS) protocol, and very large numbers of connecting models and users. Chapter VIII outlines a general sonar model, presents an example geometric sonar model, and describes how scientific visualization techniques might be applied to render the large set of important characteristic values which describe sonar behavior. Chapter IX presents experimental results for the hydrodynamics model and network performance during distributed exercises. Chapter X summarizes the many dissertation conclusions identified in preceding chapters. An acronym appendix is provided for reader convenience. Finally an accompanying video appendix documents performance of the NPS AUV operating in the underwater virtual world and presents a variety of exercise scenarios.

The structure of the accompanying software reference (Brutzman 94e) parallels the organization of this dissertation. All source code, support files and compiled executable programs are also freely available via Internet access using anonymous file transfer protocol (ftp) access. The software reference includes help files and source code for archive installation, the NPS AUV robot execution level, 3D computer graphics viewer, hydrodynamics, sonar modeling, networking, and use of the World-Wide Web (WWW) and Multicast Backbone (MBone).

II. REVIEW OF RELATED WORK

A. INTRODUCTION

This chapter reviews previous and current research pertinent to the creation of an underwater virtual world for an AUV. While no other underwater virtual worlds were encountered during this literature search, the diversity of the many components developed in this dissertation invite background examinations on a wide range of topics. Subjects examined in this chapter include underwater robotics, robotics and simulation, dynamics, networked virtual world communications, sonar modeling and visualization, and ongoing and future projects. In order to avoid becoming open-ended surveys of entire bodies of scientific literature, the following project reviews are limited to aspects directly pertaining to this dissertation.

B. UNDERWATER ROBOTICS

The AUV research community is small but steadily growing. Key papers in this field are primarily found in annual conferences (included throughout the accompanying list of references) which reach back over a decade. These include the IEEE Oceanic Engineering Society (OES) *Autonomous Underwater Vehicle (AUV)* symposia and *OCEANS* conferences, *Unmanned Untethered Submersible Technologies (UUST)* symposia, and *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments* workshops. A recent survey of previously unknown research submersibles and undersea technologies in Ukraine and Russia appears in (WTEC 93). Current capabilities in remotely operated vehicle (ROV) operations are described in (Newman 92-93). A survey of AUV capabilities emphasizing potential for commercial deployment appears in (Walsh 93-94). A detailed description of the NPS AUV appears in Chapter IV. This section provides an overview of several significant AUVs. For a dynamic view of underwater robotics, video segments of state-of-the-art AUV operations appear in recent video conference proceedings (Brutzman 93a, 94a).

A survey of all AUVs is not appropriate, but representative and pertinent AUV projects are summarized below.

1. ARPA/Navy Unmanned Undersea Vehicle (UUV)

The ARPA UUV program began in 1988 when the Charles Stark Draper Laboratories were contracted to build two large UUVs for tactical naval missions,

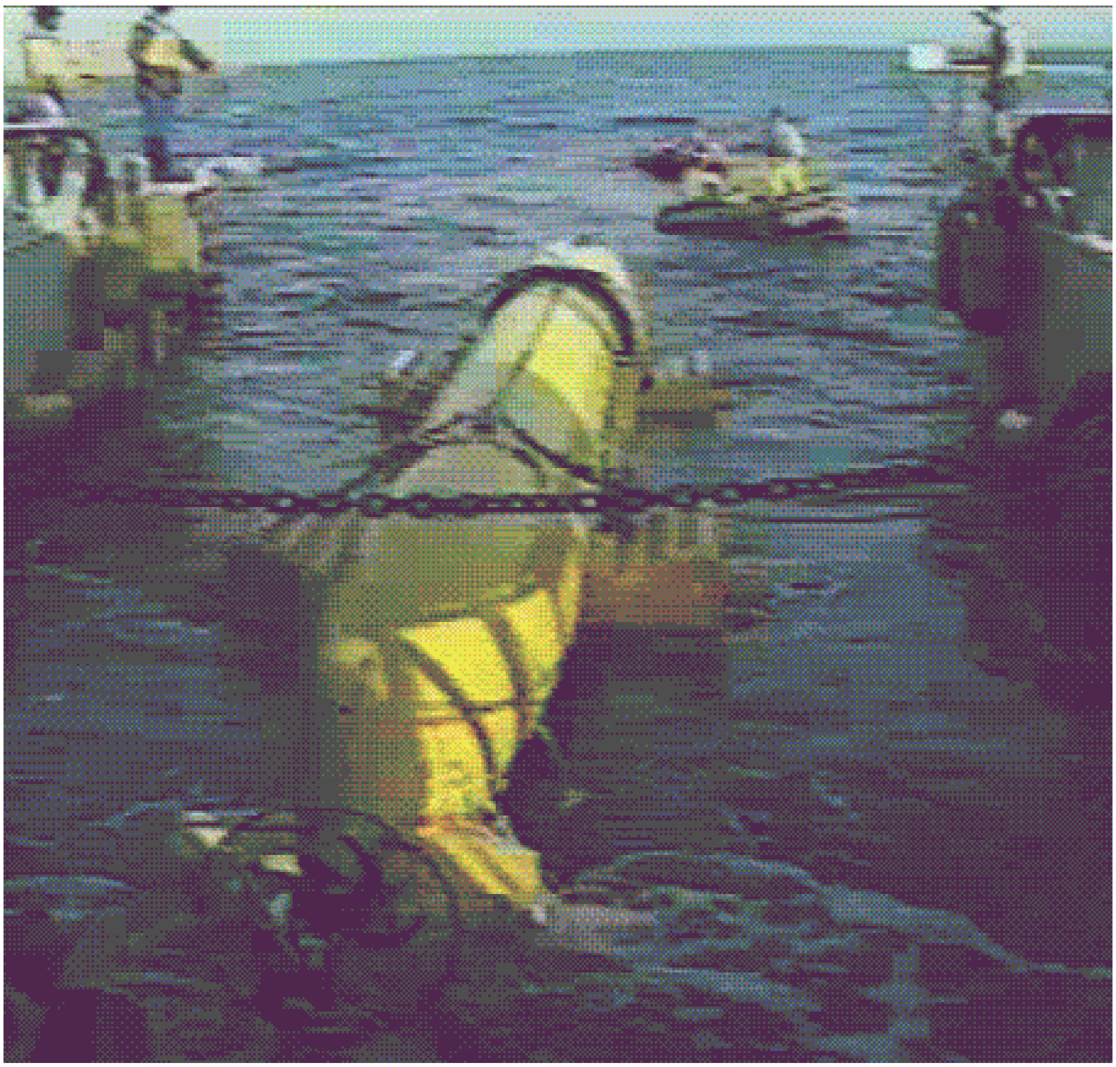

Figure 2.1. ARPA/Navy Unmanned Underwater Vehicle (UUV) being readied for launch during mission trials (Brancart 94) (Brutzman 94a).

particularly open-ocean minefield search. These vehicles are the largest, the most capable and (at approximately $9 million total) the most expensive AUVs built to date. The ARPA UUVs use high-density silver zinc batteries for 24 hours of operational endurance at 5-10 knots submerged. Weighing 15,000 pounds in air, the vehicles have titanium hulls which permit a test depth of 1,000 feet. The UUVs successfully deployed advanced sonar processors, laser communications and a variety of other advanced technologies in its 2000-pound-capacity payload section. Hybrid simulation techniques were used to test vehicle hardware and software prior to at-sea deployment. Simulation components included six-degree-of-freedom hydrodynamics and tether dynamics models, along with hardware subcomponent models and wireframe computer graphics. Vehicle overviews can be found in (Pappas 91) (Brancart 94), and extensive video footage of various testing milestones is included in (Brutzman 93a, 94a).

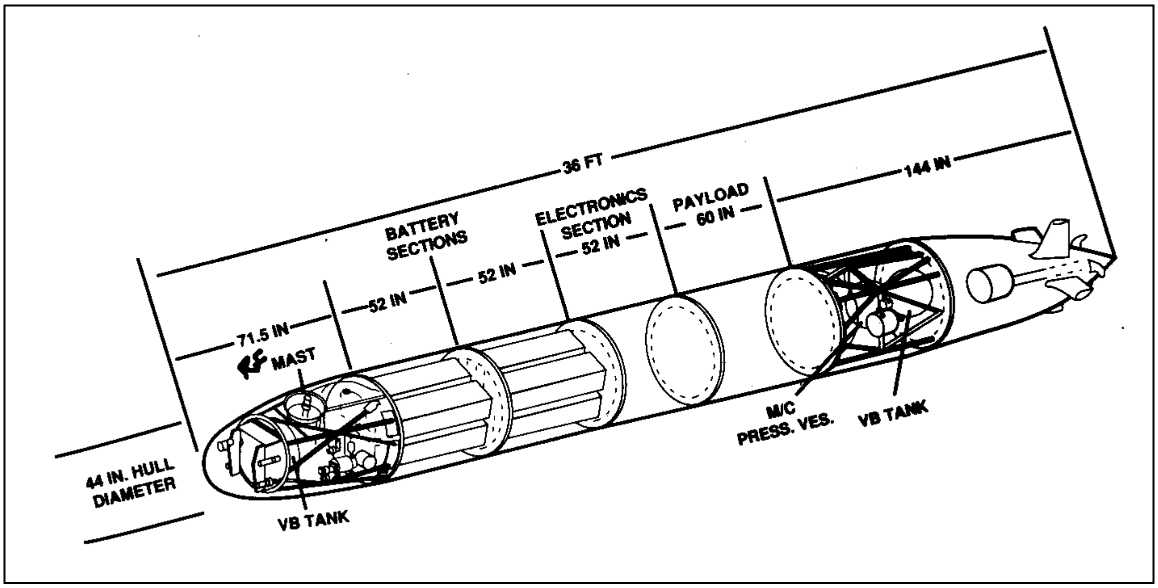

Figure 2.2. ARPA/Navy Unmanned Underwater Vehicle (UUV) internal layout (Pappas 91).

> "In March 1993, the [ARPA] Maritime Systems Technology Office successfully completed a series of at-sea tests that demonstrated the Mine Search System (MSS), a prototype minehunting system. In these demonstrations, a ship with the UUV in the lead repeatedly made safe transits through deep and shallow

mine fields. During these transits, bottom mines undetectable by ship-mounted sensors were readily detected by the UUV sensors optimally positioned with respect to the target mines... These demonstrations clearly showed for the first time the value of UUV sensors in a mine countermeasures role."
(from Brancart's ARPA abstract, Brutzman 94a)

The ARPA UUVs have been first to accomplish many important AUV tasks, but their cost is high and technical details remain out of the published literature. While they have been an excellent testbed for new technologies, the high cost of vehicle support and operations places them beyond the reach of most research efforts.

2. Massachusetts Institute of Technology (MIT) *Odyssey* Class AUVs

The MIT Underwater Vehicles Laboratory Sea Grant College Program has been building and deploying a series of low-cost AUVs for a number of years (Bellingham 94) (Fricke 94). The current *Odyssey II*, predecessor *Odyssey I* and

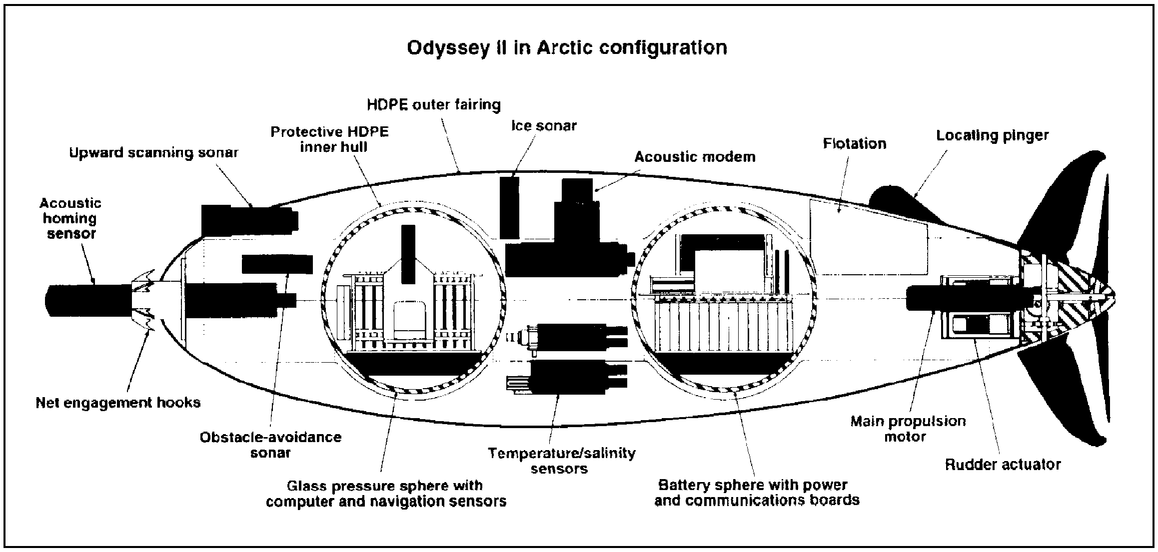

Figure 2.3. MIT *Odyssey II* in under-ice configuration. Deep-ocean configuration includes obstacle avoidance sonar, strobe light, altimeter sonar and video camera (Bellingham 94).

Sea Squirt vehicles are characterized by teardrop hull forms, 17" glass sphere internal pressure vessels, low power consumption, single 68030 microprocessor, single propeller and cruciform stern fin control. Vehicle control software uses state-configured layered control (Bellingham 90), an augmented form of subsumptive control (Brooks 86, 90) which provides a higher level of control in order to enable mission configuration. *Odyssey II* sensors include various scanning and homing sonars, depth sensor, temperature salinity and related sensors, video, inertial sensors, acoustic modem and acoustic navigation tracking pingers. Stable dynamic control is constrained by a minimum forward speed of 0.5 m/sec, and operational missions follow a cruise or survey profile. Maximum operating depth is 6,700 m. Unit costs remain low (under $75,000) even while each generation of vehicle has demonstrated significantly improved hardware and increased operational capabilities. Perhaps the greatest contributions of the *Odyssey* class vehicles have been in demonstrating operational missions in rivers, in open ocean and under the Arctic ice (Bellingham 94) (Fricke 94) (Brutzman 94a). Future work includes a variety of oceanographic missions using innovative sensors (Bales 94a, 94b) and ocean survey communications as part of a proposed Autonomous Oceanographic Sampling Network (AOSN) (Curtin 93) (Catipovic 93).

3. Marine Systems Engineering Laboratory *EAVE* Vehicles

The Experimental Autonomous Vehicle (EAVE) class of AUVs first started in 1978 when *EAVE I* demonstrated autonomous underwater pipe following (Blidberg 90) (Chappell 94). Subsequent missions have included navigation using acoustic transponders, submerged structure cleaning, underwater docking and parking, and multiple AUV submerged communication and mission coordination. *EAVE* class vehicles are constructed on open frames using large watertight cans for electronics and electrical components. Sensors include a variety of sonars, compass, temperature and pressure (depth) detector, inertial sensors, acoustic modem and video. The Marine Systems Engineering Laboratory (MSEL) was initially located at University of New Hampshire and moved to Northeastern University in 1994. Notable contributions of *EAVE* research include implementing multiple level software architectures, multiple vehicle interaction protocols, low-bandwidth acoustic communication languages, and

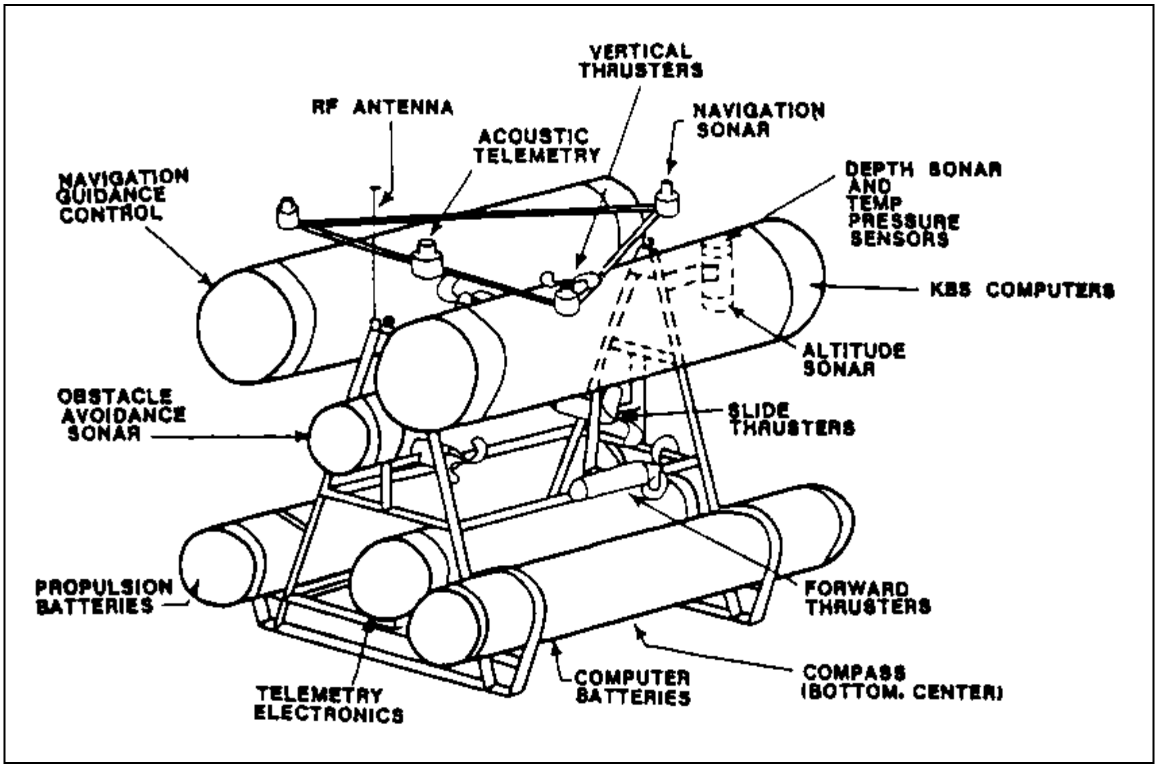

Figure 2.4. Marine Systems Engineering Laboratory (MSEL) Experimental Autonomous Vehicle *EAVE II* equipment layout (Blidberg 90).

unique missions such as rapid-response oil spill underwater survey (Brutzman 93a).

4. Florida Atlantic University (FAU) *Ocean Voyager II*

The ***Ocean Voyager II*** is a long-range AUV designed for coastal oceanography, classifying bottom types by flying at a constant altitude above the sea floor while measuring bottom albedo, light absorption and other parameters (Smith 94) (Brutzman 94a). Results of large-area surveys will be used to calibrate satellite measurements which currently have few correlation checks available with ground truth. The possibility of rapid response means that this AUV mission is also suitable for tactical oceanography. Vehicle hull form is similar in size and shape to the MIT *Odyssey* vehicles, as are most components. Navigation is by ultra-short and long baseline acoustic networks, doppler water velocity log and differential global

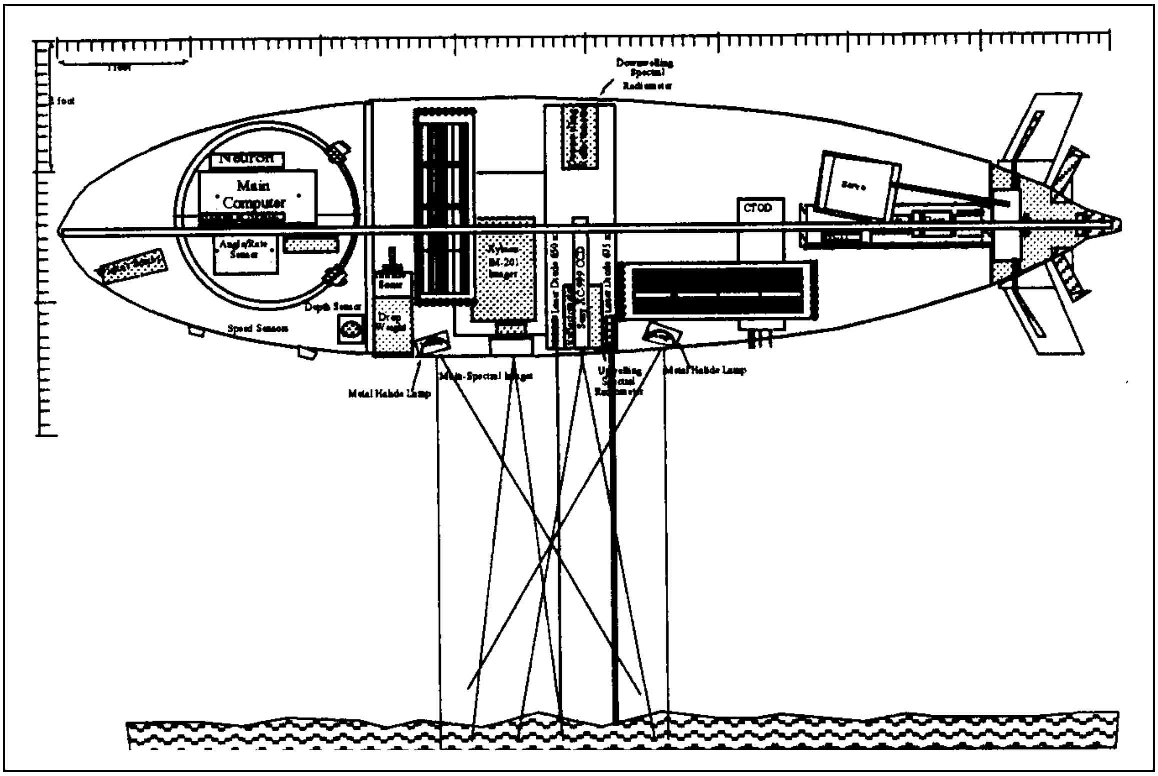

Figure 2.5. Florida Atlantic University Ocean Voyager II (Smith 94).

positioning system (DGPS). Communications are by 2400 Kbps acoustic modems or a towed float radio frequency (RF) antenna when near the surface. Vehicle endurance is 50-100 km, depth rating 600 m, and speed is 3-5 knots. Vehicle design required 1 year and $100,000 initial expense, with sensor payloads comprising over half of the total cost. Collection, correlation and evaluation of large oceanographic datasets are good candidate applications for an underwater virtual world.

5. Monterey Bay Aquarium Research Institute (MBARI) Ocean Technology Testbed for Engineering Research (*OTTER*)

In 1994, Monterey Bay Aquarium Research Institute (MBARI) and the Stanford Aerospace Robotics Laboratory built the Ocean Technology Testbed for Engineering Research (*OTTER*) AUV as a testbed for vision-based servoing for vehicle control while constructing video mosaics of the ocean floor (Marks 92, 94a, 94b) (Brutzman 93a, 94a). Stereo video cameras provide high-bandwidth streams which are

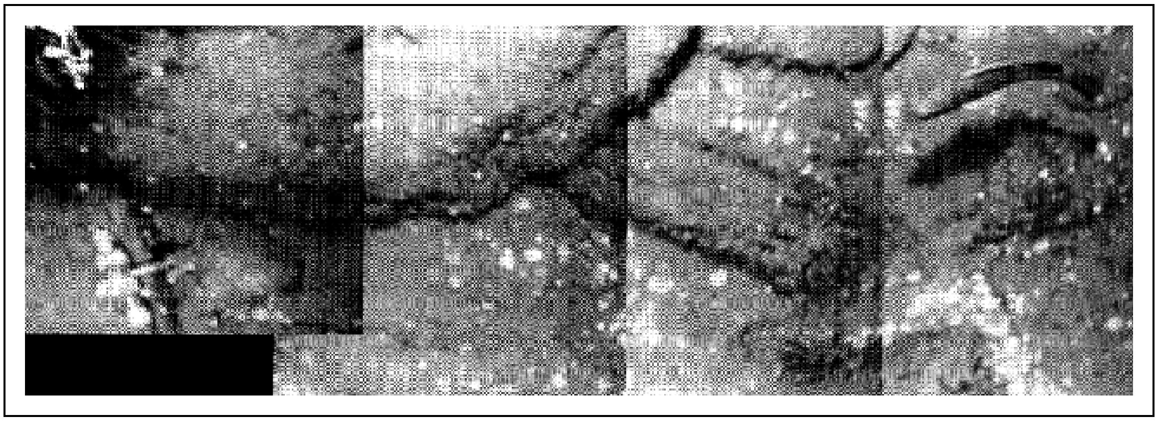

Figure 2.6. Video mosaic from Monterey Bay Aquarium Research Institute Ocean Technology Testbed for Engineering Research (*OTTER*) (Marks 94a, 94b) (Brutzman 94a). Note fish, upper right corner.

subsampled and filtered using vision-processing hardware for real-time response. As demonstrated by (Marks 94a), sequentially applying a signum function, a Laplacian function and a Gaussian correlation function produces images which can be adjusted for stereo disparity and correlated between subsequent frames. This result produces an optic flow output which can then be used for feature tracking. Once a feature has been identified, dynamic feedback to thrusters/planes/propellers controllers permits the *OTTER* vehicle to follow that object or navigate relative to the bottom. The same correlation algorithm can be used to match physically adjacent images into a large-scale video mosaic in real time, often providing a better match than is possible using manual methods. Acoustic transmittal of video mosaics is possible in real time, while transmittal of unculled video is infeasible due to excessive bandwidth requirements. Both object (e.g. sea creature) tracking and bottom mapping are extremely valuable oceanographic capabilities, and are also essential if AUVs are to be practical tools for ocean exploration. Video mosaic mapping and observation of undersea creatures *in situ* are fundamental behaviors for automatic data collection and underwater virtual world database construction.

6. **Woods Hole Oceanographic Institution (WHOI) Autonomous Benthic Explorer (*ABE*)**

The Woods Hole Oceanographic Institution (WHOI) has designed and constructed a special purpose AUV for long-term surveys of the deep ocean floor

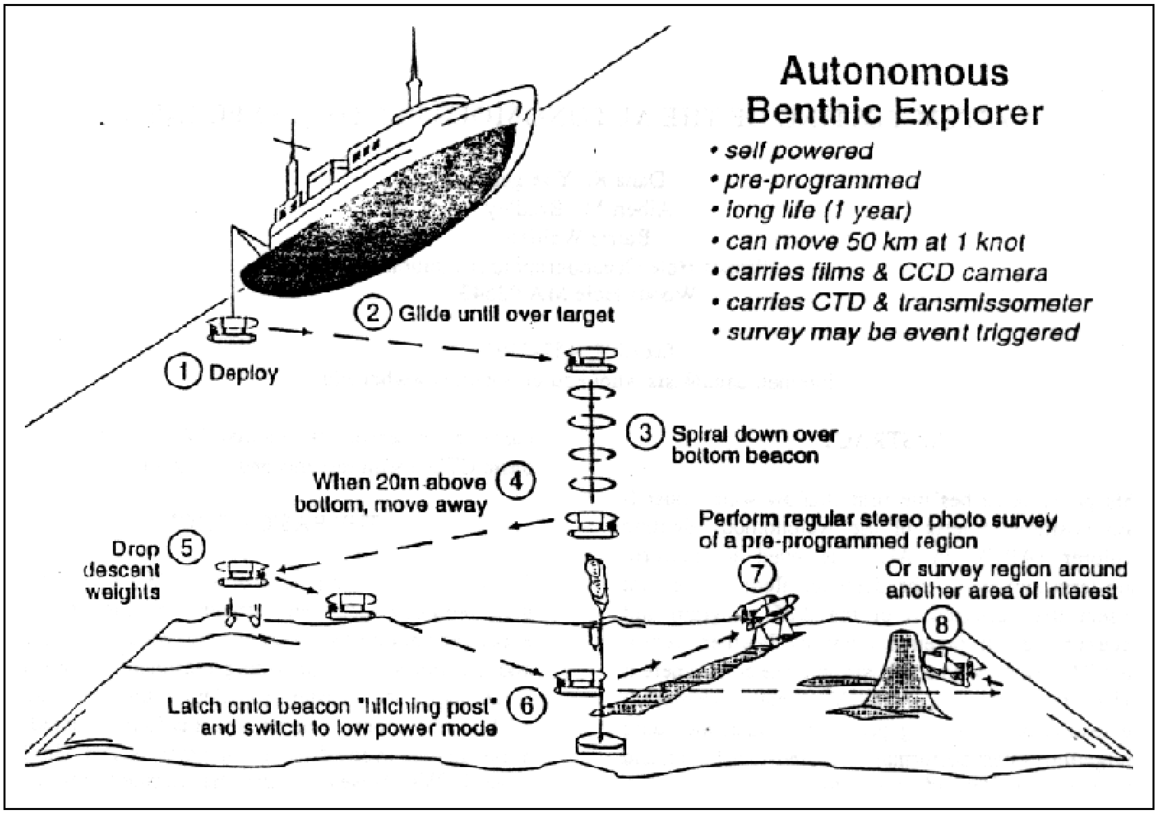

Figure 2.7. Woods Hole Oceanographic Institution (WHOI) Autonomous Benthic Explorer (*ABE*) mission profile (Yoerger 94).

(Yoerger 91, 94). The Autonomous Benthic Explorer (*ABE*) can moor at a fixed location for long periods of time in a "sleep" mode and periodically awake, perform a local survey by navigating within a short baseline acoustic transponder field while measuring water parameters and taking low light charge-coupled diode (CCD) camera photographs, then reattach to the mooring. Power consumption is extremely low in order to support 16 hours of maneuvering endurance spread over missions lasting up to a year. Science missions include observation of deep ocean hydrothermal vents and benthic biologic communities. The vehicle is retrieved following an acoustic command to drop ballast and return to the surface. *ABE* operational ranges and endurance can be significantly increased by attaching the mooring to a magnetic induction power transfer device and acoustic communications relay. Potentially high data rates and the possibility of making geologic measurements with real-time importance make *ABE* deployments a natural application to be networked with an underwater virtual world.

7. Explosive Ordnance Disposal Robotics Work Package (EODRWP)

The Lockheed Explosive Ordnance Disposal Robotics Work Package (EODRWP) is a UUV designed to assist divers in locating, classifying and neutralizing underwater mines (Trimble 94a, 94b) (Brutzman 94a). Although tethered in order to provide power and controller communications, the EODRWP has a sophisticated suite of rule-based behaviors to intelligently perform signal processing, classification, dynamics control, mission planning and mission execution with minimal human supervision. Shore-based graphical simulation connected to vehicle hardware in the laboratory is considered an essential capability and is used to visualize and test the EODRWP prior to at-sea testing. Particular contributions of this project include guidance, navigation, control and mission task integration of human and robot. Use of an underwater virtual world combined with EODRWP and externally-controlled synthetic humans has the potential to improve mine neutralization tactics while reducing risks to navy divers and ships.

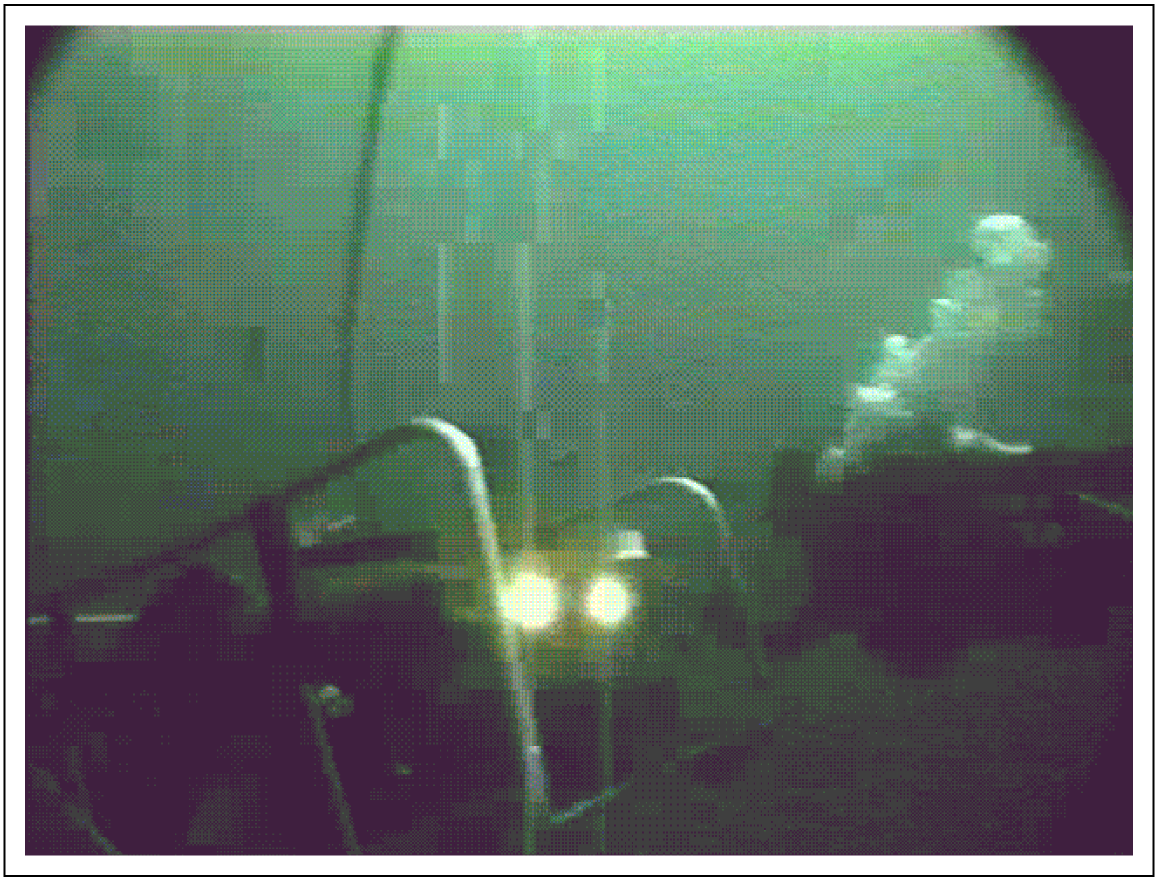

Figure 2.8. Lockheed Explosive Ordnance Disposal Robotics Work Package (EODRWP) and diver (Trimble 94a, 94b) (Brutzman 94a).

8. Miniature AUVs

With exponentially improving price/performance ratios in computer microprocessors, it is natural to expect that miniature AUVs might provide capabilities that avoid the power and propulsion handicaps of larger vehicles. The Smart Communications System (SMARTCOMMs) (Frank 94) is representative of such efforts. As fundamental AUV problems of low-power sensing, low-level dynamics control and high-level mission control are resolved, miniaturization and optimization of vehicles becomes cost effective. It is likely that large numbers of inexpensive and moderately capable AUVs will become available in the near future. Communicating with and coordinating these vehicles in the context of massive environmental datasets, numerous data streams and large ocean areas will be a significant challenge. Networking large numbers of these vehicles within an underwater virtual world can be a practical solution.

C. ROBOTICS AND SIMULATION

A very great number of robotics-related simulations have been produced, but few involve mobile robotics. Those simulations which are available are typically restricted by common limitations of simulation: problems and solutions are approached in a piecemeal and fragmented fashion. Thus simulation results remain susceptible to failure when deployed in the real world due to the untested complexity of multiple interacting processes operating within the hard real-time constraints of unforgiving environments. There is no safe and complete "practice" environment for AUVs, since test tanks cannot reproduce the variability of critical parameters found in the ocean, and since any in-water failure may lead to vehicle damage or loss due to flooding. Known simulation efforts pertaining either to AUVs or construction of robot-centered virtual worlds follow.

1. NPS AUV Integrated Simulator

Research preliminary to this dissertation established "integrated simulation" as a necessary tool for AUV development (Brutzman 92a, 92c, 92e) (Compton 92). Integrated simulation was identified as a suite of simulation tools to assist in the design and testing of all vehicle hardware and software components. An integrated simulator was built that provided real world functionality and visualization for a variety of AI-related tactical software programs. Integration of simulation throughout the software design process was shown to have tangible benefits in producing results that might otherwise have been impossible. Pertinent work preceding that thesis includes (Jurewicz 90) (Zyda 90) (Healey 92a). Confirmation of integrated simulation conclusions were subsequently reported following the successful development of the Multi-Vehicle Simulator (MVS) with the Twin-Burger AUV (Kuroda 94) (Brutzman 94a).

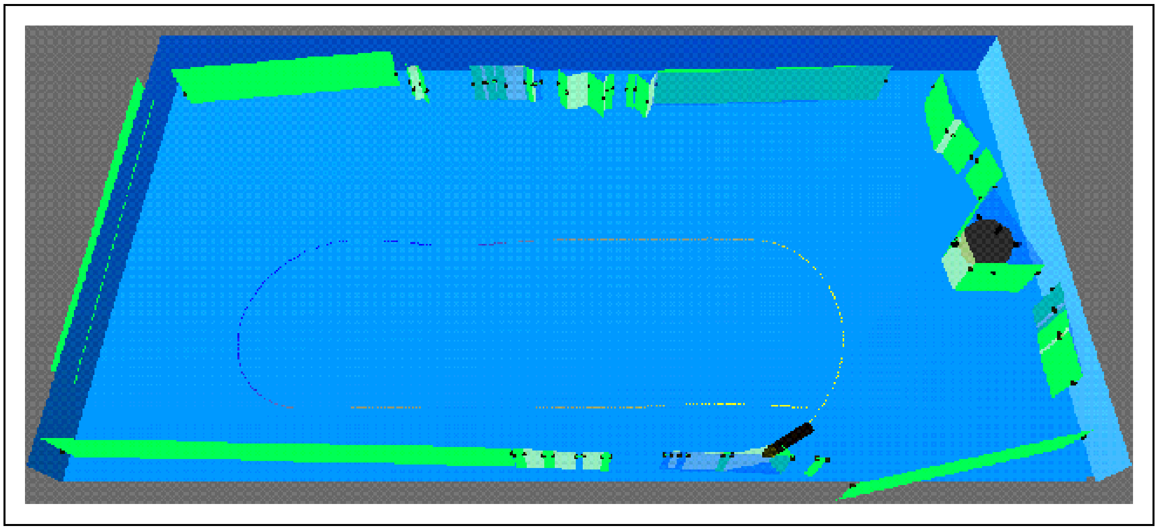

Figure 2.9. NPS AUV Integrated Simulator showing playback of pool mission with autonomous sonar classification expert system results (Brutzman 92a, 92c, 92e) (Compton 92).

Integrated simulation differs significantly from the virtual world produced in this dissertation in that robot-specific hardware and software were completely off-line, real-time response was not required, simulation models were not connected or networked, simulations were single user programs and vehicle hydrodynamics response was only available by playing back in-water test results. Developing and implementing the concepts involved in integrated simulation were important prerequisites to conceiving the notion and defining requirements to build an underwater virtual world for an AUV (Brutzman 92d).

2. ARPA/Navy UUV Hybrid Simulator

The ARPA/Navy UUV development lab at Charles Stark Draper Laboratories includes a simulator which consists of a mainframe computer, models of hydrodynamics and sensor response, and highly detailed component-level models of individual UUV internal equipment (such as motor electrodynamics models) (Pappas 91) (Brancart 94) (Brutzman 93a, 94a). A Simulation Interface Unit (SIU) provides a custom hardware interface between mainframe computer and vehicle.

Mechanisms are also provided to test individual vehicle components. At-sea test dive profiles are first undertaken in the laboratory prior to operational testing. Wireframe graphics provide a simple rendering of vehicle posture during hardware-in-the-loop testing.

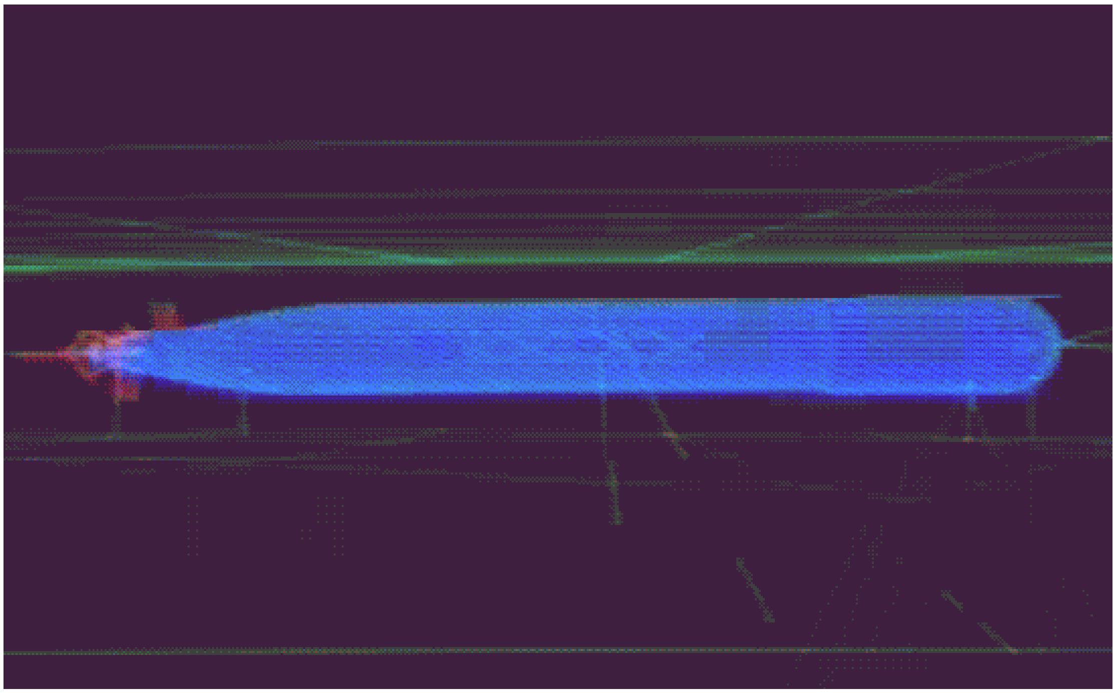

Figure 2.10. ARPA/Navy Unmanned Underwater Vehicle (UUV) Hybrid Simulator wireframe graphics rendering of hardware-in-the-loop laboratory vehicle tests (Pappas 91) (Brancart 94) (Brutzman 93a, 94a).

The ARPA/Navy UUV Hybrid Simulator has much of the functionality needed for a robot-based underwater virtual world, but several important capabilities are missing. The algorithms and source code for the hybrid simulator are not publicly available and many equipment components are proprietary. Since all software components (including computer graphics) are in a single loop on a large mainframe computer, the software architecture cannot scale up indefinitely with the addition of new world models. Graphics are particularly bound since the frame rate of screen updates are tied to the timing of the robot/simulator loop. No mechanisms are provided for remote networked collaboration. Overall, the ARPA/Navy UUV hybrid simulator is one of the best of all "hardware in the loop" simulations, where computer simulation and target system are closely coupled in isolation from any other interaction methods. The ARPA UUV Hybrid Simulator constitutes a tremendous accomplishment which provided inspiration and points of comparison for several parts of this work.

3. NASA Ames Intelligent Machines Group (IMG): Telepresence Remotely Operated Vehicle (*TROV*)

The NASA Ames Intelligent Machines Group (IMG) has worked on a variety of robots and human-computer interface devices with the long-term objective

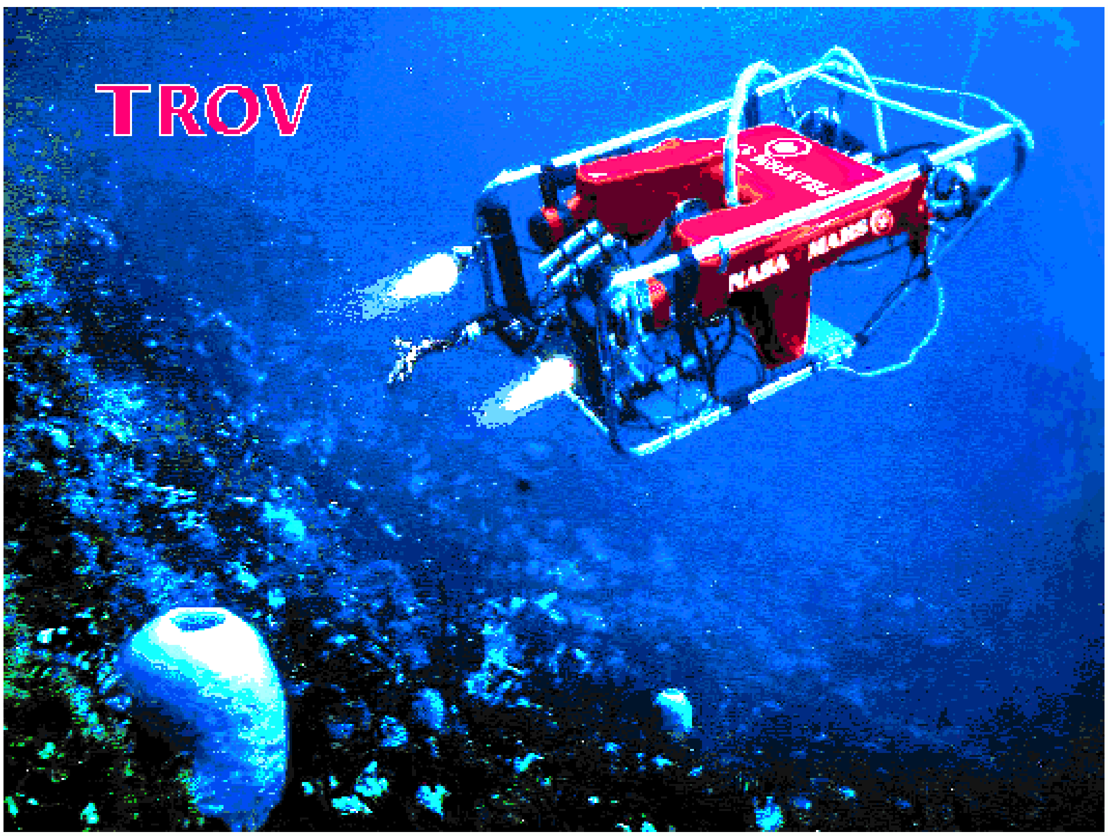

Figure 2.11. NASA Ames Intelligent Machines Group (IMG) Telepresence Remote Operated Vehicle (*TROV*) (Hine 94).

of providing effective telepresence for scientific exploration of other planetary surfaces, such as on Mars (Hine 94). Telepresence is defined as the projection of human senses into remote locations, and its effectiveness is measured by the usefulness of telepresence robotics in conducting actual scientific investigations. Human sense of presence can be enhanced by virtual reality input/output devices (such as headset and data glove) together with virtual world representations combining interactive 3D graphics with low-bandwidth high-latency network links to remote robots. In 1993 NASA Ames deployed the Telepresence Remotely Operated Vehicle (*TROV*) under Ross Sea ice near McMurdo Science Station, Antarctica. The underwater vehicle was an open-frame Phantom S2 ROV with four thrusters, stereo video cameras, a gripper manipulator, oceanographic sensors, acoustic transponder navigation, four commandable degrees of freedom and 1000 ft depth capability. Communication with the *TROV* was via a twisted-pair umbilical tether to the *TROV* controller topside and then using Internet Protocol (IP) packets over infrared (IR) laser, microwave and intercontinental satellite links. This varied communications path induced significant latencies, albeit still less than those experienced at interplanetary distances. The Virtual Environment Vehicle Interface (VEVI) modular operator interface for direct teleoperation and supervisory (task-level) control integrated all inputs and outputs, including a head device for steering the viewing cameras and incrementally updated graphics models for terrain and other pertinent physical objects. Science teams running the two-month mission focused on marine biology, chemical oceanography and benthic ecology. Science objectives were met and teleoperation was proven feasible from a variety of locations around the globe. Stereo displays provided excellent depth perception, and controller time-delay modifications for task-level control and predictive teleoperation response proved successful. Related work includes the DANTE robot exploration of the Alaskan volcano Mt. Spurr and possible regional collaboration in deep AUV exploration of Monterey Bay. *TROV* is representative of the most sophisticated teleoperated robots.

A survey and analysis of telerobotics capabilities and trends appears in (Durlach 94). Principal reference in the telerobotics field remains (Sheridan 92).

4. University of Hawaii: Omni-Directional Intelligent Navigator (*ODIN*)

The University of Hawaii Omni-Directional Intelligent Navigator (*ODIN*) project combines an AUV with an integrated graphics simulation for development of adaptive dynamics control algorithms (Choi 94). *ODIN* is a small spherical AUV with

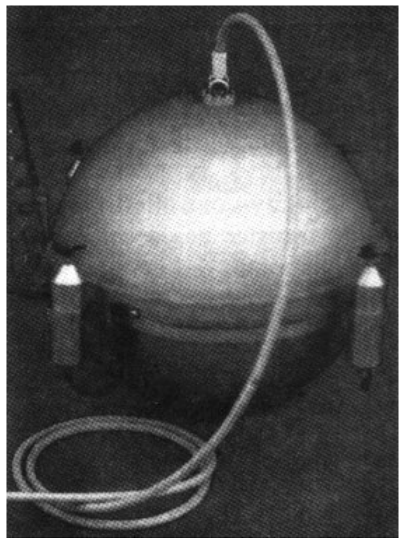

Figure 2.12. University of Hawaii Omni-Directional Intelligent Navigator (*ODIN*) (Choi 94).

a single manipulator and four steerable vertical thrusters, capable of posture control in six degrees of freedom. Primary research conducted using *ODIN* concerns determination of hydrodynamics coefficients, linear controllers, nonlinear controllers, and adaptive controllers utilizing fault detection and automatic reconfiguration using neural networks. Integration of a single graphics workstation with *ODIN* demonstrates the functionality independently described in the NPS AUV Integrated Simulator (Brutzman 92a, 92c).

5. Tuohy: "Simulation Model for AUV Navigation"

(Tuohy 94) developed a simulation model to test AUV navigation applications. An object-oriented approach organized the overall simulation model into environmental models (consisting of terrain and water column maps) and physical object models (consisting of sensor, command/program and dynamics models). Contributions of this work include a proposed general model partitioning suitable for vessels and static structures, emphasis on map decomposition using spatial data structures, and model integration with 3D graphics.

6. Chen: "Simulation and Animation of Sensor-Driven Robots"

(Chen 94) describe how most robotics simulations include robot and environment while excluding sensors, and identify the creation of realistic simulation and animation software as an important robotics research issue. They present a system for simulation and animation of sensor-driven robot manipulators and indoor mobile robots. The system hierarchy includes models for robot, tool in work cell, sensors and physical objects. Physically-based models for proximity, point laser range, laser range depth imagery and vision intensity sensors are included, with research continuing on force/torque and tactile sensors. Three-dimensional interactive graphics are used for robot and sensor visualization, although real-time performance is not guaranteed. Robots can be integrated into the simulation system to permit running in real mode or virtual mode, either interactively or through recording playback. In real mode, robot controller subsystem electronics are physically connected to ports on the simulating workstation for two-way communication of command and sensor information. In virtual mode, robot software is run on the same workstation as the computer graphics, independently of robot hardware. Primary conclusion of this work is that a simulator for an integrated sensor-driven robotic system must incorporate simulation of sensory information feedback. Planned future work includes incorporation of a voice-recognition module in the robot and adding dynamic models to other objects in the simulation environment.

7. Yale University: Ars Magna Abstract Robot Simulator

The **Ars Magna** mobile robot simulator provides an abstract planar world in which a AI planner is able to control the movement of a mobile robot (Engelson 92). The objective of the simulator is to provide a more challenging and realistic environment for developing and evaluating planning systems than was previously available. Vehicle motion is purely kinematic and is based on a single point. Simulated manipulators are included. Sensor values are provided by geometric range models with adjustable noise distributions. Robot planning programs are written in a variant of the ***Lisp*** programming language. The useful but limited capabilities of the **Ars Magna** are representative of most other robot simulators currently in use.

D. UNDERWATER VEHICLE DYNAMICS

The study of dynamics and physics-based motion has long been recognized as a necessary prerequisite for realistic computer graphics rendering and valid robotics performance modeling. Although numerous articles pertaining to underwater hydrodynamics exist, almost without exception they focus on some small aspect of hydrodynamics performance. A complete hydrodynamic model suitable for real-time simulation response has not been available prior to this dissertation. An overview comparison of dynamics models in different environments appears in the hydrodynamics chapter. In this section key references preceding the new hydrodynamics model are identified.

1. Healey: Underwater Vehicle Dynamics Model

An earlier underwater vehicle hydrodynamics model presented in (Healey 92c, 93) provided the fundamental basis for the general hydrodynamics model. Strengths of the model included theoretical rigor, completeness for cruise operations using propellers/rudders/plane surfaces, and several years of empirical testing which produced an initial working set of hydrodynamics coefficients. Limitations include missing terms for thruster forces and moments, missing terms for low-speed hovering drags, extraneous terms corresponding to an unusual vehicle configuration, and an arrangement of multiple differential equations not easily adapted to real-time temporal integration. Further details are provided in Chapter VI. Of all dynamics models examined, this was by far the best. Work presented in this dissertation extends and generalizes that fundamental contribution.

2. Fossen: *Guidance and Control of Ocean Vehicles*

Numerous texts exist on marine vehicle dynamics, most notably (Lewis 88), but their focus is almost exclusively on surface ships. (Fossen 94) provides a thorough treatise on both surfaced and submerged vehicle dynamics and control. He also examines stability, ocean modeling of wind and waves, and advanced control techniques. Theoretic derivations and explanations are provided throughout. Of relevance is that (Fossen 94) includes a total of three example underwater vehicle hydrodynamic models: two simplified linearized models (each by Healey) and the verbatim original six-degree-of-freedom model of (Healey 93).

3. ARPA/Navy UUV Hydrodynamics Simulation

The Navy/ARPA UUV design and development team has reported using a full six-degree-of-freedom hydrodynamics model for development and testing of sophisticated vehicle controllers (Pappas 91) (Brancart 94). Further details have not been published publicly.

4. Yuh: "Modeling and Control of Underwater Robotic Vehicles"

(Yuh 90) provided an important contribution to the underwater vehicle hydrodynamics literature. Although presented as an remotely-operated vehicle (ROV) model, it is pertinent to any type of underwater vehicle. He describes "added mass" and most other relevant terms. Nomenclature and algebraic differences make this model different but still close to the (Healey 93) model described earlier.

5. U.S. Navy Submarine Hydrodynamics

The subject of U.S. naval submarine dynamics is classified and was not considered during this work. Some open literature exists. (Jackson 92) provides an overview of the basic submarine design process, examining general requirements and how design tradeoffs must be weighed. (Gertler 67) and (Feldman 79) present the general form of dynamics equations and coefficient nomenclature, closely conforming to the standard mechanical engineering reference (Lewis 88). No claims or suppositions regarding any classified work are made, implied or conjectured in this dissertation.

E. NETWORKED COMMUNICATIONS FOR VIRTUAL WORLDS

Networking considerations in the construction of virtual worlds have gained increasing importance in recent years. As virtual worlds grow in complexity and quantity of information represented, the ability to scale up and accommodate arbitrarily large numbers of information sources and interacting entities becomes a crucial requirement. Currently there are many bottlenecks preventing unlimited and seamless virtual world communications. Research in this area is very active (Zyda 95). Multicast network protocols are a fundamental development in this regard and are examined further in Chapter VII. This section examines recent work in networking virtual worlds with an emphasis on scalability considerations.

1. SIMulation NETworking (SIMNET) Architecture

SIMNET was the first architecture that permitted large numbers of simulated entities to interact together in real time, using heterogenous hosts and distributed communications over a network (Calvin 93). With over ten years of development and operation, SIMNET is a proven system. Key design principles are that objects interact in the virtual world by communicating events, all objects must relay valid data, network bandwidth is reduced by only transmitting state changes, and dead reckoning algorithms are used to predict intermediate postures. Enabling technologies for SIMNET were real-time computer graphics (image generators), distributed dynamic and static virtual world databases, semi-automated forces (SAF) which provide realistic entity or aggregate force behaviors, high speed local area networks (LANs) coupled with an interaction protocol, and free choice of human-computer interfaces. SIMNET effectiveness in Army tactical team training for combat has been documented on many occasions, such as the Battle of 73 Easting during the Iraq war (Calvin 93). The biggest theoretical success of SIMNET has been implementation of the interaction protocols, which became the foundation for the DIS protocol (IEEE 94a, 94b). As might be expected with any first-generation system there are some problems with the SIMNET architecture concerning scalability, many of which are addressed by ongoing DIS protocol development efforts. SIMNET protocols do not use Internet Protocol services, but instead require root superuser permissions for execution since they access hardware interfaces at the data link layer directly. In practice SIMNET capacity is limited to 300 simultaneous players (Durlach 94).

2. Distributed Interactive Simulation (DIS) Protocol

The DIS protocol is an approved IEEE standard for communications between entities in small or large scale virtual environments (IEEE 93). From the recent proposed DIS standard revision:

> "Distributed Interactive Simulation (DIS) is a government/industry initiative to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual ’worlds’ for the simulation of highly interactive activities. This infrastructure brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services and permits them to interoperate. DIS exercises are intended to support a mixture of virtual entities (human-in-the-loop simulators), live entities (operational platforms and test and evaluation systems), and constructive entities (wargames and other automated simulations)." (IEEE 94a, 94b)

The principal type of interaction in DIS is transmission of entity state information via Protocol Data Units (PDUs) which include position, orientation, time and (optional) velocity and acceleration values. A variety of standardized dead reckoning algorithms are available to maximize positional information transfer while minimizing bandwidth consumed. Numerous other PDU types are included which relate to exercise management, collisions, sensor emissions, and entity interactions such as weapons fire and logistic support. Free and commercial DIS software libraries are available. The DIS protocol development community is very active and DIS continues to evolve. Current efforts are focused primarily on supporting larger numbers of simultaneous entities, and also on extending DIS functionality to support additional world information such as environmental effects and distributed terrain databases (IEEE 94a, 94b).

3. NPSNET

NPSNET is a networked virtual environment for battlefield simulation. Key strengths are high performance, a distributed software architecture, ability to

Figure 2.13. NPSNET-IV virtual battlefield showing multiple active DIS-based entities, textured terrain and atmospheric effects running at high frame rates in real time (Pratt 93, 94b) (Zyda 93b).

handle large numbers (hundreds) of interacting human and autonomous entities in real time, initial implementation of multicast DIS libraries, public distribution, and insertion of remotely-controlled synthetic human models in virtual environments. NPSNET has over one hundred institutional users and has been a key component in numerous large-scale Army simulation exercises. NPSNET is likely the broadest and highest-performance virtual environment software that currently exists. Software distributions are free to registering users. Ongoing research efforts include object-oriented techniques for virtual environment construction, application level and network level communication protocols, hardware and operating system optimization, real-time physically-based modeling (e.g. smoke, dynamic terrain and weather), integration of multimedia, AI for autonomous agents, integration of analytic models such as JANUS, and human interface design (e.g. stereo vision and system controls) (Pratt 93, 94a, 94b) (Macedonia 95b) (Zyda 93a, 93b).

4. Macedonia: "Exploiting Reality with Multicast Groups"

Although DIS can scale to permit simulation exercises with several hundred interacting entities, several bottlenecks constrain current DIS network implementations from going much higher. This is a problem since distributed simulations accommodating tens of thousands of active entities are needed. One key difficulty is that participating hosts must listen to every DIS report, a requirement that eventually consumes all host processing cycles. (Macedonia 95a, 95b, 95c) proposes partitioning the communications space into more manageable streams through the considered use of multicast channels. Since multicast packets can be collected or discarded using network interface hardware at the data link layer, hosts need only process DIS traffic corresponding to subscribed multicast channels. Large-scale virtual worlds can thus be partitioned according to geographic space subdivisions, functional classes (such as radio frequencies), and temporal classes (such as normally static buildings or highly dynamic jet aircraft). Development of area of interest management protocols thus becomes necessary for retaining complete state corresponding to a given channel, providing a state snapshot to newly joining entities, and handing off control to another manager if necessary. Protocol extension implementation, experimentation and analysis is in progress. The capability to use multicast protocols will be required for future DIS compliance (IEEE 94a), underscoring the importance of these concepts. These ideas are explained within the larger context of state-of-the-art trends in virtual reality networking and communications in (Durlach 94).

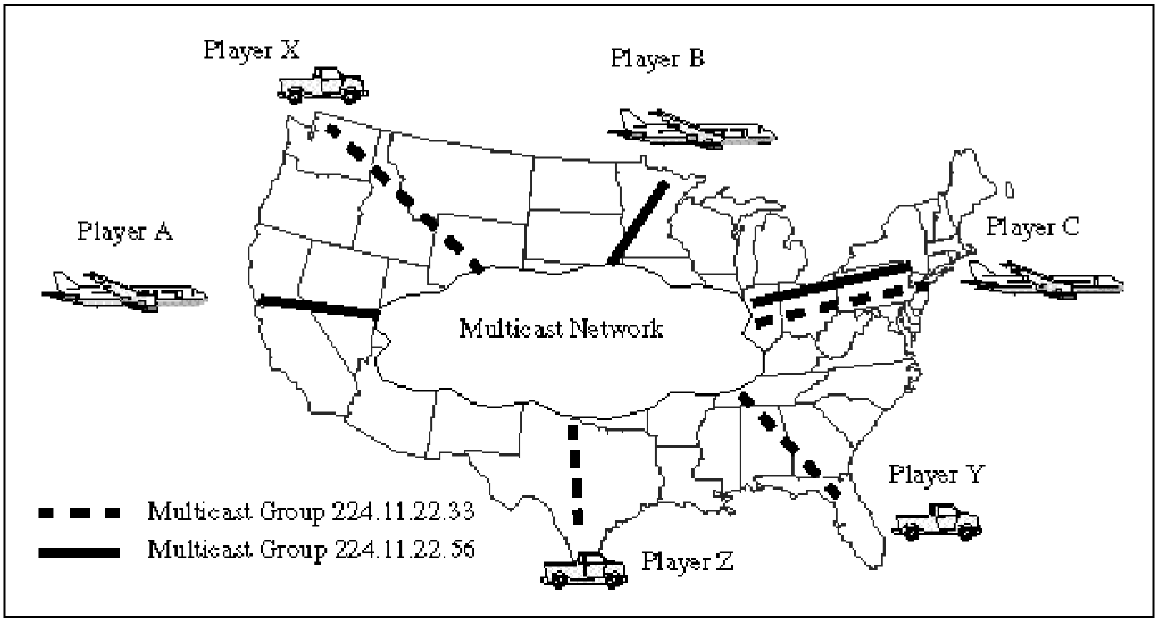

Figure 2.14. "Exploiting Reality with Multicast" - multiple DIS channels for geographic sectors, functional classes (e.g. communications) and temporal classes (e.g. highly dynamic aircraft) (Macedonia 95a).

5. Gelernter: *Mirror Worlds* and Linda

(Gelernter 92a, 92b) describes a powerful set of abstractions for networked virtual world communications. He extends and simplifies the message-passing paradigm used by communicating software objects through creation of a "tuple space." Tuples are persistent messages without a specific addressee. Tuples are ordered lists that begin with some keyword and contain any number of additional elements. Processes have three operations to use with tuples: jettison, grab and read (alternatively publish, consume and nondestructive read). Processes can access tuples by pattern matching against any or all potential tuple elements, thus retrieving individual tuples or groups of tuples. Tuple space consists of these persistent tuples being read and generated by information machines (i.e. processes), somewhat similar to a blackboard architecture. Since tuple elements might be further tuples, and because tuples can themselves be programs, recursive hierarchies and distributed processing are natural possibilities without explicit specification by the original programmer. This communication methodology has also been shown to be identically portable to massively parallel processors, permitting programmers to concentrate on developing parallel algorithms for problem solving rather than tuning the idiosyncracies of the underlying hardware (Gelernter 92b).

These concepts define the characteristics of coordination languages, which extend computational programming languages in a general and orthogonal way (Gelernter 92b). Arguably coordination languages provide the ability to scale up the number of interacting computational processes to a degree that can reflect real world functionality; hence "mirror worlds" (Gelernter 92a). Initial implementation of these ideas is demonstrated by the Linda communication system (Carriero 91) (Gelernter 92a, 92b). As virtual worlds continue to grow and network bottlenecks permit much larger numbers of entities to interact, implementing the functionality of nonhierarchical nonimperative distributed communication schemes as described in *Mirror Worlds* will be essential.

6. Distributed Interactive Virtual Environment (DIVE)

Distributed Interactive Virtual Environment (DIVE) is a heterogeneous distributed world representation that shares copies of a world database to permit multiple users and applications to simultaneously interact in a single virtual 3D space (Carlsson 93). The world database serves as a global memory shared over the network using a reliable ordered multicast scheme. Maintaining global database consistency is an important problem in large-scale virtual worlds. Multicast protocol packet delivery is ordinarily "best effort" and not guaranteed. Including sequential numbers to each message can achieve reliability for multicast through retransmission, but the cost of that error recovery is expensive and such approaches (as exemplified by DIVE) currently do not scale past several dozen peers (Macedonia 95c). Static and dynamic distributed databases are another key bottleneck that must be addressed for arbitrarily scalable virtual worlds.

7. Other Network Communication Systems for Virtual Worlds

Many other active research projects are working on eliminating the barriers which prevent arbitrarily scaling up distributed virtual world communications. Recommended references are (Zyda 95) (Singh 94) (Bricken 94) (Shaw 93) (Morrison 95) (Codella 93) (Kazman 93). Overlapping and interdependent areas of investigation include:

- peer-to-peer versus client-server models
- network bandwidth reduction
- network processing reduction for participating hosts
- reliable versus best-effort delivery
- object-oriented functional partitioning
- parallelization to improve performance
- decoupling user interfaces (input devices and output graphics)
- persistent and coherent distributed global database management
- open toolkit construction
- compatibility over heterogenous platforms, peripheral hardware independence
- operating system modifications for improved performance
- defining temporal relations, establishing synchronization
- application interaction protocols

Aside from the common denominator of Internet Protocol (IP) use and occasional compliance with the Distributed Interactive Simulation (DIS) application protocol, there is little direct compatibility among any of the aforementioned approaches. Even if a "silver bullet" solution were to emerge from these many efforts, current virtual worlds are likely to remain isolated as closed, incommunicado islands of functionality. General requirements for open interoperability between virtual worlds are examined in (Bréant 94). Specification and development of a specific open communications model as an extension of World-Wide Web (WWW) is a goal of the Virtual Reality Modeling Language (VRML) working group (Bell 94) (Pesce 94). A commonly accepted baseline interaction model for virtual world communications is needed.

F. SONAR MODELING AND VISUALIZATION

Sonar modeling attempts quantify and predict the highly variable behavior of sound waves underwater. A large number of sonar models have been in use since sonar was first widely employed in the 1940s, and development of effective sonar models is the subject of ongoing research. Sonar visualization is the application of scientific visualization techniques for rendering sonar information, in an attempt to better understand the temporal, spatial and physical behavior of underwater acoustics. It is a relatively new area of study. This section identifies prominent related work in sonar modeling and sonar visualization.

1. Etter: Acoustic Modeling

(Etter 91) presents a comprehensive treatment of underwater acoustic modeling, defined as "the translation of our physical understanding of sound in the sea into mathematical formulas solvable by computers." He first treats the physics of underwater sound and acoustical oceanography, synopsizing another key reference on sonar behavior (Urick 83). Sound speed in the ocean is identified as the single most important acoustic variable. Etter then identifies three broad classes of sonar models and organizes the wide variety of existing sonar models into a conceptual hierarchy, shown in Figure 2.15. Each model type is examined in depth. There is no "perfect" sonar model suitable to all situations, and users must carefully choose models (or combinations of models) based on problem requirements. Typically models become less general and more specific to individual sonar systems as one proceeds up the hierarchy.

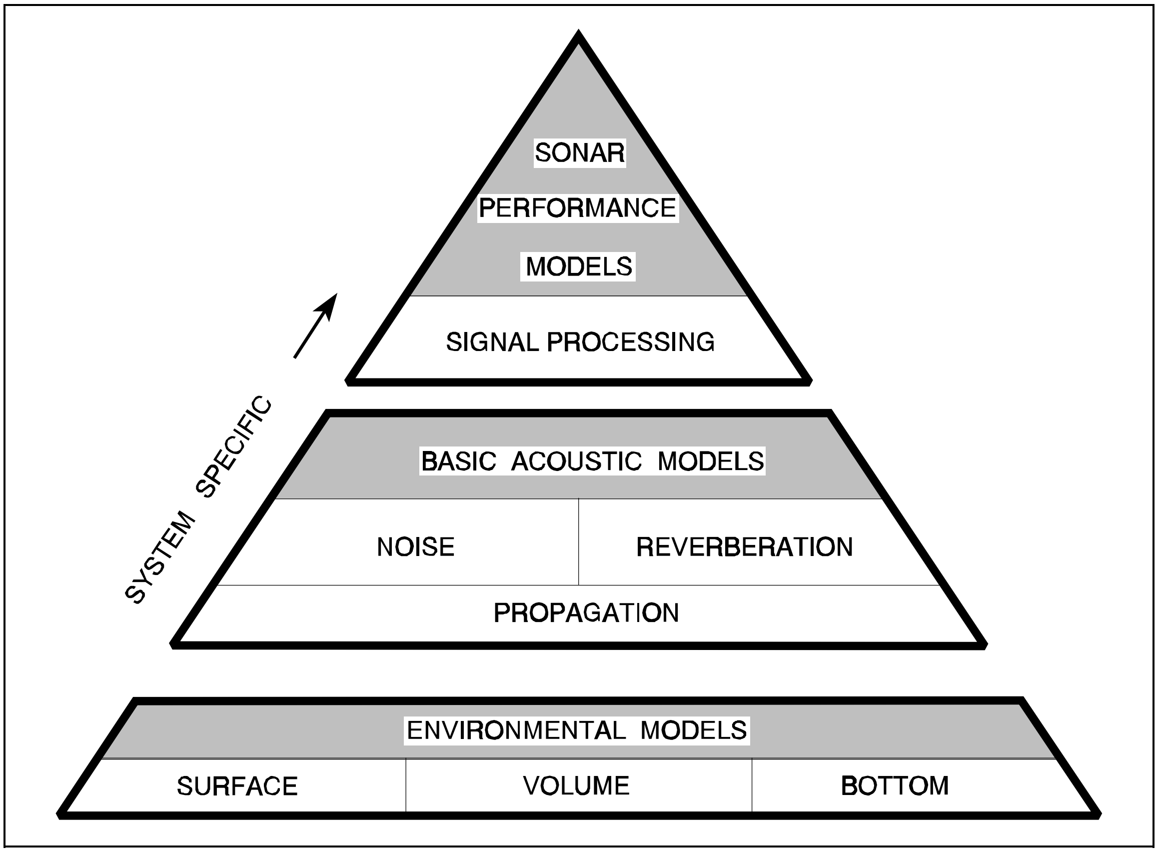

Figure 2.15 Generalized relationships among Environmental Models, Basic Acoustic Models and Sonar Performance Models (Etter 91, p. 3).

The three types of models identified are Environmental Models, Basic Acoustic Models and Sonar Performance Models. Environmental Models examine ocean surface and bottom boundary conditions as well as volumetric effects. Basic Acoustic Models represent the physics or empirical behavior of noise, reverberation and propagation (transmission loss). Sonar Performance Models combine signal processing theory with the preceding Environmental Models and Basic Acoustic Models to enable end-to-end solution of typical sonar detection problems particular to specific types of sonar equipment.

The field of sonar modeling is characterized by tremendous variety. Most models have very narrow domains of applicability and may need to be used in combination with others for the solution of specific problems. Management of this complexity has even led to the development of model operating systems (MOSs) which attempt to assist users by managing the selection of multiple models and appropriately connecting their various input/output requirements. Initial examination of the subject of sonar modeling from the perspective of underwater virtual world construction identified this plethora of models as a key obstacle to scalability and generality. This difficulty is compounded by the fact that many models are reported to be classified (Etter 91) and unavailable for use in an open, arbitrarily scalable virtual world.

2. Stewart: Stochastic Backprojection and Sonar Visualization

(Stewart 88) presents a novel approach to modeling underwater objects. Sonar data are typically high-bandwidth high-noise information streams that include

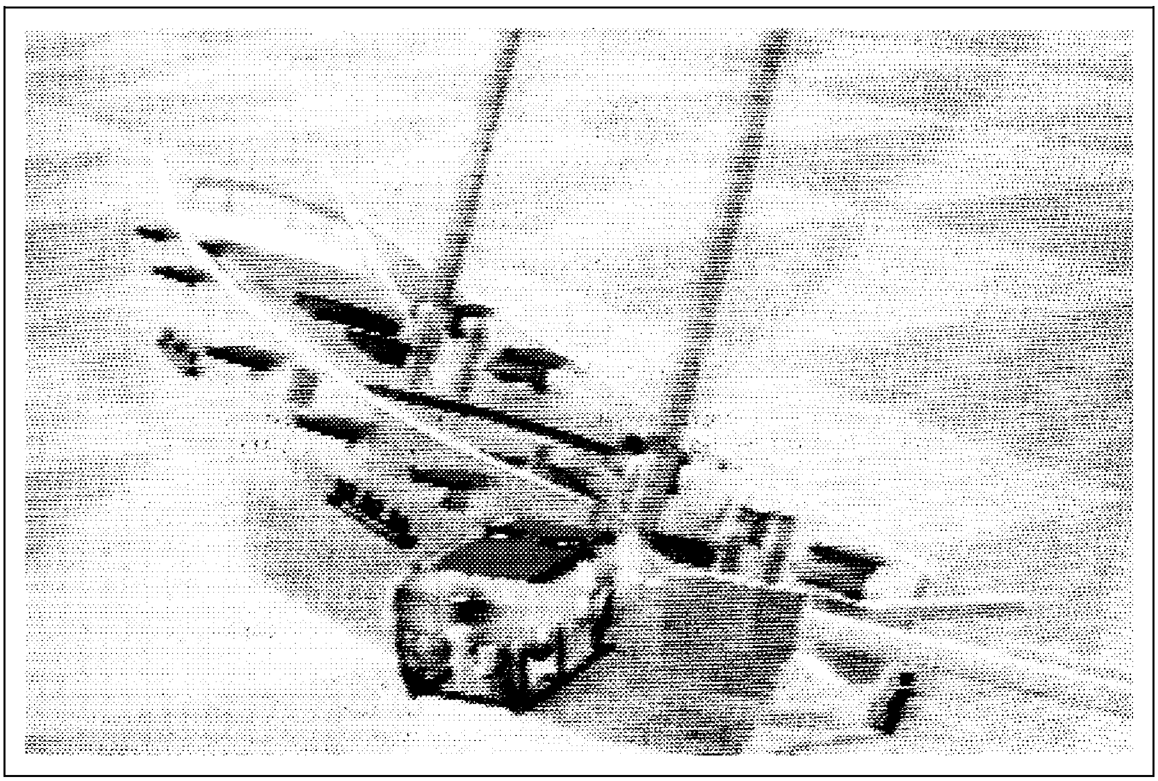

Figure 2.16. Graphics visualization of *JASON* ROV approaching submerged wreck *HMS SCOURGE* (Stewart 92).

redundant returns from target of interest, as well as a large proportion of signal corresponding to false returns or objects of little interest. Key characteristics of underwater sensing applications include "real-time constraints; unstructured, three-dimensional terrain; high-bandwidth sensors providing overlapping, redundant coverage; lack of prior knowledge about the environment; and inherent inaccuracy in sensing and interpretation." Sonar and other sensor returns are treated as probability distributions which are adaptively combined to create 3D maps of terrain and object surfaces using a new statistical technique, *stochastic backprojection*. Model representation accuracy and certainty improve as redundant data accumulates. Intermediate results are available and steadily improve in real time, permitting "anytime" use by operators or robots. Reduction of bandwidth and extraction of useful information are also significant benefits. Stochastic backprojection is appropriate for use in bathymetric mapping, ROV piloting control, and world modeling for AUVs.

Sonar visualization techniques were essential to the successful development of stochastic backprojection methods, since qualitative visual inspection of results were used to evaluate model effectiveness. In addition to the sonar visualization techniques presented in (Stewart 88), an illustrated survey of underwater visualization in (Stewart 92) supplemented by (Stewart 89, 91) and (Rosenblum 93) presents a thorough state-of-the-art summary of sonar visualization and underwater sensor visual representations.

3. Ziomek: Recursive Ray Acoustics (RRA) Algorithm

As previously noted, a key difficulty in sonar modeling as applied to underwater virtual world use is the very large numbers of models that are available for different ocean conditions and different sonars. The Recursive Ray Acoustics (RRA) algorithm (Ziomek 93, 94) provides an approach which appears to be general and well-suited for real-time graphics rendering. A ray tracing algorithm, RRA derives the fundamental wave equations describing sound propagation from a differential equation form to a difference equation form. Three-dimensional models for sound speed profile (SSP) and terrain bathymetry are retained as independent inputs. The algorithm is fast

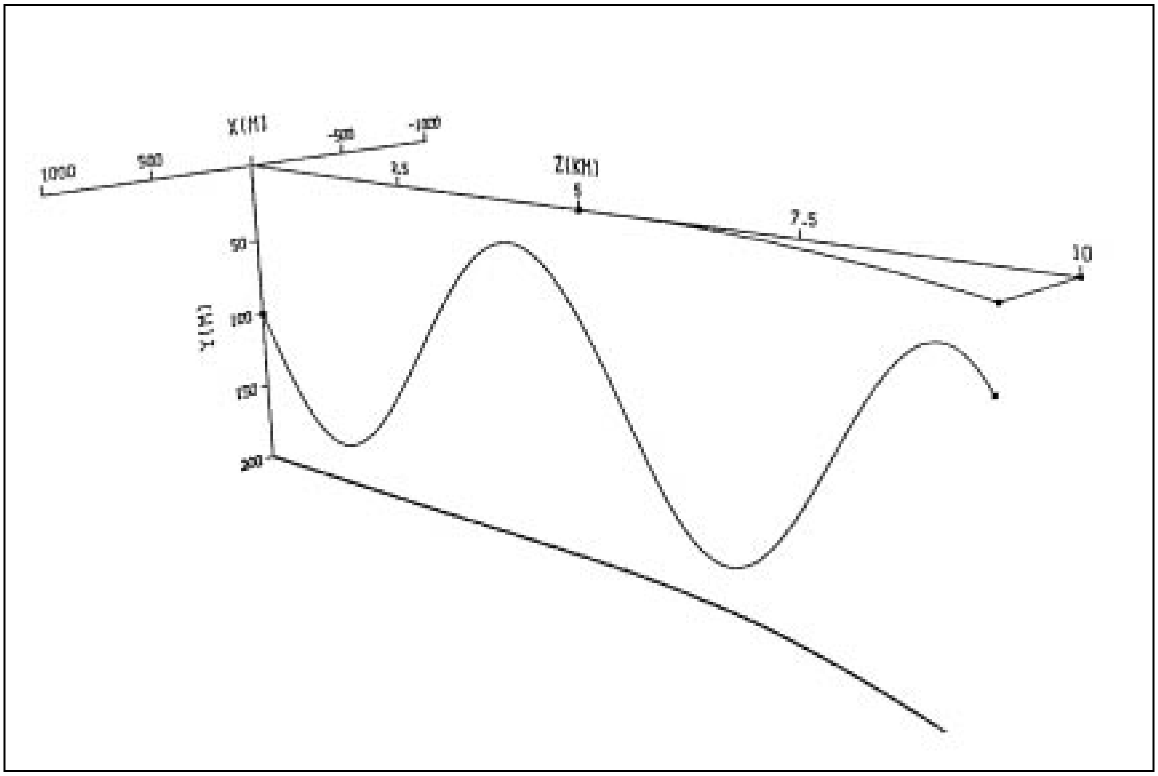

Figure 2.17. Example Recursive Ray Acoustics (RRA) algorithm plot showing sound ray bending due to vertical and down-range sound speed profile (SSP) variations (Ziomek 93).

since each short ray segment in a long ray path is calculated recursively based on the ray segment preceding. RRA can be used to calculate position, propagation angles, sound pressure level (SPL) and travel time along a ray path. Most significantly it appears to be applicable over a wide range of frequencies since approximations and empirical simplifications are avoided in the original RRA derivation. Comparison of RRA results with different models validated in a variety of problem domains has been excellent. RRA appears to be a general, precise and rapid algorithm suitable for real-time sonar modeling and visualization.

4. Additional Work in Sonar Visualization

(Rosenblum 93) presents an overview of current work relating to sonar visualization. Additional images and explanation appear in (Rosenblum 92)

(Kamgar-Parsi 92). (Karahalios 91) examines volumetric sonar visualization concepts and presents example visualizations using near-field sonar processing data. Additional images from her work appear in (Keller 93, p. 122). A summary of underwater acoustic models which includes example sonar visualizations is (Porter 93). Wireframe sonar visualization is included in simulated AUV use of mine avoidance tactics in (Hyland 93). Occupancy grid methods presented in (Elfes 86) are further considered in (Auran 95). A variety of 2D line drawings which incorporate uncertainty information appears in (Leonard 92). Scientific visualization techniques applied to the display and interpretation of very large environmental datasets appear in (Rhyne 93a, 93b).

G. ONGOING AND FUTURE PROJECTS

Directions taken in this work have also considered current and future efforts which might benefit from an underwater virtual world approach. The following projects represent many diverse and fascinating research areas which might benefit from connection to a distributed underwater virtual world architecture.

1. *JASON* ROV and the Jason Project

The *JASON* remotely operated vehicle (ROV) has been used to conduct scientific exploration on a wide range of oceanographic and historic sites of interest (Ballard 93), including investigation of benthic chemosynthetic tubeworm communities and discovery of *HMS TITANIC*. Deep ocean investigations using *JASON* are supported by a surface ship with a control van, as well as the intermediate tow sled *MEDEA* which provides lights and local decoupling from long trailing tethers. In addition to power and control signals, the use of fiber optics permits transmission of high-bandwidth sensor and video data from vehicle to support ship.

In 1989 the JASON Foundation for Education was formed to utilize scientific exploration missions best exemplified by the *JASON* ROV as a catalyst and central focus for widely distributed distance learning (Brown 93). JASON Project missions are held annually. Students first learn about science objectives in detail

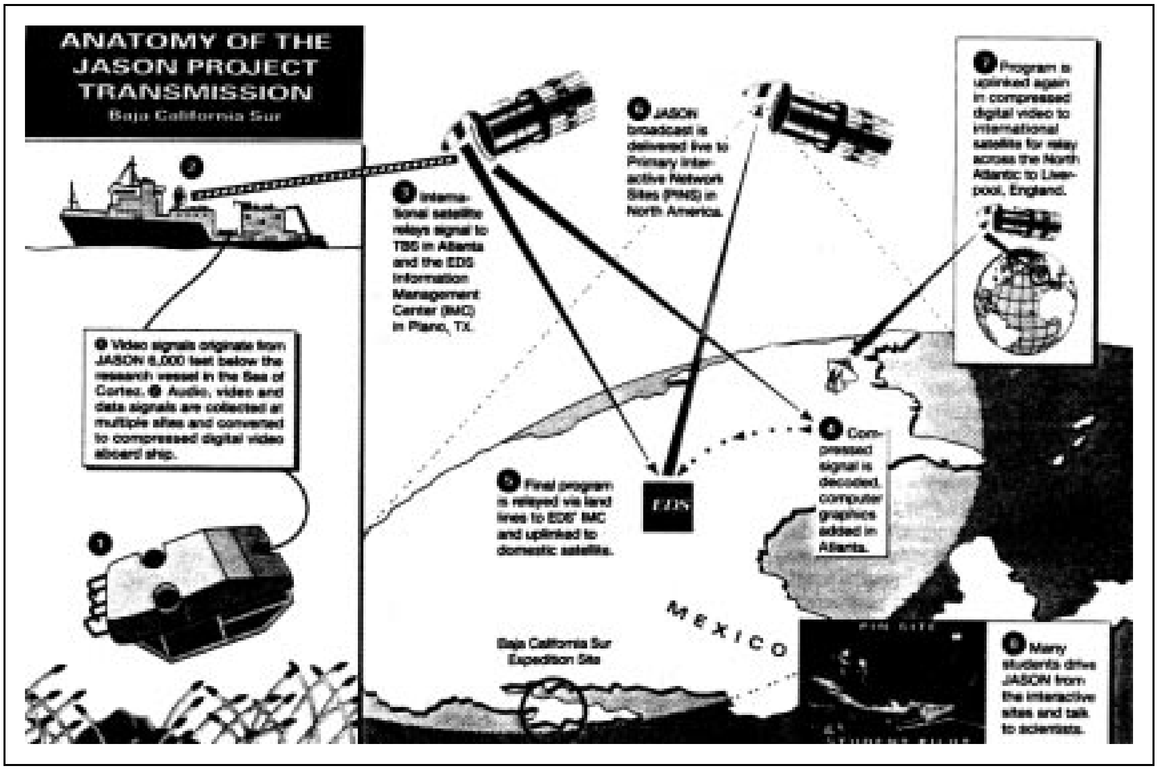

Figure 2.18. *Jason* ROV mission profile and JASON Project communications links (Brown 93).

during regular classes, and then observe and participate in the expedition as it occurs. A team of about a dozen students assists researchers on site while tens of thousands of remote students watch live video streams via satellite downlink. A small number of these remote students are also able to teleoperate the ROV via the satellite link. Months prior to each annual expedition, teachers are given a comprehensive multidisciplinary instructional guide which helps integrate subjects such as oceanography, physics, archaeology, history, biology etc. into the regular school curricula. Students are thus provided live real world examples to motivate and invigorate their studies.

Scientists also remotely participate in these missions. Scientific objectives are not diluted but rather extended to include students in the conduct of significant actual research. Live real-time multicast dissemination of *JASON* vehicle telemetry

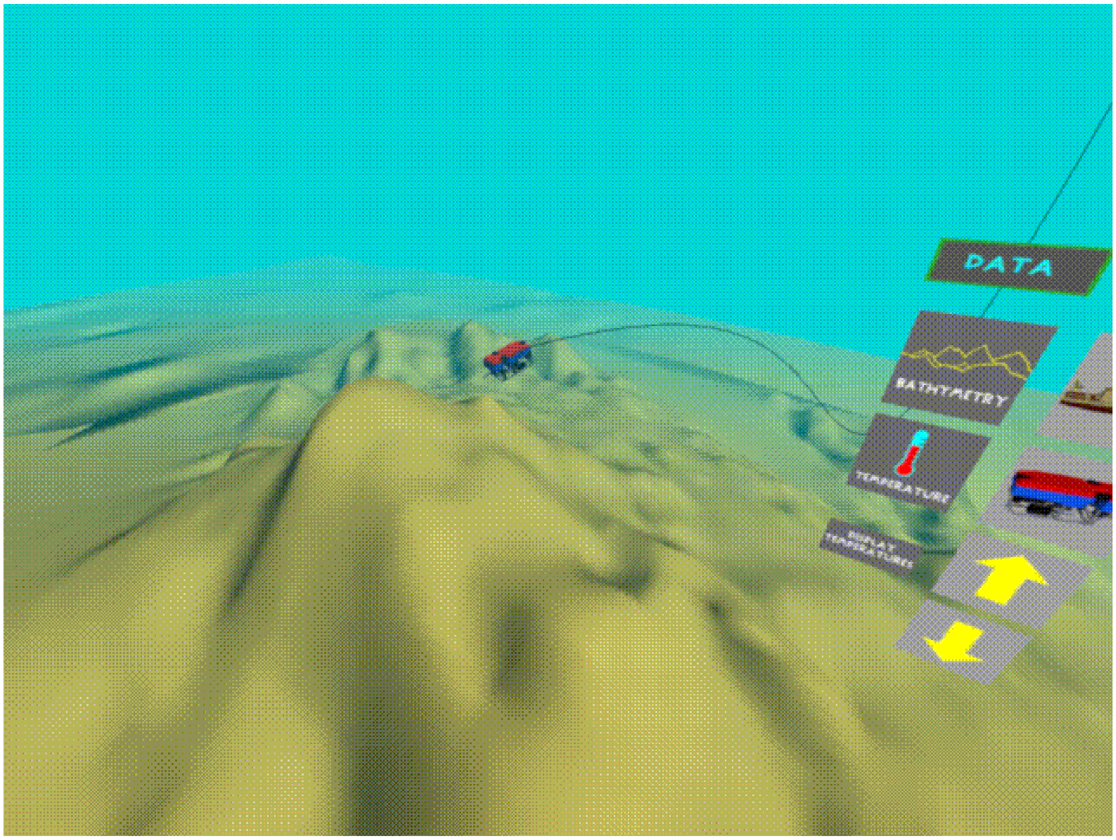

Figure 2.19. *Jason* ROV mission playback from JASON Project 94 operating in an immersive CAVE environment at SIGGRAPH 94 (Feldman 94).

and imagery over the Internet was one of the first widespread scientific collaborations that employed the MBone. Remote users have been able to download visualization software to observe the progress of each mission. Visual results are documented in (Stewart 92) (Pape 93) (Rosenblum 93), including rendering of results using a walk-in immersive display room called the CAVE Automatic Virtual Environment (CAVE) (Feldman 94) (Cruz-Neira 93). Extension of these results using a comprehensive underwater virtual world has the potential to further support distance learning and scientific research objectives. The involvement of motivated and inquisitive students can doubtless increase the realism and effectiveness of an underwater virtual world.

2. Acoustic Oceanographic Sampling Network (AOSN)

A convergence of developing technologies is enabling ambitious new approaches to oceanography. Autonomous Oceanographic Sampling Networks

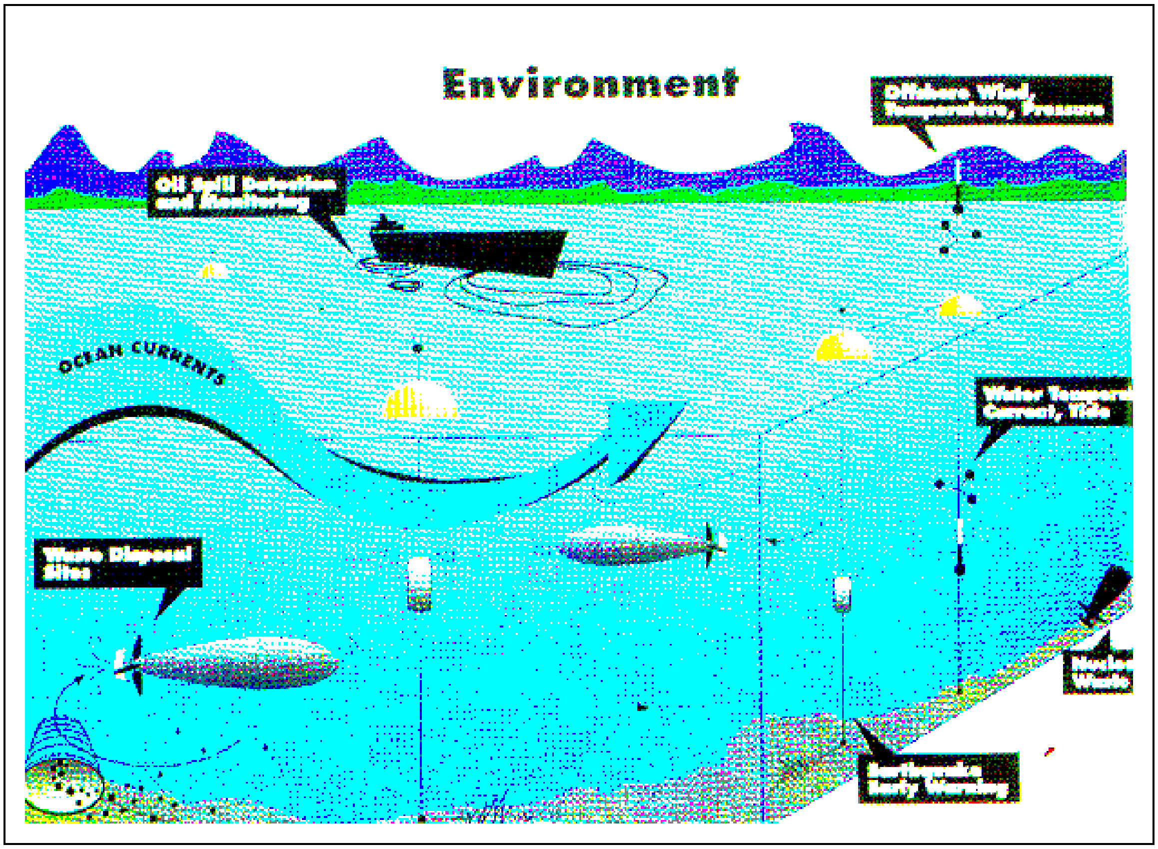

Figure 2.20. Autonomous Oceanographic Sampling Network (AOSN) environmental mission profile. Other planned mission profiles include marine operations, mineral resources and fisheries (Fricke 94) (Curtin 94).

(AOSN) are an ambitious plan for large-scale long-duration synoptic data sampling using multiple networked autonomous vehicles and sensors (Curtin 93). Untethered network connections for AUVs and underwater sensors are via acoustic modems to network nodes which relay data to shore over radio frequency (RF) links (Catipovic 93). Numerous competing design tradeoffs must be considered. AUV propulsion endurance and communications efficiency must meet energy expense per survey area criteria. The limited bandwidth and noisy acoustic channel of the water column must be effectively and reliably exploited. The physics of underwater transmission are far different than RF transmission, so packet network protocol design is not easily adapted. Currently it is not clear that multiple vehicles and sensors will effectively interconnect with the Internet. Numerous cost-effectiveness issues must be addressed simultaneously. Nevertheless it is clear that such an approach holds the promise of revolutionizing oceanographic sampling and ocean exploration. Interconnecting large numbers of information entities and diverse data products in a comprehensible fashion is an excellent application for implementation in an Internet-wide underwater virtual world.

3. MBARI-NASA Ames-Postgraduate School-Stanford Aerospace Robotics Lab (MAPS) Project

Four research institutions in the Monterey Bay region have begun a cooperative collaboration to design and build a next-generation AUV. Proposed rapid-response science missions for this AUV call for deep depth capability, single work day operating endurance between recharging, moderate cost and interchangeable mission-specific sensor suites. Use of an underwater virtual world is likely to reduce impediments to regional research collaboration, improve access to scientific data measurements, maximize utilization of shared resources and enhance a common understanding of vehicle challenges.

4. Live Worldwide Distribution of Events

Collaboration, distance learning, human interaction and communication of ideas do not magically happen when a computer is connected to the Internet. We have found that people issues and technical issues are equally important when building large open networked virtual workspaces. To improve our understanding of these issues and increase the accessibility of those worlds, we have performed an ambitious series of regional and world-wide multicast sessions using the MBone (Brutzman 94a, 94b, 94c, 94d, 94f) (Macedonia 95b) (Gambrino 94).

Regardless of whether participants are scientists, naval officers, school children or interested bystanders, it is always the same real world that we are trying to recreate virtually. Ongoing efforts to further develop the underwater virtual world will continue to narrowcast computer graphics, video, audio, hypermedia and DIS-compatible AUVs with anyone interested in participating. These events will continue to extend and strengthen the empirical basis underlying this work.

5. Monterey Bay Regional Education and the Initiative for Information Infrastructure and Linkage Applications (I^3LA)

A regional network is being planned and built which will connect researchers, educators and students throughout the tricounty Monterey Bay region via interactive multimedia, audio and video (Brutzman 94f). Named the Initiative for Information Infrastructure and Linkage Applications (I^3LA), this group project is an exciting broad-based collaboration which teams educators, scientists, business and government. We hope to fundamentally change local schools by connecting education with active ocean-related research at the individual classroom level. Our educational network design approach follows the Internet model (Gargano 94). I^3LA will give individuals at 51 different schools and research institutions interactive access to any type of live or archived media using a variety of bandwidth rates. Student ages range from kindergarten to postgraduate. I^3LA exemplars for education include daily science missions using the Monterey Bay Aquarium Research Institute (MBARI) *Ventana* ROV, Monterey Bay Aquarium (MBA) exhibits, and San Jose Technical Museum for Innovation programs. A similar regional effort which uses underwater vehicle technology as a focus to enhance science education is described in (Babb 92-93). Helping to build a regional information infrastructure with strong ties to education has benefited design of the network architecture presented in this dissertation. Current work on the underwater virtual world includes adapting the software to be suitable as

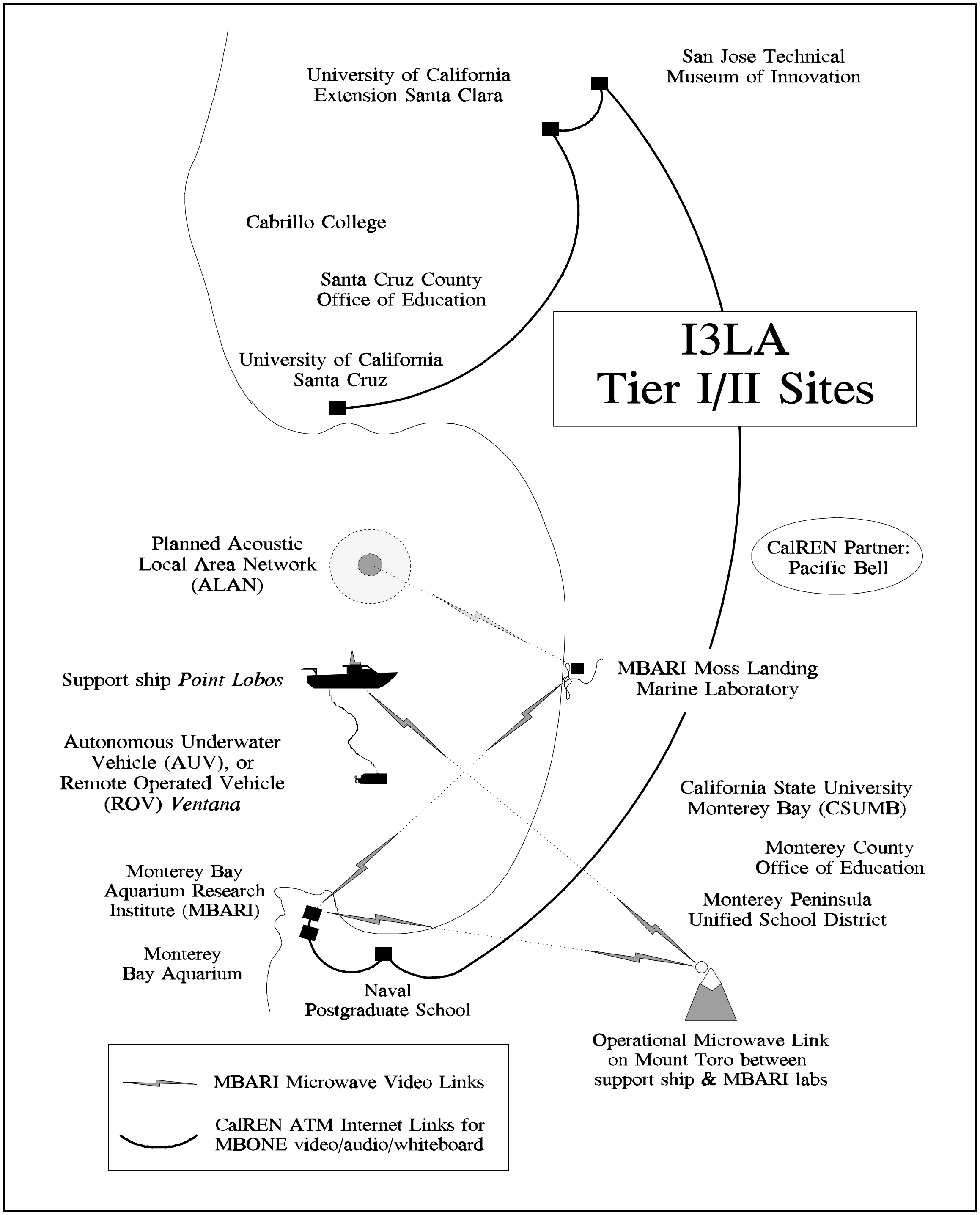

Figure 2.21. Initiative for Information Infrastructure and Linkage Applications (I^3LA) high speed communications links. Fifty one schools and research institutions are being connected.

an education application, which will further encourage extension of distributed virtual worlds as mechanisms for human interaction and information correlation.

H. SUMMARY AND CONCLUSIONS

This section presented work related to the design and construction of an underwater virtual world for an AUV. Overview summaries were provided for underwater robotics, robotics and simulation, underwater vehicle dynamics, networked communications for virtual worlds, sonar modeling and visualization, and ongoing and future projects.

Virtual reality as exemplified by immersive human-computer interface devices is a much larger albeit related field which is outside the scope of this work. Key surveys and bibliographies of virtual reality concepts, systems and trends appear in (Durlach 94) (U.S. Congress 94) (Pantelidis 94) (Emerson 94).

A number of scientific disciplines and new technological capabilities are becoming mutually compatible thanks to the multiplying effects of network connectivity. Presentation of these diverse fields under the unifying perspective of designing AUVs and virtual worlds shows that many new possibilities are becoming feasible. The review presented in this chapter shows that creation of a comprehensive networked virtual world for an autonomous robot has not been previously proposed or attempted. Following chapters will specifically show how numerous competing research objectives can be resolved and implemented to produce an underwater virtual world for an autonomous underwater vehicle.

III. PROBLEM STATEMENT AND SOLUTION OVERVIEW

A. PROBLEM STATEMENT

A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive.

B. PROPOSED SOLUTION

An underwater virtual world can comprehensively model all necessary functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot controller, enabling realistic AUV evaluation and testing in the laboratory. Three-dimensional real-time graphics are our window into the virtual world. A networked architecture enables multiple world components to operate collectively in real time, and also permits world-wide observation and collaboration with other scientists interested in the robot and virtual world.

C. AUV DEVELOPMENT DIFFICULTIES

The primary difficulty facing AUV developers is a challenging physical environment: an operating AUV is inaccessible, remote, and unattended. It is subjected to extremes of pressure, temperature, corrosion. Communications are intermittent or nonexistent. Sonar sensing is physically slower and very much different from vision. Vehicle deployment, operation and recovery are time-consuming and expensive. Vehicle physical dynamic control is very challenging. There are six spatial degrees of freedom (three dimensions each for position and rotation), not all physical control issues are solved, and there may be an unpredictable influence by ocean currents. Propulsion is costly, slow and limited. A typical vehicle only has a few hours endurance.

There is clear empirical evidence of a severe bottleneck in underwater robotics. There are thousands of indoor and outdoor land-based mobile robots, many hundreds of airborne and space-based autonomous robots, and many hundreds of underwater remotely operated vehicles (ROVs). In contrast there are perhaps a dozen working AUVs in existence, each with limited functionality. A harsh working environment and susceptibility to physical failure are among the major reasons for this scarcity. AUV failure in the ocean is unacceptable for several reasons: any failure may become catastrophic, recovery may be difficult or pointless, and replacement costs in time and money are prohibitive. We can conclude the following about AUV design: reliability, stability and autonomy are paramount, AUV constraints are often worst-case for any type of robot due to challenges inherent in the underwater environment, and many theoretical and engineering problems remain open.

D. WHY AN UNDERWATER VIRTUAL WORLD?

The broad requirements of underwater robot design provide a strong argument against piecemeal design verification. Individual component simulations are not adequate to develop effective intelligent systems or evaluate overall robot performance. A precise definition of a virtual world follows to eliminate any possible ambiguity in this term.

> Virtual world system..characteristics are seeing and interacting with distant, expensive, hazardous, or non-existent 3D environments. The technology for "seeing" is real-time, interactive 3D computer graphics and the technology for "interacting" is evolving and varied. (Zyda 92b)

An underwater virtual world for an autonomous underwater vehicle is intended to provide complete functionality of a submerged environment in the laboratory. A virtual world can provide adequate simulation scope and interaction capability to overcome the inherent design handicaps imposed when building a remote robot to operate in a hazardous environment. Construction of a virtual world for robot development and evaluation is hereby proposed as a necessary prerequisite for successful design of a complex robot which operates in a hazardous environment, such as an AUV.

A virtual world used to recreate every aspect of the environment external to the robot must also include robot sensors and analog devices (such as thrusters and rudders) which are impossible to realistically operate in a laboratory. Interactions between software processes, vehicle hardware and the real world must all be comprehensively modeled and mutually consistent. Robot physical behavior and sensor interactions must be modeled and simulated exactly. The robot controller itself is directly plugged into the virtual world using normal sensor and actuator connections, either physically or logically. The difference between operation in a virtual world or an actual environment must be transparent to robot software in order to be effective.

The current underwater robot development paradigm is inadequate and costly. Piecemeal design verification and individual component simulations are not adequate to develop and evaluate sophisticated artificial intelligence (AI)-based robot systems. Virtual world systems provide a capability for robots and people to see and interact within synthetic environments. The research goal of this dissertation is to provide complete functionality of the target environment in the lab, providing adequate simulation scope and interaction capability to overcome the inherent design handicaps of classical simulation approaches. AUV underwater virtual worlds may break the AUV development bottleneck.

E. AUV UNDERWATER VIRTUAL WORLD CHARACTERISTICS

The underwater virtual world must recreate the complete environment external to the robot. Robot physical dynamics behavior must be correctly reproduced, since underwater vehicles are prone to nonlinear dynamic instabilities and unpredicted physical responses may result in vehicle loss. Robot sensors and analog devices must be also modeled accurately. To minimize sources of simulation error, an exact copy of robot hardware and software is plugged into the virtual world using physical or logical sensor and actuator connections. The difference between operation in a virtual world or an actual environment must be transparent to the robot software. Finally, successful implementation of a virtual world can be quantitatively validated by identical robot performance in each domain. This is a type of Turing test from the robot's perspective: if robot performance is identical in each domain, then the virtual world is functionally equivalent to the real world.

Numerous component models make up the virtual world. Principal among them are a six degree-of-freedom hydrodynamics model and geometric sonar model. All models must interact with the robot in real time. Additionally, to be fully effective, the virtual world needs to provide connectivity to viewers at any location for remote observation and participation. A carefully constructed set of network connections enables all of these goals to be met simultaneously.

The overall structure of the AUV underwater virtual world software architecture is illustrated in Figure 3.1. This architectural structure diagram is very broad and is intended to show how many component models can work together. Most virtual world components have been implemented in this dissertation, demonstrating the soundness, validity and scalability of the resulting virtual world.

F. NETWORKING

Distribution of underwater virtual world components enables scalability and real-time response. A distributed approach also minimizes dependence on unique (or hard-to-replace) hardware and software. A standard point-to-point socket connects the robot and the virtual world allowing rapid and direct two-way interaction. The IEEE Distributed Interactive Simulation (DIS) protocol (NPS implementation version 2.0.3) is also used for compatible interaction with other virtual worlds and users listening on the Internet (IEEE 93) (Zeswitz 93).

This project is an excellent application to take advantage of a high-bandwidth Internet, further extending the capabilities of multiple researchers. The network approach allows many individuals dynamic remote access, which is demonstrated by Multicast Backbone (MBone) transmission of video, graphics, sound and DIS reports

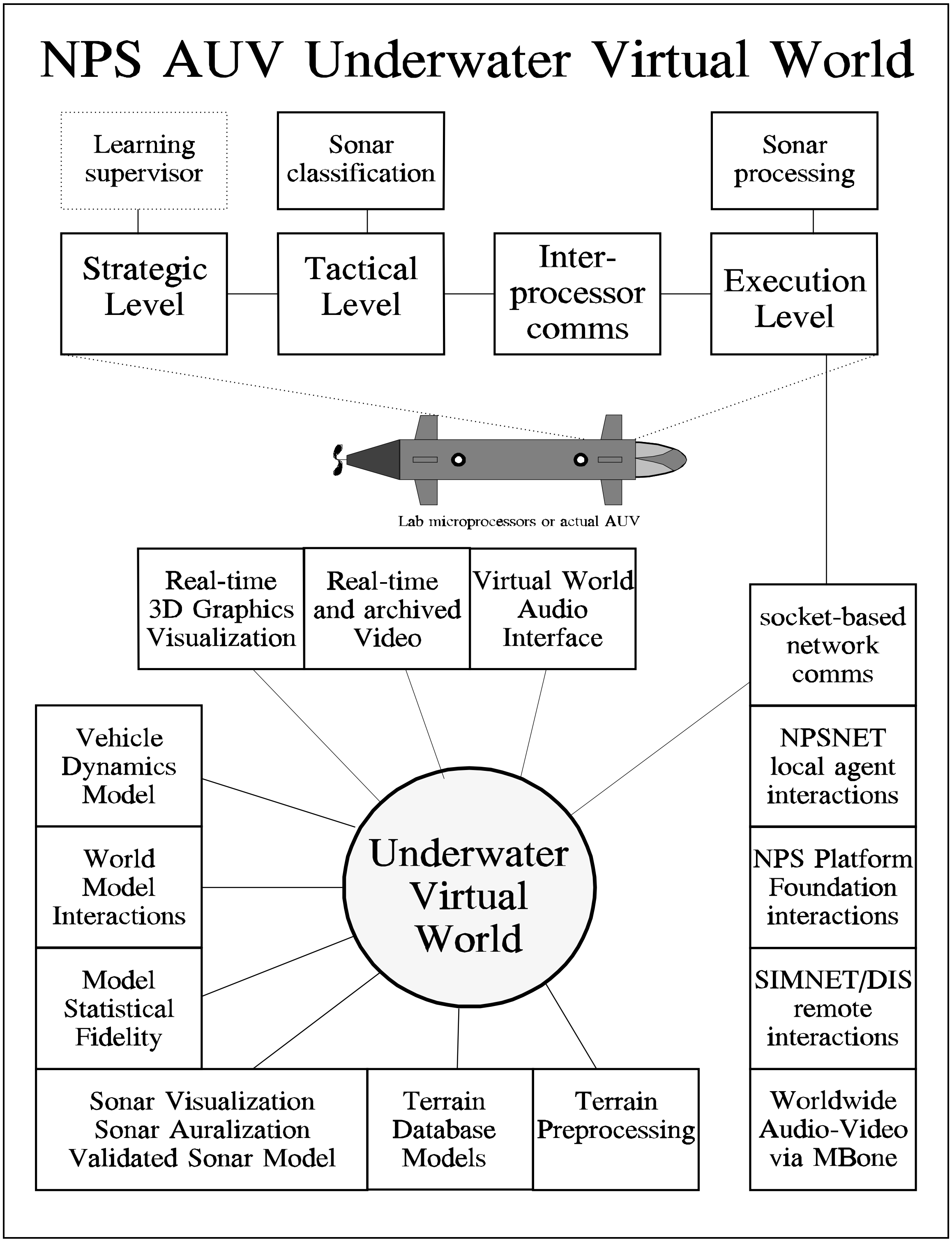

Figure 3.1. NPS AUV underwater virtual world software architecture.

for collaboration with other participants outside the site where the robot and virtual world are operating. Providing hypermedia access via publicly available World-Wide Web (WWW) network browsers such as *Mosaic* makes a complete variety of pertinent archived information available to anyone. Retrievable information resources include images, papers, datasets, software, sound clips, text, speech, source code, executable programs, live or archived video, and any other computer-storable media. Together MBone and the World-Wide Web provide the infrastructure of the information superhighway, letting anyone listen in and watch your work. Addition of multicast networked DIS packets and publicly available software lets people observe an identical interactive virtual world from any location with minimum burden on the global Internet. Remote interaction by numerous players within the virtual world of robot and environment becomes feasible and even convenient.

G. IMPORTANCE OF SENSORS

Design of autonomous underwater robots is particularly difficult due to the physical and sensing challenges of the underwater environment. Robot performance is often very tightly coupled to sensor accuracy and interpretation. Emergent behavior from interaction between robot processes and the environment can only be determined through experimentation. Having valid sonar and terrain models is very valuable for robot design and testing, since sensor interactions can be repeated indefinitely. Many new research projects become possible. Machine learning based on massive repetitive training is feasible, such as the design and implementation of trainable genetic algorithms or neural networks. Potentially fatal scenarios can be attempted repeatedly until success is reliably achieved, without risk to robot, human or environment.

H. SONAR VISUALIZATION

Visualization of robot sensor interactions within a virtual world permits sophisticated analyses of robot performance that are otherwise unavailable. Sonar visualization permits researchers to accurately "look" over the robot’s shoulder or even "see" through the robot’s eyes to intuitively understand sensor-environment interactions. Similar in-depth analysis is not possible using traditional test methods such as individual software module evaluation, direct robot observation or post-mission scenario reconstruction. In particular, the overwhelming size and information content of ocean-related and robot-related datasets means that visualization is essential to extract meaning from numerous simultaneous quantitative relationships. Visualization of the robot in its surroundings greatly improves human understanding.

An initial geometric sonar model implementation demonstrates how larger-scale sonar and terrain models can fit into the underwater virtual world architecture. More detailed visualizations of environmental datasets and a general sonar model have been implemented offline. They are included to show how additional sonar visualization capabilities can extend even further the functionality of the implemented underwater virtual world. Future work in sonar and terrain includes scaling up these models for interaction using world spaces of arbitrary sizes.

I. PARADIGM SHIFTS: CONTENT, CONTEXT, AND WORLD IN THE LOOP

Within two lifetimes we have seen several paradigm shifts in the ways that people record and exchange information. Handwriting gave way to typing, and then typing to word processing. It was only a short while afterwards that preparing text with graphic images was easily accessible, enabling individuals to perform desktop publishing. Currently people can use 3D real-time interactive graphics simulations and dynamic "documents" with multimedia hooks to record and communicate information. Furthermore such documents can be directly distributed on demand to anyone connected to the Internet. In this project we see a further paradigm shift becoming possible. The long-term potential of virtual worlds is to serve as an archive and interaction medium, combining massive and dissimilar data sets and data streams of every conceivable type. Virtual worlds will then enable comprehensive and consistent interaction by humans, robots and software agents within those massive data sets, data streams and models that recreate reality. Virtual worlds can provide meaningful context to the mountains of content which currently exist in isolation without roads, links or order.

As networked virtual worlds mature they will become more robust, efficient and portable. Going past the logical conclusion of "hardware in the loop" use of robots within a virtual world, as is presented in this dissertation, eventually virtual world models will be embeddable back into the robots. Having a "world in the loop" as an embeddable component in this manner will extend the capabilities of robots to sense, interpret and interact with the real world around them. The fidelity and scope of virtual world models and representations will improve steadily as robots and humans operate interchangeably in virtual worlds and the real world.

IV. NPS AUTONOMOUS UNDERWATER VEHICLE

A. INTRODUCTION

Detailed knowledge regarding robot requirements is a necessary prerequisite for implementing robot operation in a virtual world. This chapter describes key considerations in underwater robotics hardware and software, particularly as instantiated in the NPS AUV. Familiarity with the Chapter II review of related robotics projects is recommended. An overview of generic AUV hardware and software is followed by NPS AUV hardware specifications and software characteristics. Additional overview descriptions of the NPS AUV and related research appear in (Brutzman, Compton 91) and (Healey 92a). Due to the large variety of critical tasks an autonomous underwater robot must perform, a robust multilevel software architecture is essential. The software architecture used by the NPS AUV is the Rational Behavior Model (RBM). The three levels of RBM are described with emphasis on the real-time characteristics of each level. Details are also provided regarding vehicle software developed in this work. Specific contributions of this dissertation include extending the RBM execution level and improving implemented RBM interprocess communication (IPC).

B. UNDERWATER ROBOTICS

Although there are far fewer robots designed to operate underwater than in other environments, there is much diversity in the hardware and software of those robots that exist. Underwater robot hardware is mostly concerned with watertight integrity, maneuvering and sensing. Underwater robot software is usually preoccupied with real-time hardware control. Implemented higher-level functions are rarely as sophisticated or capable as desired. Although manipulators and intervention tools are common on remotely-operated vehicles (ROVs), they remain a rarity on autonomous robots because fundamental problems of ship control, navigation and classification of detected objects are not well solved. Recent overviews of prominent AUVs and related technical problems are (Fricke 94) (Zorpette 94). The best way to understand the capabilities and weaknesses of these vehicles is to watch them in operation. High quality videotape footage and written summaries of state-of-the-art underwater robots appear in recent video conference proceedings (Brutzman 93a) (Brutzman 94a).

1. Underwater Vehicle Hardware

Unfortunately the cost in time and money of assembling an AUV is high and currently beyond the reach of most academic institutions. Nevertheless most hardware components are commercially available, particularly since the remote operated vehicle (ROV) industry is well established and thousands of ROVs have been deployed. Institutions considering building an AUV are advised to start by looking at existing ROVs and related components that can be adapted for autonomous operation.

Pressure hulls for AUVs typically fall into two categories: streamlined and open frame. Streamlined hulls are useful for operating at high speed, or minimizing drag so that propulsion endurance is maximized. Open frame hulls typically consist of a framework of piping open to the ocean, with all components bolted onto the frame wherever appropriate. At low operating speeds drag is not a significant handicap, and the open frame simplifies placement and adjustment of hardware devices.

Power supplies and propulsion endurance are a significant weak point in current AUVs. Most vehicles are powered by lead-acid or silver-zinc batteries with usable capacity ranging from several hours to about a day. Hydrogen gas generation during battery charging or discharge is a serious personnel and equipment hazard. Research and development work in improving power density has focused for a number of years on alternative battery electrochemistries, closed-cycle (self-oxidizing) engines and aluminum hydroxide fuel cells, but dramatic improvements in cost or capability are not soon expected. Eventually the active research and development of improved battery technology for electric cars and laptop computers may provide useful power supply alternatives.

Sensors are one of several key technologies that distinguish underwater robots from ground, air and space-based robots. Since the oceans are generally opaque to visible light at moderate-to-long ranges, vision-based video systems are unreliable in turbid water and are ordinarily of use only at short distances. Vision systems usually require intense light sources which further deplete precious energy reserves. In comparison to underwater computer vision, sonar (acoustic detection) has long been a preferred sensing method due to the very long propagation ranges of sound waves underwater. However, sound waves can be bent by variations in depth, temperature and salinity. A variety of problems including ambient noise, multipath arrival, fading, shadow layers, masking and other effects can make sonar use difficult. Since active sonar typically provides good range values with approximate bearing values, algorithms for sonar recognition are much different than vision algorithms. Blue-green lasers are relatively new underwater sensors that are useful since they can provide accurate range and accurate bearing data at short-to-moderate ranges with low power consumption. Other hardware sensors of interest to AUVs include pressure instruments, flow detectors, inertial navigation acceleration and angular rate sensors, and fast Global Positioning System (GPS) receivers. New and varied sensors are being developed for oceanographic survey measurements and trace chemical detection (Bales 94a, 94b).

Communications with underwater vehicles are notoriously difficult. Tethers can provide high bandwidth and even a power supply, but remain subject to entanglement and breakage with the subsequent possibility of vehicle loss. Tethers typically require tether management systems which can be very costly in their own right. Tethers also induce undesirable and varying drag forces on the underwater vehicle. Acoustic modems are a useful innovation that can provide communications links, but are very susceptible to channel noise and channel loss problems. A serious limitation in current acoustic modems is incompatibility with the Internet Protocol (IP), and further network research efforts are necessary to incorporate forward error correction (FEC) and transport protocol functionality for reliable internetworking of underwater devices. Acoustic long-baseline and short-baseline navigation can be used to determine underwater vehicle location by measuring time of flight of pings between beacons at fixed locations and a transponder located on the vehicle. Beacon pings can be further encoded to pass positional information back to the vehicle. Unfortunately, the primary limitation of navigation in an acoustic field is that beacons must be deployed beforehand in known locations around the area of interest.

2. Robot Software Architectures

Designing an AUV is complex. Many capabilities are required for an underwater mobile robot to act capably and independently. Stable physical control, motion control, sensing, motion planning, mission planning, replanning and failure recovery are example software components that must be solved individually for tractability. The diversity and dissimilarity of these many component subproblems precludes use of a single monolithic artificial intelligence (AI) paradigm.

Distributed AI usually addresses specifications and protocols between similar autonomous agents working cooperatively on global problems. Hybrid reasoning often refers to novel combinations of two or three techniques to improve overall performance when solving a single problem type. Neither definition appears suitable for general robot control. Multiple dissimilar AI processes must interact in an intelligent manner to achieve the robust capabilities and multiple behaviors needed by a mobile robot (Elfes 86). A variety of robot architectures have been proposed and developed to provide the control framework under which multiple AI processes can interact. A brief discussion of current robot architectures is therefore useful to clarify the scope of robot design issues.

Robot architectures can be classified over a spectrum that ranges from hierarchical to reactive (Byrnes 93). Hierarchical architectures can be characterized as being deliberative, symbolic, structured, "top down," goal-driven, and having explicit focus of attention. They are often implemented using backward inferencing. Hierarchical approaches typically contain world models and use planning and search techniques to achieve strictly defined goals. Hierarchical architectures tend to be somewhat rigid, unresponsive in unpredicted situations and computation-intensive. Nevertheless they remain capable of highly sophisticated performance.

Reactive architectures are subsumptive, "bottom up," sensor-driven, layered and may often be characterized by forward inferencing. Reactive architectures attempt to combine robust subsuming behaviors while avoiding dynamic planning and world models. Reactive architectures appear to behave somewhat randomly and achieve success without massive computations by using well-considered behaviors that tend to lead to task completion (Brooks 86, 90). Scaling up to complex missions is difficult. Stability and deterministic performance is elusive.

It is interesting to note that numerous robot architecture researchers have recently proposed hybrid control architectures (Kwak 92) (Bonasso 92) (Bellingham 90) (Payton 91) (Spector 91). A common theme in these proposals is integrating the long-term deliberation, planning and state information found in hierarchical approaches with the quick reaction and adaptability of subsumptive behaviors. Individual weaknesses of hierarchical and reactive architectures appear to be well-balanced by their respective strengths.

Physical stability and reliability deserve repeated mention in the context of multiple interacting processes. Control system considerations are often overlooked under the guise of simplifying assumptions that hide important real world restrictions and pitfalls. Robot survivability dictates that physical and logical behavior must always converge to a stable yet adaptive set of states. Divergence, deadlock, infinite loops and unstable dynamic behavior must be detectable and preventable. Hard real-time operating constraints on sensing, processing, action and reaction must be similarly resolved. Robotics research in other environments are expected to be pertinent and useful; for example, physical stability prerequisites become similarly important for ground robots as they progress from structured to unrestricted environments. Finally it is worth reiterating that an underwater virtual world is proposed as the best way to enable repeated testing of underwater vehicle control, stability and reliability.

C. NPS AUV HARDWARE

The NPS AUV has four paired plane surfaces (eight fins total) and bidirectional twin propellers. The hull is made of pressed and welded aluminum. The vehicle is ballasted to be neutrally buoyant at 387 lb. Design depth is 20 ft (6.1 m). A pair of sealed lead-acid gel batteries supports vehicle endurance of 90-120 minutes at speeds up to 2 ft/sec (0.61 m/sec).

A free-flooded fiberglass sonar dome supports two forward-looking sonar transducers, a downward-looking sonar altimeter, a water speed flow meter and a depth pressure cell. Five rotational gyros mounted internally are used to measure angles and rates for roll, pitch and yaw respectively. Cross-body thruster tunnels were designed and built for the NPS AUV. An inline bidirectional propeller in each thruster can provide up to 2 pounds of force (Cody 92) (Healey 94b).

Detailed specifications of all NPS AUV hardware components are presented in (Torsiello 94). An external view of the vehicle is shown in Figure 4.1 and primary internal component arrangements are shown Figure 4.2. A detailed schematic of vehicle internal components appears in Figure 4.3. A photograph showing the NPS AUV in the test tank is provided in Figure 4.4.

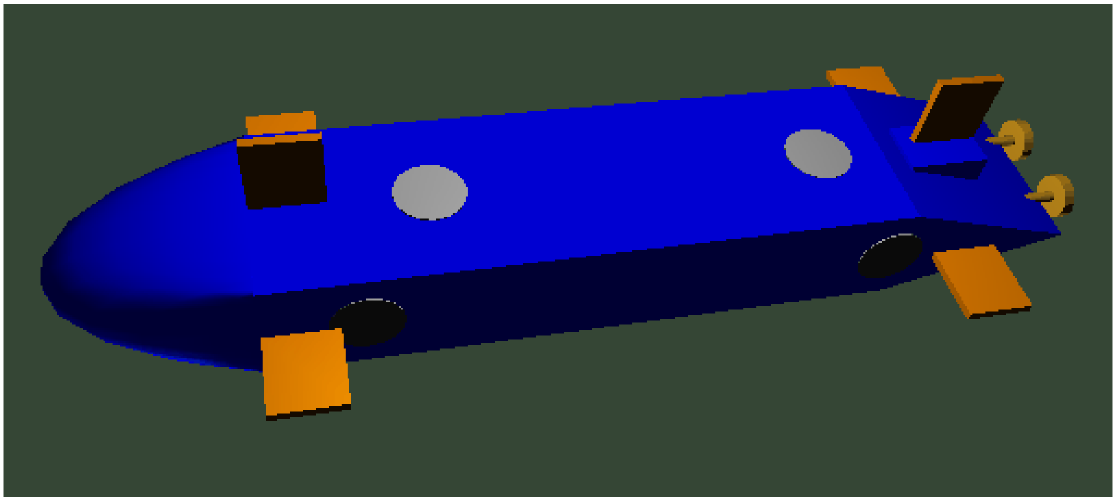

Figure 4.1. Exterior view of NPS AUV, 8 plane surfaces and twin propellers.
Length is 8’ (2.4 m), height 10" (25.4 cm), width 16.5" (41.9 cm).
Weight and buoyancy are each 435 lb (197.5 kg) when submerged.

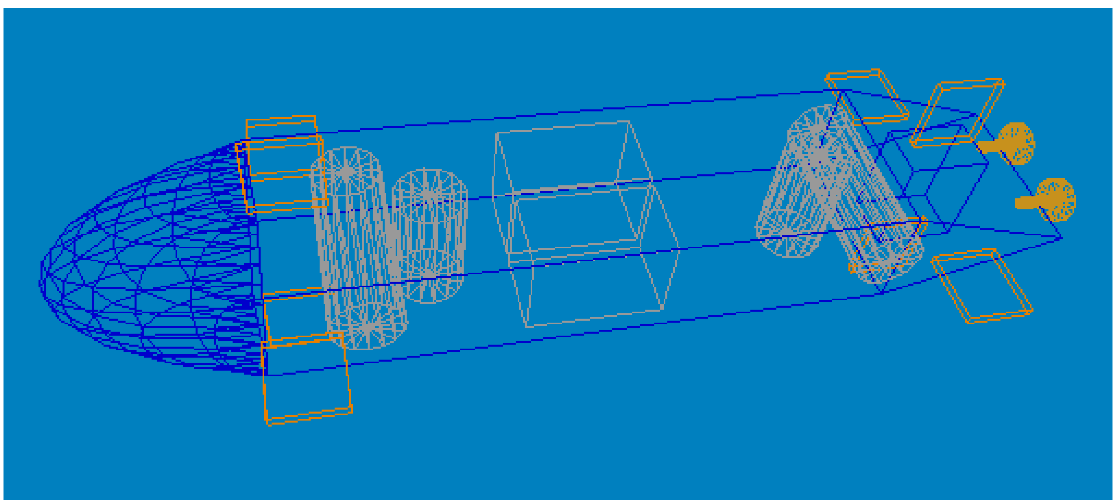

Figure 4.2. Internal view of principal NPS AUV components.
Four cross-body thrusters: two lateral and two vertical.
Two card cages contain 68030/OS-9 and 386/DOS microprocessors.

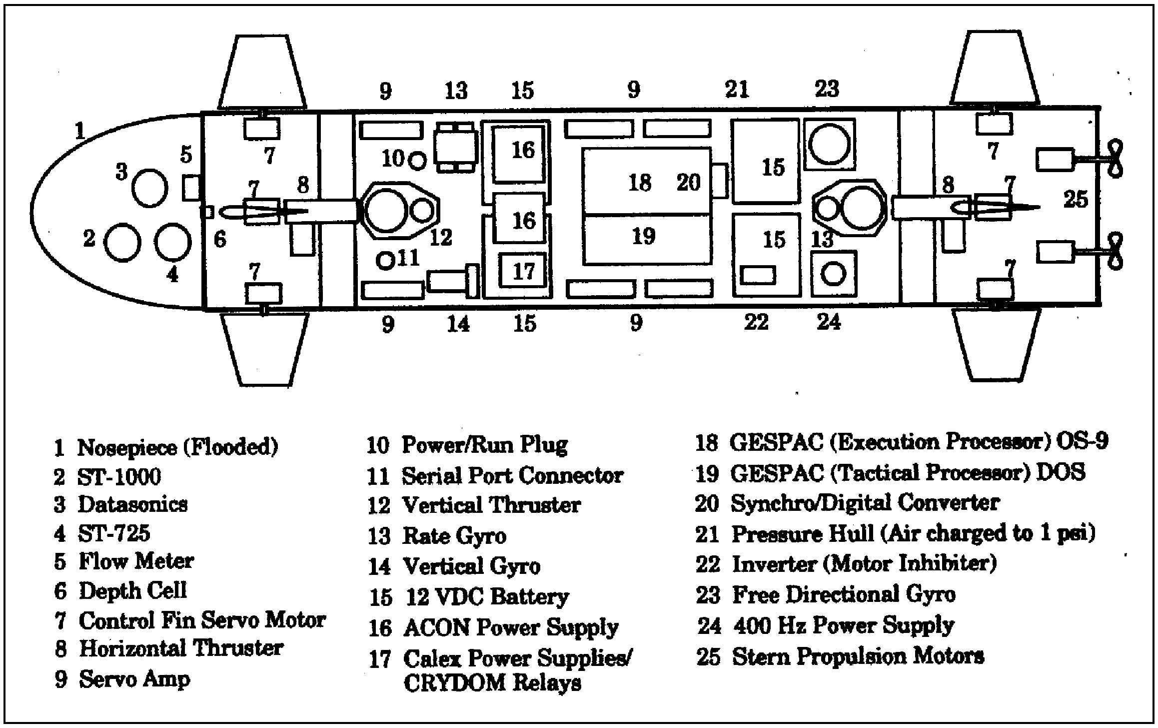

Figure 4.3. NPS AUV II internal components layout (Torsiello 94).

Figure 4.4. NPS AUV shown in test tank (Torsiello 94).

The NPS AUV is primarily designed for research on autonomous dynamic control, sensing and AI. Software control of the vehicle is provided at a high level corresponding to strategic planning and tactical coordination, as well as at a low level corresponding to hydrodynamics control of plane surfaces and propellers. Sensors are also controlled via execution level microprocessor-hardware interfaces, although some sensor functions (such as steering individual sonar transducer bearing motors) may be optionally commanded by the supervising tactical level. TRITECH sonar range resolution varies with maximum range and selected range bin size. Sonar specifications appear in Table 4.1, derived from (Torsiello 94).

Table 4.1. NPS AUV Sonar Types and Specifications.

Sonars and parameters	Tritech ST-1000	Tritech ST-725	Datasonics PSA-900
Function	Conical scan	Vertical sector scan	Depth sensing
Beam shape	1° pencil cone	24° vertical by 1° wide	10° cone
Frequency	1250 KHz	725 KHz	210 KHz
Maximum range	4..50 m	6..100 m	27 m
Range resolution	1..80 cm	4..80 cm	~ 1 cm
Steering increment	0.9° horizontal mechanical drive	0.9° horizontal mechanical drive	fixed downward not steerable
Operating modes	Sector Profile, Sector Scan	Sector Scan	Data averaging (4 ping window)
Ping rate	10 Hz	10 Hz	10 Hz
Location in bow	port below	port above	starboard below

Two microprocessors are available for use aboard the NPS AUV: a Motorola 68030 and an Intel 30386. Each is mounted on a 4" by 6" Eurobus card manufactured by Gespac Inc. The operating system for the 68030 is the OS-9 real-time operating system by Microware Inc. Support for the OS-9 operating system running on Gespac computers is problematic at best; a recommended reference for OS-9 users is (Dayan 92). Operating system for the 30386 is Digital Research (DR) DOS 6.0, chosen for small kernel size and the ability to manually switch between tasks. Multitasking operating systems such as Windows and OS/2 were earlier considered and rejected due to their insistence on graphical user interface overhead. To date the 30386 has not been used for in-water missions. A variety of different processors and operating systems are being considered for future NPS AUV configurations. Unfortunately the large volume of intricate legacy code dedicated to controlling numerous analog-digital controller cards and devices has so far precluded wholesale replacement of the current microprocessor/operating system combinations.

Vehicle designers must note that sealed lead-acid gel batteries are still susceptible to hydrogen gas generation and venting (Calder 94), which becomes an explosive hazard above 5% by atmospheric volume. Reliance on NPS AUV battery seals together with excessive recharging, mission repetition and insufficient venting resulted in a submerged hydrogen explosion in early 1994. Significant hull damage resulted and most electrical equipment was a complete loss due to flooding under power. No personnel were injured. Repairs took most of the year, but the refurbished and renamed NPS AUV "Phoenix" resumed submerged testing in October 1994.

D. NPS AUV SOFTWARE

Ongoing development of NPS AUV software has continued for over eight years. A great deal of novel software research has been conducted during this period. Underwater robot software architectures are a particular challenge because they include a great many of the hardest problems in robotics and AI over short, medium and long time scales. The principal features and lessons learned to date relating to NPS AUV software are summarized in the following sections.

1. Rational Behavior Model (RBM) Software Architecture

The Rational Behavior Model (RBM) is a trilevel multiparadigm software architecture for the control of autonomous vehicles (Kwak 92) (Byrnes 92, 93, 95). Strategic, tactical and execution levels correspond roughly to high-level planning, intermediate computational processing of symbolic goals, and direct interaction with vehicle hardware and the environment. The three levels of RBM correspond to levels of software abstraction which best match the functionality of associated tasks. Temporal requirements range from soft real-time planning at the strategic level to hard real-time requirements at the execution level, where precise control of vehicle sensors and propulsion is necessary to prevent mission failure and vehicle damage.

RBM provides an overall structure for the large variety of NPS AUV software components. A particular advantage of RBM is that the three levels are analogous to the watchstanding organization of naval ships. The strategic level matches long-range planning by the commanding officer. The tactical level corresponds to officer of the deck, navigator and officer watchstanders. The execution level corresponds to helmsmen, planesmen and sonar operators. Such analogies are particularly useful for naval officers working on this project who know how to drive ships, since it provides a well-understood partitioning of duties and a precisely defined task lexicon.

Programming paradigms are explicitly defined at each level of RBM in order to best match programming languages to objectives. Strategic level goals are typically defined and met using backwards chaining. Tactical modules are object-oriented and use message passing to communicate. The execution level is imperative. Typical languages for each level are ***Prolog***, object-oriented ***Classic Ada*** and ***C***, respectively. Variations on the strategic level have produced provably equivalent variations using forward chaining and backward chaining (Scholz 93). (Byrnes 93) implemented all three RBM levels concurrently in simulation, running under networked Unix workstations but not on the NPS AUV proper. Initial implementation efforts for this dissertation included integrating and testing a tactical level with an improved execution level, where source code for both levels compiles identically and runs compatibly either on vehicle hardware or on networked workstations.

The primary contribution of this dissertation to RBM is extensive development and implementation of the execution level (Brutzman 94e), as well as formal specification and implementation of execution level communications requirements. As predicted by (Brutzman 92a, 92c), availability of hydrodynamics and sonar models for integrated simulation during robot development have been invaluable for development of robot control algorithms. Implementations of strategic and tactical levels in previous theses have only run in isolation and have never been tested underwater due to inadequate execution level functionality (Brutzman 92a) (Byrnes 93) (Compton 92) (Ong 90) (Scholz 93) (Thornton 93) (Wilkinson 92). Completion of a robust execution level in this dissertation now permits meaningful integration of strategic and tactical RBM levels with a capable execution level.

2. Multiple Operating Systems and Multiple Programming Languages

Given the relative uniqueness and slowness of the NPS AUV microprocessors, operating systems and interfaces, it is desirable to be able to compile, run and test AUV software on a variety of platforms. The predominant computing asset available to the NPS AUV research group is Unix workstations, particularly Silicon Graphics Inc. (SGI) graphics workstations. Although Unix is not a real-time operating system, it can be made to emulate the functionality of the real-time operating system OS-9 used for the execution level, and the more common DOS operating system used for the tactical level. To date the strategic level has not been implemented in the vehicle due to lack of a workable multitasking environment on the tactical 80386 microprocessor. Several attempts to multitask strategic and tactical levels using the ***Ada*** and/or ***CLIPS*** languages were unsuccessful (Scholz 93) (Thornton 93).

The area of greatest interest to robotics researchers is developing source code that implements proposed algorithms. Standardized languages are an essential requirement for source code that is portable across multiple hardware platforms. Languages used in the NPS AUV project reflect this criteria: ***Prolog***, ***CLIPS***, ***Ada***, ***Classic Ada***, ***C*** and ***C++*** have all been used. Theoretically, compilers for different architectures will compile source code identically on each platform. In practice, successful compilation of a single version of source code by multiple compilers is a rarity. Modified compilation control *makefiles* and context-sensitive compiler directives may be able overcome variant compiler limitations (Brutzman 94e). Such an approach is essential because there then needs to be only one single version of robot code that can compile and run successfully in any appropriate environment. NPS AUV project experience has repeatedly shown that failure to insist on cross-platform compatibility leads to "versionitis" and configuration control problems which prevent research code from being successfully implemented and integrated with previous vehicle software efforts. Such failures are unacceptable.

Given that source code is written in a standardized language and compiles on all pertinent platforms, a further significant problem can occur. Although the hybrid language approach espoused by RBM provides an excellent match between software abstraction and intended functionality, getting dissimilar languages to compile, link and execute compatibly is extremely difficult. In every possible combination that we have examined and tested, implementation of hooks between languages and linking multiple language object files were not standardized. Furthermore external language hooks typically do not perform as advertised. Despite Herculean efforts, several RBM-related research efforts have failed to get different levels of RBM communicating properly due to this vulnerability.

Fortunately, we have encountered one widely available IPC technique that is likely to support any choice of programming language, operating system or hardware architecture: use of standard Berkeley Standard Distribution (BSD) sockets compatible with the Internet Protocol (IP) (Stevens 90). IP-compatible socket communications are implemented on all computer platforms, and are available as auxiliary function libraries in most programming languages of interest. Use of sockets has several added benefits: processes can run independently, interchangeably and remotely on vehicle processors or networked workstations. Current NPS AUV implementation efforts call for replacing the hard-wired and hard-coded serial and parallel port communications between processors with a network interface for each vehicle microprocessor. Building a small network internal to the vehicle eliminates specialized hardware and software communications, and does not impose a noticeable performance penalty. It also permits connecting vehicle processors and processes to any remote entity on the Internet. Even tethers between an unmanned underwater vehicle (UUV) and the surface can be Ethernet connections (Bellingham 94). The strength and numerous benefits of this approach have led us to network all possible components of AUV-related software, both internal and external.

3. Execution Level Software

The *execution* program is a new and extended implementation of the RBM execution level for the NPS AUV (Brutzman 94e). Originally based on the work of (Marco 95) and others, *execution* now includes a command language and runs identically on the laboratory AUV hardware or on a networked SGI workstation. Real-time performance for a 10 Hz control cycle was maintained in each environment. Code development on workstations enables faster compilation and provides more robust debugging tools which are nontrivial benefits for such a large program. Principal components of the *execution* program include invocation and stored mission script file commands, communications to the tactical level, communications to the virtual world, vehicle hardware interfaces, maneuvering control algorithms, standardized telemetry data recording, and supplemental mathematical functions to support computational geometry calculations. These supplemental functions include *normalize (angle)* which normalizes an angle to the range $[0..\pi)$, *normalize2 (angle)* which normalizes an angle to the range $(-\pi/2..\pi/2]$, and *atan2 (y, x)* which returns the angle to a point in the proper quadrant.

Vehicle control algorithms are implemented using either thrusters (hovering modes), planes/rudders/propellers (cruise modes) or all in combination. Control algorithms for the following behaviors are included: depth control, heading control, open-loop rotation, open-loop lateral motion, waypoint following and hovering. Control algorithms are permitted to operate both thrusters and planes/rudders/propellers simultaneously when such operation does not mutually interfere. All control code has been developed and tested in conjunction with the construction of the hydrodynamics model presented in Chapter VI. Design, tuning and optimization of control algorithms in isolation and in concert is the subject of active research (Cristi 89) (Yoerger 85, 90) (Papoulias 89, 91) (Healey 89, 92b, 93) (Fossen 94) (Marco 95) and remains an important area for future work. Control algorithm robustness is a particularly important topic since potentially fatal nonlinear instabilities are possible and vehicle reliability is paramount. Individual control algorithms created as part of this dissertation follow.

Rudder steering control equations:

$$\begin{aligned} \delta_{rudder\ bow} &= -\delta_{rudder\ stern} \\ &= k_{\psi} \cdot \text{normalize2}\,(\psi - \psi_{command}) + k_r \cdot r + k_v \cdot v \end{aligned} \quad \textbf{(4.1)}$$

Planes depth control equations:

$$\begin{aligned} \delta_{planes\ bow} &= -\delta_{planes\ stern} \\ &= k_z \cdot (z - z_{command}) + k_{\theta} \cdot \theta + k_q \cdot q - k_w \cdot w \end{aligned} \quad \textbf{(4.2)}$$

Note that planes and rudder are each constrained $\leq \pm\ 40°$ to prevent excessive deflection and subsequent reduction of control above $\pm\ 45°$. Planes and rudders are zeroed at very low forward speeds in order to eliminate synchro hunting and chattering.

Vertical thruster depth control equations:

$$thruster_{bow\ vertical} = thruster_{stern\ vertical} \\ = -k_{thruster\ z} \cdot (z - z_{command}) - k_{thruster\ w} \cdot w \tag{4.3}$$

Lateral thruster heading control equations:

$$thruster_{bow\ lateral} = -thruster_{stern\ lateral} \\ = +k_{thruster\ \psi} \cdot \text{normalize2} \ (\psi - \psi_{command}) \\ - k_{thruster\ r} \cdot r \tag{4.4}$$

Waypoint hovering mode control equations:

$$waypoint\ distance = \sqrt{(x - x_{command})^2 + (y - y_{command})^2} \tag{4.5}$$

$$waypoint\ angle = \text{normalize}(\text{atan2}(y_{command} - y,\ x_{command} - x)) \tag{4.6}$$

$$track\ angle = \text{normalize}(waypoint\ angle - \psi) \tag{4.7}$$

$$along\ track\ distance = \cos(track\ angle) \cdot (waypoint\ distance) \tag{4.8}$$

$$cross\ track\ distance = -\sin(track\ angle) \cdot (waypoint\ distance) \tag{4.9}$$

$$port, stbd\ propeller\ rpm = k_{propeller\ hover} \cdot (along\ track\ distance) - k_{surge\ hover} \cdot u \quad \textbf{(4.10)}$$

$$thruster_{bow\ lateral} = + k_{thruster\ \psi} \cdot \text{normalize2}\ (\psi - \psi_{command}) + k_{thruster\ r} \cdot r - k_{thruster\ hover} \cdot (cross\ track\ distance) + k_{sway\ hover} \cdot v \quad \textbf{(4.11)}$$

$$thruster_{stern\ lateral} = - k_{thruster\ \psi} \cdot \text{normalize2}\ (\psi - \psi_{command}) - k_{thruster\ r} \cdot r - k_{thruster\ hover} \cdot (cross\ track\ distance) + k_{sway\ hover} \cdot v \quad \textbf{(4.12)}$$

Associated *k* coefficients are all positive and appear in Figure 4.5.

4. Communications Among AUV Processes and the Virtual World

Since RBM is a multilevel architecture, communications between levels must be formally defined. Communications between robot and virtual world must also be clearly specified. Defining communications includes establishing a physical path for data transfer as well as defining the syntax and protocol of exchanged messages. Design objectives include reliability and clarity so that messages are easily created and easily understood, either by software processes or by people.

```
 ________________________________________________
 AUV execution level control algorithm coefficients
 ________________________________________________

k_psi   k_r    k_v    k_z    k_w    k_theta     k_q
1.00    2.00   0.00  15.00   2.00    4.00       1.00

k_thruster_psi      k_thruster_r    k_thruster_rotate
     0.60               5.00              2.25

k_thruster_z        k_thruster_w
   10.00                80.00

k_propeller_hover   k_surge_hover
     200.00              6000.00

k_thruster_hover    k_sway_hover
    4.00                40.00
```

Figure 4.5. Control algorithm coefficients from *mission.output.constants* file.

Two kinds of messages are defined for use by robot and virtual world. The first is the telemetry vector, which is a list of all vehicle state variables pertinent to hydrodynamic and sensor control. Telemetry vectors are passed as a string type. The second kind of messages allowed are free-format commands. Free-format command messages are also string types, starting with a predefined keyword and followed by entries which may optionally have significance depending on the initial keyword. Messages with unrecognized keywords are treated as comments. These two kinds of messages (telemetry and commands) can be used for any communication necessary among robot-related entities. Employment of string types facilitates transfer between different architectures, transfer via sockets, and file storage. String types also ensure that all communications are readable by both robot and human, a trait that is particularly useful during debugging. An open format for command messages permits any user or new application to communicate with little difficulty.

Within the AUV, the basic communications flow between execution level and tactical level is straightforward. All telemetry vectors are sent from the execution level to the tactical level, providing a steady stream of time-sensitive, rapidly updated information. The tactical level may send commands to the execution level as desired, and the execution level may return informational messages between telemetry vectors as appropriate. Nonadaptive tactical level functionality can also be provided by prescripted mission command files. Telemetry vector records and command messages are logged in separate mission output files for post-mission analysis and replay. Each of these communications message types has been implemented and tested satisfactorily (Brutzman 94e). Communication protocols between tactical level and strategic level are presented in (Byrnes 93) and are not examined here.

Specific elements of the telemetry record appear in Figure 4.5 below. Both data communications internal to the vehicle (execution and tactical levels) and data communications external to the vehicle (execution level and virtual world) utilize the telemetry vector and keyword command message conventions. Currently the data path between execution and tactical levels consists of paired simplex text streams over serial and parallel connections between the two microprocessors. The data path between the execution level and the virtual world is via an Ethernet socket. In the future, serial and parallel port data paths between the execution and tactical processors are expected to be replaced by Ethernet sockets. Figure 4.7 shows physical data paths and information flow both internal and external to the vehicle.

```
time
x      y        z         phi       theta      psi
u      v        w         p         q          r
x_dot  y_dot    z_dot     phi_dot   theta_dot  psi_dot

delta_rudder              delta_planes
propeller_port_rpm        propeller_stbd_rpm
thrusters_bow_vertical    thrusters_stern_vertical
thrusters_bow_lateral     thrusters_stern_lateral

ST1000_range    ST1000_bearing     ST1000_strength
ST725_range     ST725_bearing      ST725_strength
```

Figure 4.6. Telemetry vector elements.

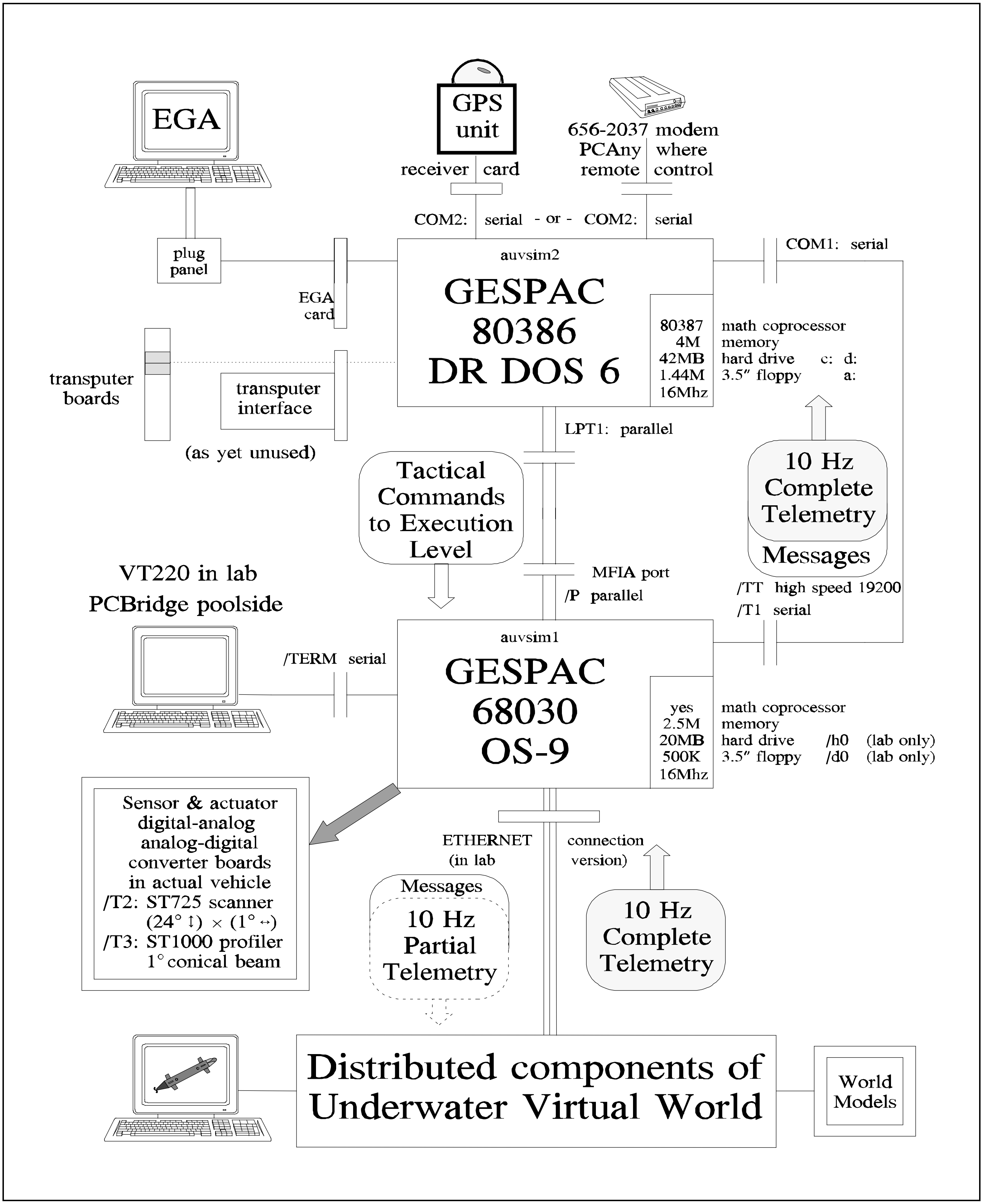

Figure 4.7. NPS AUV hardware configuration and internal interprocess communication (IPC).

The telemetry vector serves several essential purposes. In addition to providing a steady stream of information from the execution level to the tactical level, the telemetry vector also serves as the data transfer mechanism between execution level and virtual world. Efficient communications between robot and virtual world are essential if rapid real-time 10 Hz robot response is to be maintained. The telemetry record is a concise and complete way to support all of these data communications requirements.

Robot execution software is designed to operate both in the virtual world and in the real world. While sensing in the virtual world, distributed hydrodynamics and sonar models fill in pertinent telemetry vector slots. While sensing in the real world, actual sensors and their corresponding interfaces fill in pertinent telemetry vector slots. In either case, the remainder of the robot execution program which deals with tactical communications, command parsing, dynamic control, sensor interpretation etc. is unaffected. While operating in the virtual world, robot propulsion and sensor commands are communicated via the same telemetry vector. While operating in the real world, robot propulsion and sensor commands are sent directly to hardware interfaces for propellers, thrusters, planes, rudders, sonar steering motors etc. Again almost all parts of the robot execution program are completely unaffected by this difference.

The telemetry vector is therefore the key data transfer mechanism whereby vehicle operation remains transparent and identical either in the virtual world or in the real world. Telemetry vector updates also define the communication protocol between execution level and virtual world. As might be expected, the execution level program follows the common robotics cyclic paradigm of *sense-decide-act*. Figure 4.8 shows in detail how the flow of control proceeds and the telemetry vector is modified during each *sense-decide-act* cycle. Figure 4.9 provides an overview of the telemetry vector update sequence as an alternate means of portraying the validity of this approach.

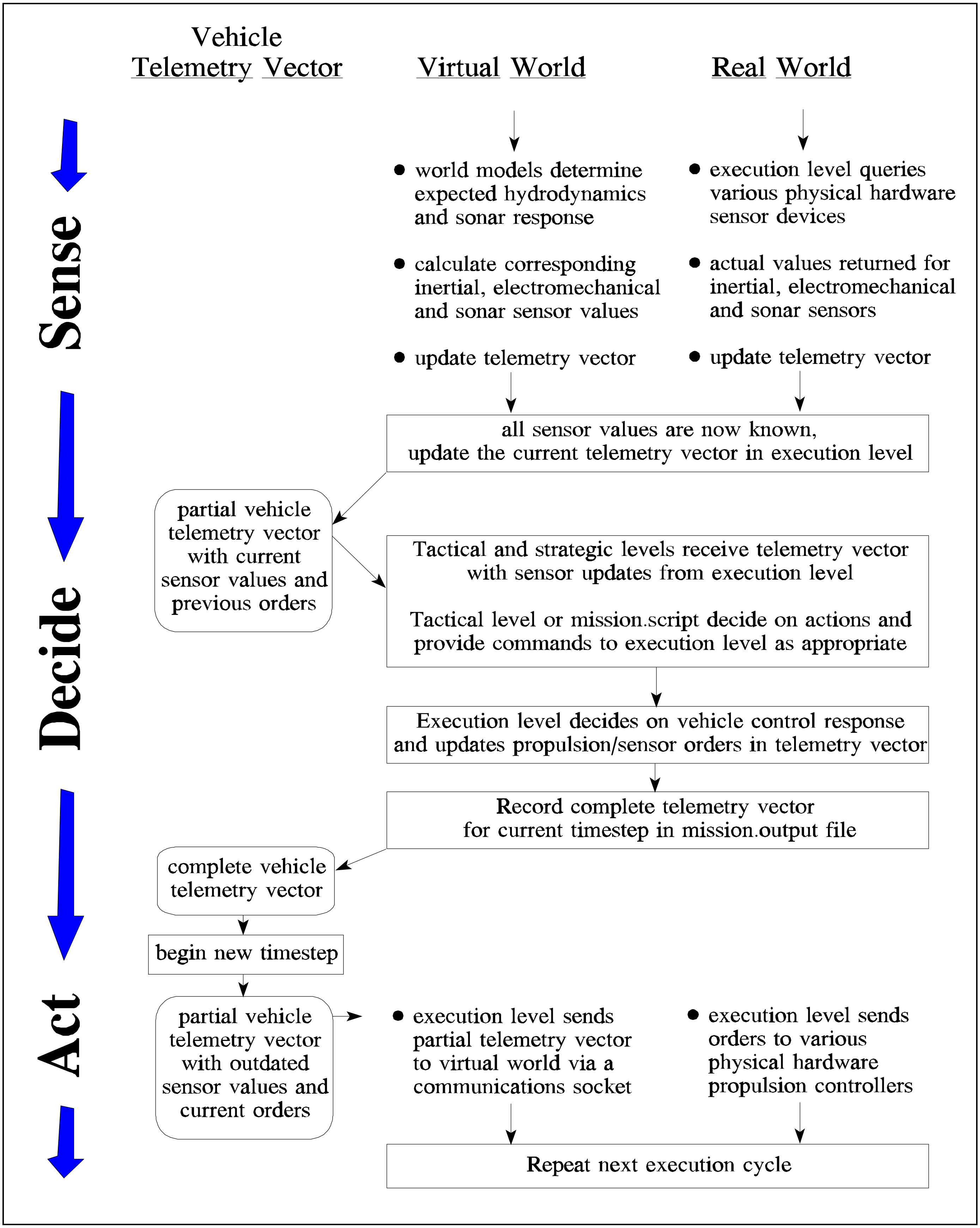

Figure 4.8. Data flow via the telemetry vector during each *sense-decide-act* cycle.

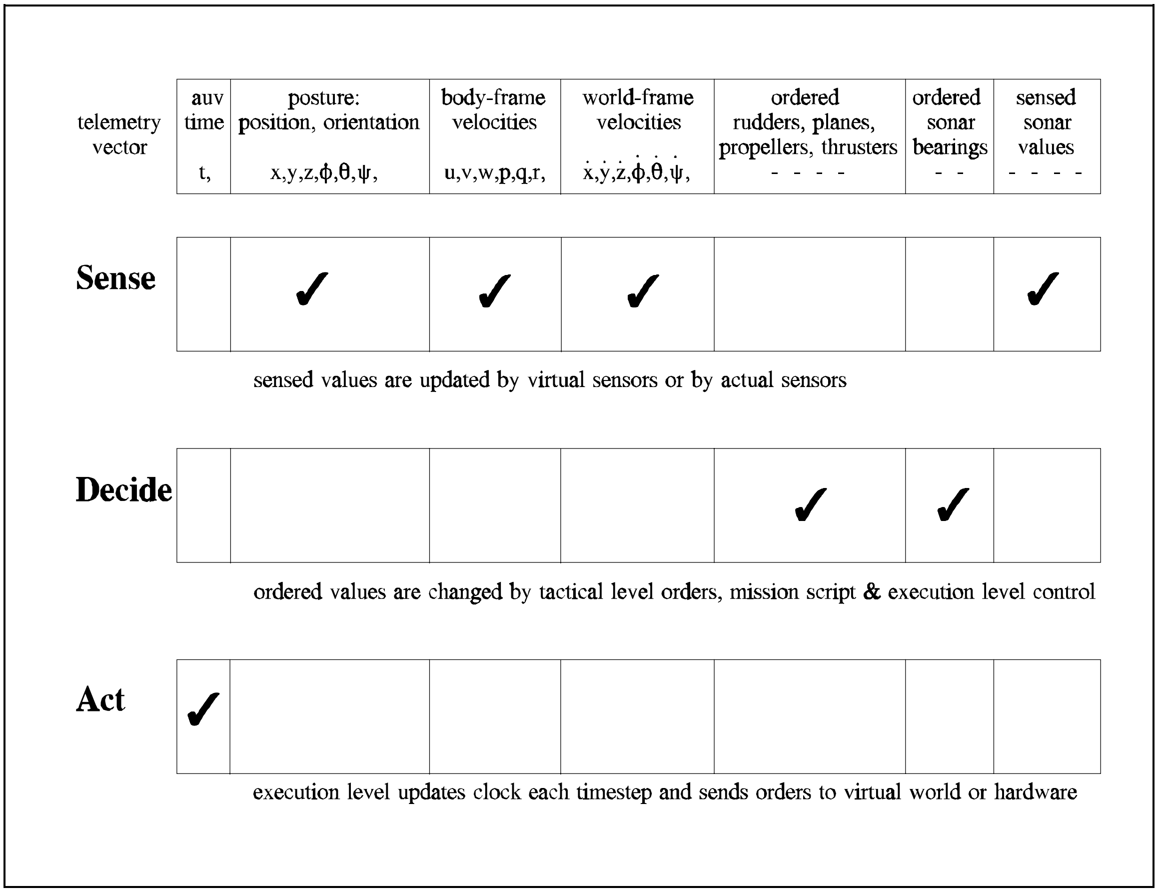

Figure 4.9. Telemetry vector modifications during each *sense-decide-act* cycle.

E. SUMMARY AND FUTURE WORK

This chapter discussed general underwater robotics hardware considerations and software architectures. Hardware specifics of the NPS AUV are outlined. Descriptions of NPS AUV software focus on the Rational Behavior Model (RBM) architecture. Multiple operating system and multiple programming language strengths and drawbacks are presented from the perspective of several years of implementation. Significant contributions to NPS AUV execution level functionality are described in detail, including a tactical command language and multiple dynamics control algorithms. Specifications are then presented for communications between execution level, tactical level and virtual world. A combination of two record types is shown to be essential and complete: the telemetry vector of vehicle state, and free format command messages. Telemetry is particularly important as the key to real-time data transfer and interaction between the robot execution level, the underwater virtual world and distributed users observing robot operation.

Future work includes many projects. Completing vehicle repairs and duplicating test results from prior to the 1994 mishap are nearly complete. The execution level program created for this dissertation needs to be reintegrated with the repaired vehicle. Integration of GPS, internal network connections between microprocessors, implementing strategic and tactical levels in the water, and porting sonar classification algorithms are all planned or in progress. NPS AUV capabilities are nearly ready to support AI research in robot architectures, sensing, classification and planning identically in the water or in the virtual world.

V. THREE-DIMENSIONAL REAL-TIME COMPUTER GRAPHICS

A. INTRODUCTION

This chapter describes the principal characteristics needed for the creation of object-oriented graphics viewers for visualizing a large-scale virtual world. Open standards, portability and versatility are emphasized over platform-specific performance considerations in order to support scaling up to very large numbers of users, platform types and information sources. The *Open Inventor* toolkit and scene description language has all of the functionality needed, and it is described briefly. The potential integration of network connections to logically extend graphics programs is examined in detail.

B. DESIRED CHARACTERISTICS OF GRAPHICS VIEWER PROGRAMS

A good graphics toolkit for building a virtual world viewer has many requirements to fill. Rendered scenes need to be realistic, rapidly rendered, permit user interaction, and capable of running on both low end and high end workstations. Graphics programmers must have a wide range of tools to permit interactive experimentation and scientific visualization of real world datasets (Nielsen 90) (Thalmann 90). The ability to read multiple data formats is also important when using scientific and oceanographic datasets. Scientific data format compatibility can be provided by a number of data function libraries which are open, portable, reasonably well standardized and usually independent of graphics tools (Fortner 92) (Rhyne 93b). Viewer programs need to be capable of examining high-bandwidth information streams and large archived scientific databases. Thus the ability to preprocess massive datasets into useful, storable, retrievable graphics objects will be particularly important as we attempt to scale up to meet the sophistication and detail of the real world. Adequate standardization of computer graphics and portability across other platforms is also desirable but has been historically elusive.

C. *Open Inventor*

Open Inventor is an object-oriented 3D graphics toolkit for graphics applications design (Strauss 92) (Wernecke 94a). Based on the *Open GL* graphics library, *Open Inventor* provides high-level extensions to the ***C++*** (or ***C***) programming language and a scene description language. It is designed to permit graphics programmers to focus on what to draw rather that how to draw it, creating scene objects that are collected in a scene database for viewpoint-independent rendering. User-triggered events are an integral part of the graphics rendering engine in order to permit rapid interactivity. A flexible design enables programmers to employ a variety of object representations and interaction modes. Object-oriented functionality allows users to customize and extend toolkit functionality through creation of new classes, subclassing and inheritance (Wernecke 94b).

The graphics capabilities of *Open Inventor* are extensive, including most (if not all) of the functionality described in canonical computer graphics reference (Foley, van Dam 90), as well as hooks to X-Windows and Motif-compliant window functions. *Open Inventor* is well suited to build graphics viewers for interactive real-time virtual worlds. It has been used to produce the graphics viewer for the NPS AUV underwater virtual world (Brutzman 94e). Particularly important and useful capabilities of *Open Inventor* are examined in the following paragraphs.

1. Scene Description Language

The ability to store graphics objects as readable, editable files is especially appealing for the creation of large-scale virtual worlds. Since the performance of computer graphics is highly dependent on the computational complexity of scenes to be rendered, it is inevitable that truly large-scale world scene databases will eventually overload viewing graphics workstations. Such overload will occur regardless of the efficiency of viewpoint culling algorithms and graphics pipeline optimizations, unless partitionable and networked scene databases are used. Furthermore, since populating a virtual world is a task that needs to be open and accessible to large numbers of people, an open graphics data standard is needed for virtual world construction. The ability to selectively load graphics objects and scenes from files is also an important distribution mechanism which can take advantage of World-Wide Web connectivity. Thus use of *Open Inventor* scene description files permits individual workstations to act as graphics file servers, and also allows a large variety of viewers to examine individual graphics objects.

Elements in a scene graph can also be represented by icons pertaining to node type for easy reference. An abbreviated scene graph for the NPS AUV graphics object file appears in Figure 5.1.

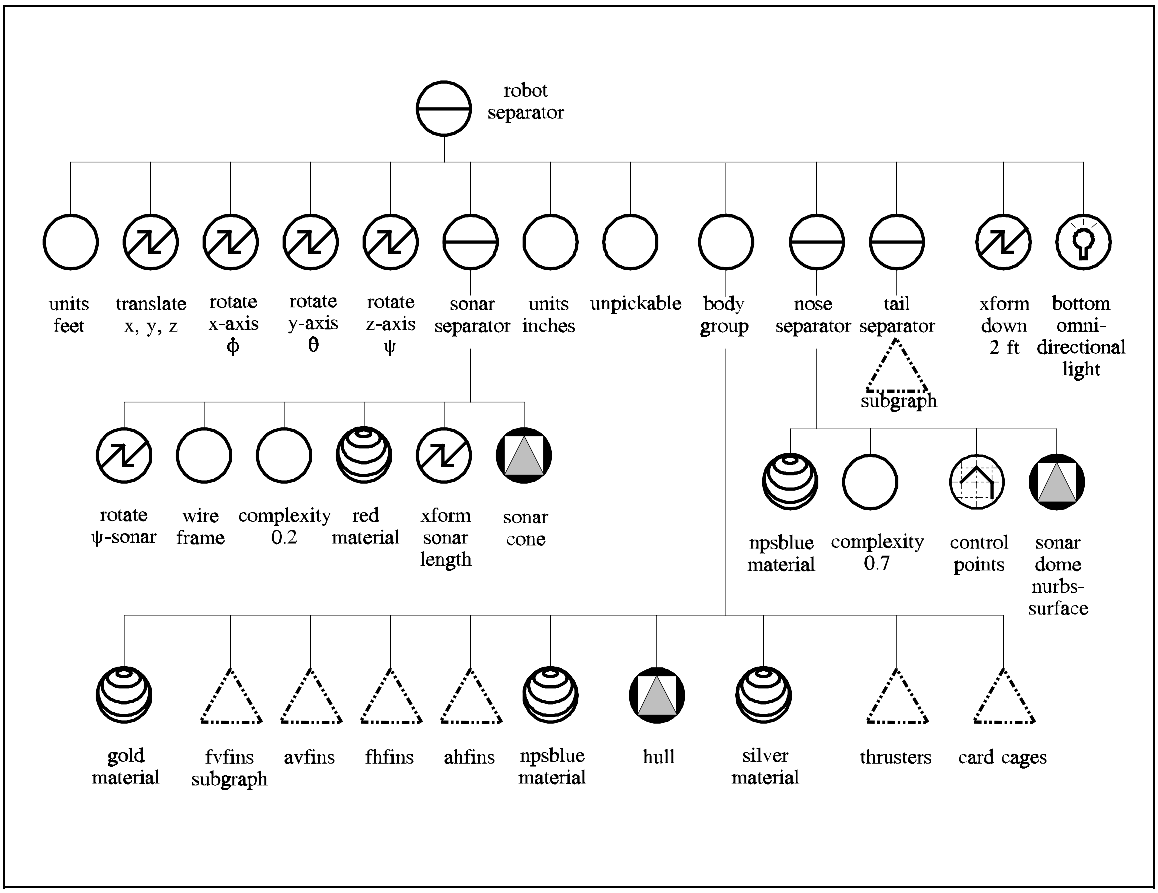

Figure 5.1. *Open Inventor* scene graph for the NPS AUV graphics model (*auv.iv*).

Open Inventor is release 2 of the Inventor toolkit and includes numerous performance improvements. A performance optimization guide and an offline scene graph optimization tool have also been included. These are both useful for tuning Open Inventor applications to achieve near-optimal graphics pipeline performance.

2. Open Standards and Portability

Silicon Graphics Inc. (SGI) is the preeminent company producing 3D computer graphics workstations and software, including *Open Inventor*. SGI has made a corporate commitment to maintain the *Open Inventor* scene description language as their preferred open standard graphics file format. Although *Open Inventor* syntax is not in the public domain like *Open GL,* SGI has further committed to maintaining backwards compatibility through future versions of *Open Inventor*. (Wernecke 94c) provides a methodology for writing translators from other scene description languages to *Open Inventor,* further encouraging nonproprietary portability among graphics models. Numerous third-party vendors are porting the *Open GL* and *Open Inventor* programming environments to other operating systems and architectures (Macintosh, Windows, Sun, HPUX etc.), further extending the expected portability of *Open Inventor* models and viewers. Ubiquitous portability for analytic, hypermedia, network, multicast and graphics tools is an extremely desirable feature for virtual world model builders. Suitability of *Open Inventor* for this role was recently underscored by an open working group examination and ballot which chose *Open Inventor* over a dozen competitors as the baseline for the draft Virtual Reality Modeling Language (VRML) specification (Pesce 94) (Bell 94).

Open Inventor file formats can be specified as ASCII (plain text) or binary format files. *Open Inventor* specifications require that the first line of any file (ASCII or binary) contain a plain-text declaration of Inventor version number for forward compatibility with current and future file readers. Binary file formats are not openly published by SGI but binary file readers are openly available, a design decision made to ensure efficient backward format compatibility in future versions. As might be predicted from an information-theoretic perspective, compressed ASCII *Open Inventor* files are about the same size as binary files. Thus compressed ASCII (i.e. human readable) *Open Inventor* files are suitable for network distribution with minimal bandwidth load. *Open Inventor* scene description files in text format (or forthcoming VRML file format extensions) are therefore excellent candidates for object definitions in a large-scale virtual world.

3. Behavior Animation through Data Sensors, Timer Sensors and Engines

Once graphics objects are specified, most graphics programmers expend a great deal of effort connecting devices, data or algorithms to animate the scene. These animation techniques are typically the heart of any graphics program and the specific reason that most graphics programs are needed, because the only way to explicitly specify behaviors is through the programming language itself. This also means that most graphics scenes are not portable as scene description files, only as programs. *Open Inventor* significantly extends the capabilities of scene description files by providing data sensors, timer sensors and behavioral "engines" which can be connected to automatically animate elements of the scene graph. Sensor and engine functionality and connections can still be written out to file, preserving behavioral connections.

Behavioral extensions to a scene description language are very useful. A simple example of engine functionality from (Wernecke 94a) is used to animate the static *JASON* ROV graphics model (which was originally donated via electronic mail). A graphics rendering of *JASON* appears in Figure 5.2. This figure moves about the base of an oil platform in the underwater virtual world. The corresponding animation scene graph is shown in Figure 5.3. Further work on extending behavioral definitions to include detailed physically-based dynamics is desirable and has been demonstrated independently (Zyda 92a).

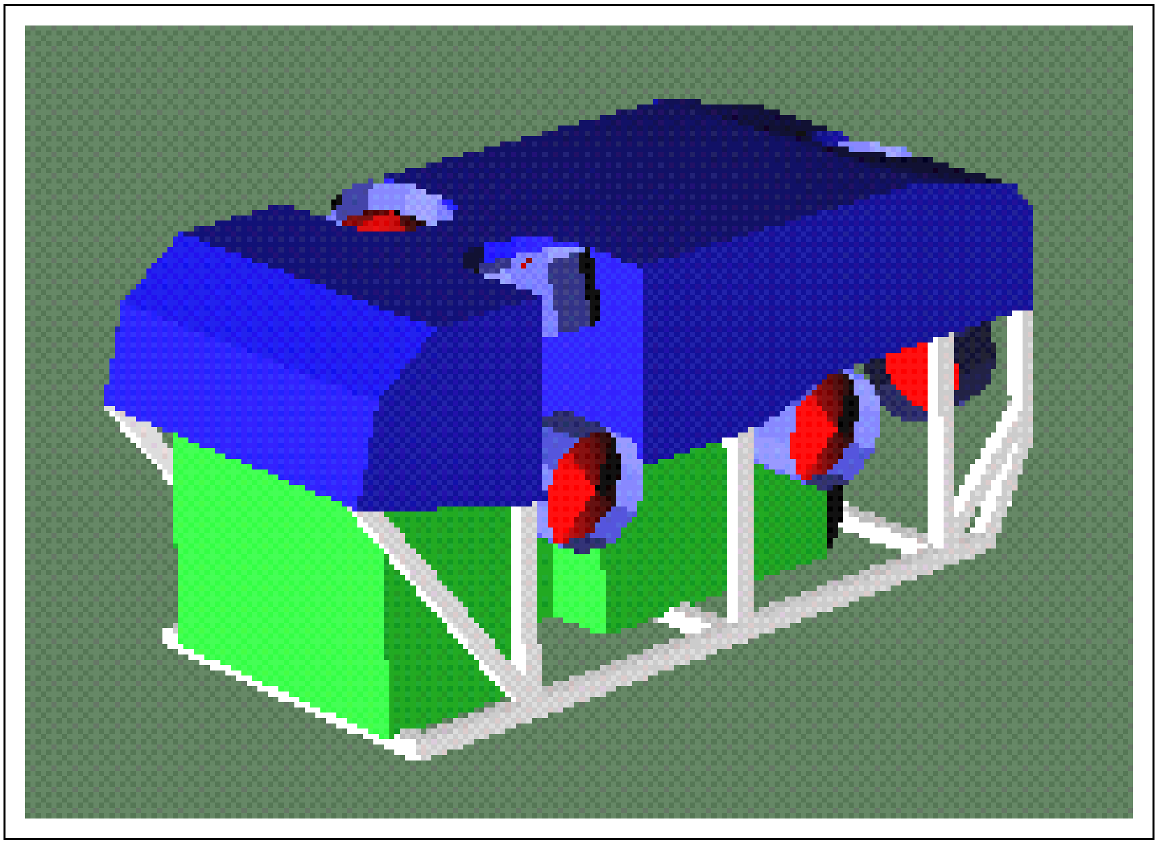

Figure 5.2. *Open Inventor* rendering of *JASON* ROV graphics model.

D. NETWORK LINKS TO GRAPHICS OBJECTS

As the use of the DIS standard becomes widespread, implementation of DIS library functionality will be more frequent and a good candidate for tool automation. Currently, vehicle graphics model connections to a DIS interface can be manually programmed or specified through initialization files within virtual world viewers such as NPSNET (Pratt 93). Increased user familiarity and availability of DIS libraries will increase the population of DIS-compatible graphics-based entities. Creation of DIS-compliant physical and graphical models is becoming progressively easier.

A recommended area for future implementation is the use of the DIS Message PDU to augment the announced arrival of new entities. The Message PDU might specify Internet Universal Resource Locator (URL) addresses which contain the

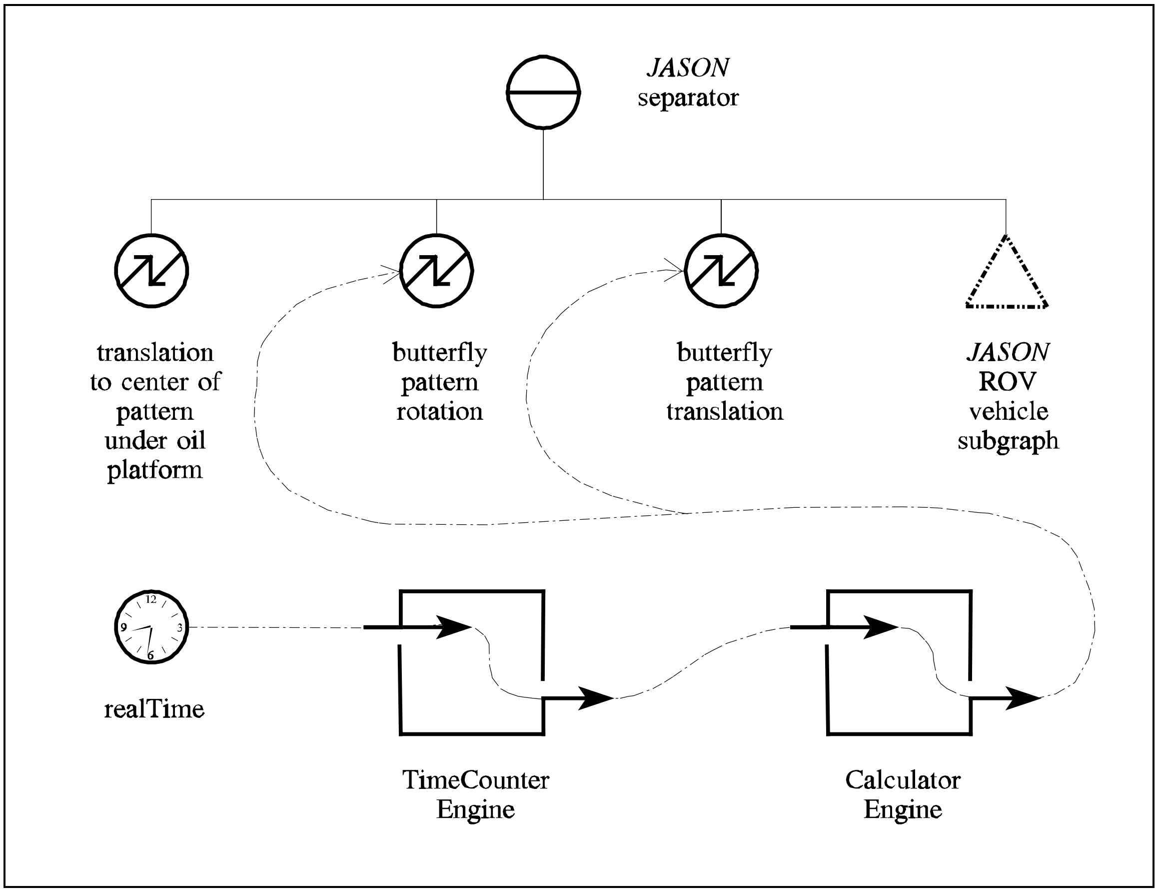

Figure 5.3. Engine animation scene graph for *JASON* ROV wandering behavior.

graphics model and operational characteristics for entity types that were previously unknown or unavailable. Such an extension permits the introduction of new DIS entities automatically without requiring pre-exercise coordination. Another interesting use of DIS Message PDUs in this underwater virtual world application might be to relay the robot commands which are being spoken to all viewers. Possible formats for this information include the original text, or a URL pointing to the synthesized audio file to minimize duplicate sound server queries.

As the use of the World-Wide Web becomes ubiquitous, the placement of graphics objects, images and datasets at well-defined network locations on public servers will become commonplace. Individual and institutional domain experts can maintain and update sophisticated world databases for open retrieval on demand.

Textures corresponding to specific locations in terrain datasets can be used to map available imagery to the real world. An example texture image manually collected from a MBARI ROV *Ventana* video stream via the MBone is shown in Figure 5.4.

Figure 5.4. Example Monterey Canyon bottom image recorded via MBone video from the MBARI ROV *Ventana*. This image is applied as a bottom texture in the underwater virtual world. Used with permission.

Widespread application of textures is particularly suited to the automatic collection of image data by robots. Automated collection and recording of video mosaics can be registered with terrain and stored on public file servers to build textured maps for large-scale virtual worlds. Extension and standardization of such approaches is also furthered by the combination of graphics and networking mechanisms proposed in the draft VRML specification (Bell 94).

E. SPECIAL METHODS

Much more work is possible to extend and augment the graphics viewer. User interface extensions will be focused exclusively on X-Windows Motif, Tk/Tcl (Osterhout 94) or hypertext markup language in order to maximize portability. Sound and data sonification can add an extra dimension to the display of scientific information. Automatically embedding hypermedia links inside scene graphs is expected to be possible using VRML extensions to *Open Inventor*, hopefully through use of embedded comments or future compatibility between the two scene graph languages.

F. SUMMARY AND FUTURE WORK

The characteristics of an open graphics viewer for underwater virtual world rendering are presented. Principal requirements include capable flexibility for scientific visualization and portability across multiple hardware and software platforms. *Open Inventor* is demonstrated as an effective programming toolkit in this regard. The desirability of scaling to very large numbers of users and information sources leads to a great deal of interesting future work which can extend graphics capabilities by embedding network capabilities. The use of multicast DIS message PDUs for distribution of World-Wide Web pointers, extending scene description languages to include dynamic behaviors and the proposed functionality of the Virtual Reality Modeling Language (VRML) are especially promising possibilities.

VI. UNDERWATER VEHICLE DYNAMICS MODEL

A. INTRODUCTION

Underwater vehicle design and construction is almost completely preoccupied with environmental considerations. The ocean completely surrounds the vehicle, affects the slightest nuance of vehicle motion and poses a constant hazard to vehicle survivability. Many of the effects of the surrounding environment on a robot vehicle are unique to the underwater domain. Vehicles move through the ocean by attempting to control complex forces and reactions in a predictable and reliable manner. Thus understanding these forces is a key requirement in the development and control of both simple and sophisticated vehicle behaviors. Unfortunately, the underwater vehicle development community has been hampered by a lack of appropriate hydrodynamics models. Currently no single general vehicle hydrodynamics model is available which is computationally suitable for predicting underwater robot dynamics behavior in a real-time virtual world.

The intended contributions of the hydrodynamic model in this dissertation are clarity, analytical correctness, generality, nomenclature standardization and suitability for real-time simulation in a virtual world. Many interacting factors are involved in underwater vehicle dynamics behavior. These factors can result in oscillatory or unstable operation if control algorithms for heading, depth and speed control do not take into account the many complex possibilities of vehicle response. Laboratory modeling of hydrodynamics response to underwater vehicle motion is essential due to the need to avoid control law errors, sensing errors, navigational errors, prematurely depleted propulsion endurance, loss of depth control, or even catastrophic failure due to implosion at crush depth. An analytically valid hydrodynamics model must be based on physical laws and sufficiently accurate for the study and development of robust control laws that work under a wide range of potential vehicle motions. The real-time hydrodynamics model is therefore an essential component of a robot-centered underwater virtual world.

Detailed analysis of underwater vehicle hydrodynamics behavior is beyond the current state of the art using real-time simulation techniques. In many cases, detailed data on underwater vehicle hydrodynamics response is unavailable even in real world test programs. Development of a general physics-based real-time model fills a gap in the robotics and simulation literature that does not exist for corresponding robot operating environments such as indoors, space or air. Inclusion of an analytically correct and verifiable hydrodynamics model in an underwater virtual world will permit meaningful and timely analysis of realistic robot-environment interactions.

It must be noted that "correctness" may not be rigorously possible for any hydrodynamics model. So many interrelated factors are present that precise testing and verification of all parameters is unlikely or impossible. While a model of hydrodynamics forces may never be perfect, it can achieve sufficiency in that vehicle responses can be predicted by physical laws at a level of detail adequate to develop, test and evaluate vehicle performance under a variety of control laws. The quantifiable goal for correctness in this work is a generalizable model that predicts vehicle physical response with sufficient rapidity and accuracy to permit equivalent robot behavior whether in the laboratory or underwater. Such a model will also enable realistic and repeatable design and evaluation of vehicle control systems, again either in the laboratory or underwater.

The model presented herein was intentionally developed with complete independence from classified U.S. Navy research on vehicle hydrodynamics. No classified documents were consulted during the literature search for this work. Research statements and conclusions are derived solely from the extensive open literature on hydrodynamics, in no way confirming, denying or implying the existence of similar work in the classified arena. A good unclassified tutorial on the fundamentals of naval submarine design is (Jackson 92). Details on experimentally and analytically developing submarine hydrodynamics models are in (Huang 88).

Unclassified overview descriptions of the use of a hydrodynamics model in the design and testing of the ARPA/Navy Unmanned Underwater Vehicle (UUV) are in (Pappas 91) and (Brancart 94). Primary references for this dissertation are (Healey 92c, 93) and (Fossen 94). Finally, a large number of papers and theses have been written at the Naval Postgraduate School (NPS) pertaining to hydrodynamics modeling of the NPS AUV, and each contributed to the theoretical and experimental knowledge presented here (Papoulias 89) (Cristi 89) (Jurewicz 90) (Zyda 90) (Warner 91) (Bahrke 92) (Brutzman 92a) (Brutzman 92c) (Cooke 92a, 92b) (Cody 92) (Brown 93) (Belton 93) (Haynes 93) (Zehner 93) (Cottle 93) (Torsiello 94) and (Marco 95).

This chapter begins with a comparison of dynamics considerations for different vehicles and their respective environments. A description of world and body coordinate systems is used to derive Euler angle kinematics equations of motion. A rigorous real-time six degree-of-freedom hydrodynamics model is then derived based on the work of (Healey 92c, 93) (Fossen 94) and others. Verified coefficient values for the NPS AUV II *Phoenix* vehicle are included and experimental model coefficient determination for other vehicles is considered. Different representations for calculating vehicle motion are compared using Euler angle or quaternion methods. Network protocol considerations are then examined for integration of the hydrodynamics model into a wide-scale distributed virtual world using the Distributed Interactive Simulation (DIS) protocol. A general object-oriented networked underwater rigid body class hierarchy is presented. Simulation of on-board sensors is considered, and the relationship of robust control system design to hydrodynamics modeling is briefly examined. Finally, future work is discussed concerning tether dynamics, ocean current modeling and collision detection. and addition of hydrodynamics models to on-board robot autopilots.

B. COMPARISON OF DYNAMICS FOR GROUND VEHICLES, AIR VEHICLES, SPACE VEHICLES, SURFACE SHIPS AND UNDERWATER VEHICLES

Dynamics models are available for a wide variety of vehicles and articulated bodies (Fu 87) (Greenwood 88) (Wilhelms 91) (Green 91) (Barzel 92) (Witkin 93). In every case it is desirable that the physical laws governing vehicle interaction with its environment be specified as exactly and correctly as possible. Constraints on vehicle motions vary greatly during interaction with different environments. A brief examination of each basic type of vehicle environment and the physics associated with those environments is useful in understanding the nature of hydrodynamics modeling.

One well-specified objective of the hydrodynamics model is to repeatedly determine system state, defined as follows:

> The *state* of a system is a mathematical structure containing a set of n variables $x_1(t), x_2(t), ..., x_i(t), ..., x_n(t)$, called the *state variables*, such that the initial values $x_i(t_0)$ of this set and the system inputs $u_j(t)$ are sufficient to uniquely describe the system's future response for $t \geq t_0$. There is a minimum set of state variables which is required to represent the system accurately. The m inputs, $u_1(t), u_2(t), ..., u_i(t), ..., u_n(t)$, are deterministic; i.e., they have specific values for all values of time $t \geq t_0$. (D'Azzo 88)

An alternative definition of state is the minimum set of variables from which the position, orientation and combined kinetic and potential energy of the vehicle can be determined uniquely. Unique descriptions of vehicle state also require inclusion of an accompanying dynamics model, consisting of an equal number of simultaneous equations as there are state variables expressed in list order form. One further clarification of the quoted definition is that input forcing functions need not be deterministic and can be stochastic.

A key characterization of any set of dynamics laws is whether the system is holonomic or nonholonomic. These two terms are frequently misunderstood and merit definition here. *Holonomic* describes motion that includes no constraints between any of the independent state variables of a rigid body; literally, the motion is "whole." Ordinarily, for a single rigid body, twelve state variables pertain to holonomic motion, corresponding to six physical degrees of freedom. Specifically these twelve state variables include six values for linear and rotational velocities, and six values for position and orientation (i.e. posture). *Nonholonomic* motion indicates that there are interdependent constraints on rigid body motion, or that variation in one or more of these state variables is dependent upon or constrained by other state variables. Nonholonomic constraints prevent direct integration of accelerations and velocities into posture. Examples of nonholonomic motion constraints include a rolling ball that can not slip (i.e. lose traction) relative to a surface, or parallel parking an automobile where no sideslip is allowed. Another example is a falling cat as it moves in midair, which must obey the conservation law for angular momentum. In each case, nonholonomic constraints limit the freedom of motion. Further descriptions and recent research in nonholonomic motion are examined in (Greenwood 88), (Latombe 91) and (Li, Canny 93).

1. Ground Vehicles

Ground vehicles are constrained by contact with a surface that generates normal and frictional forces between vehicle and terrain. On a surface that is predominantly planar, high frequency vertical components of motion are relatively small contributors to horizontal motion, particularly since they may be intentionally damped or compensated for by mechanical devices such as shock absorbers, tire wheels, suspension systems or flexible legs. Vertical forces merely displace the vehicle a small and independent amount in the vertical direction with little effect on horizontal velocity. Travel up and down hills can add a vertical component to the direction of motion but does not fundamentally change the two-dimensional nature of vehicle travel relative to the surface. Often simple kinematic models suffice for wheeled robots (Alexander 90), especially when surface vehicle motion is slow and constrained to follow roads and tracks when outside or flat floor surfaces when indoors. Legged robot interactions with surfaces are complex and require dynamics models (Frank 69) (McGhee 79) (Raibert 86).

Ground vehicle motion is complicated by operation at the interface between two media: ground and atmosphere. Aerodynamics loading is usually secondary, but must be considered during high wind conditions or in conjunction with the response at high relative speeds between robot and ground. Detailed analysis of the mechanisms governing vehicle interaction with various surface types is extremely complex, particularly during traversal of rough terrain by off-road vehicles (Bekker 56) (Bekker 69). Fortunately for most robot operations, however, the dynamics of ground-vehicle interaction rarely has a direct bearing on vehicle stability, reliability, navigation or higher-level control functions. Ground robots may further attempt to take advantage of ground contact for navigational purposes by measuring wheel rotation, frictional contact or leg motion (MacPherson 93). In this overall robotics context, regardless of how motion is estimated, ground vehicle dynamic behavior is often well approximated by kinematic models, with dynamics considerations typically having only secondary effects on robot control logic. Ground vehicles remain highly constrained by the nonholonomic nature of contact between vehicle and environment.

2. Air Vehicles

Air vehicles differ from ground vehicles in that vertical components of motion are coupled to interactions in the local horizontal plane. Interactions with the atmosphere due to aerodynamic forces have a significant effect on vehicle motion. There is no direct constraint on air vehicle posture analogous to ground contact, and aircraft flight dynamics are holonomic. Fixed wing air vehicle dynamics are typically dominated by the high speed forward motion which is necessary to generate sufficient lift to carry vehicle weight. The density of air is low, and thus vehicle accelerations do not produce significant acceleration-related aerodynamic forces. This means that the atmosphere does not induce significant "added mass" effects (Yuh 90).

Helicopters differ in many respects from fixed wing aircraft. Helicopter rotors have high degrees of freedom due to multiple rotor blades, each of which have individual mechanical articulations for twist and lag. Additional degrees of freedom occur due to many factors, including flexible rotational twist of individual blades, tail rotors, optional jet assist, and airstream interactions during phenomena such as turbulence and ground effects. Despite this high degree of complexity, helicopter dynamics can be well specified (Saunders 75), modeled in real time (Williams 85) (Offenbeck 85) and visually verified during repeated testing.

In fixed-wing aircraft, wings support the weight of the vehicle and also support control surfaces. Although aircraft weight and balance variations can produce large effects, they are ordinarily maintained within carefully specified ranges that the wings and control surfaces can accommodate. Wing aerodynamics have been extensively studied under steady motion conditions and are easily generalizable. Thus the overall lift and drag behavior of most air vehicles can be predicted with reasonable accuracy and in real time using simultaneous differential equation solutions (Cooke 92a, 92b) (Rolfe 86). Nevertheless precise localized modeling of high-performance aircraft dynamics for design purposes does not permit general closed-form solutions. Feasible solutions for precision design include massive finite element analysis, a large-scale computational fluid dynamics (CFD) approach, and wind tunnel testing. These approaches do not suit real-time application since large-scale finite element analysis and CFD are considered computational "grand challenges" (Draper 94). Scientific visualization and virtual reality techniques have also been applied with some success in advanced aircraft design (Bryson 91). These complex advanced techniques are special cases, however, compared to the general state of the art in aircraft design. Aircraft dynamics are typically well defined, well understood, and directly verifiable through visual examination during in-flight tests and wind tunnel experiments.

3. Space Vehicles

Space vehicle dynamics are principally determined by orbital mechanics. Friction between vehicle and environment is almost non-existent, and thus the equations of motion include only gravitational, inertial and thrust effects. There are few (if any) uncertain vehicle parameters, and vehicle postures can be tracked both locally and remotely with great precision. Interestingly, ballistic missiles can be considered a special class of orbital vehicle whose path intersects the Earth's surface (Bate 71). Many summaries of spacecraft dynamics are available, including (Larson 92) (Bate 71) (Allen 91). Translation and angular movements for orbital vehicles may be counterintuitive from an everyday perspective but can be calculated exactly. Under some conditions this motion can be nonholonomic, since six degree-of-freedom space vehicles controlled by internal motors must still conserve angular momentum. If thrusters are used, spacecraft motion is holonomic. Additionally some orbital vehicles (such as an astronaut in a space suit) have a variable mass distribution and may not strictly behave as rigid bodies. Other motions at higher frequencies may exist if vehicle components are flexible, in which case detailed partial differential equation solutions are required for twist, bending, shear and axial deformation. Nevertheless, in many respects the mathematical and empirical foundations of equations predicting spacecraft motion are the best defined, best understood and most directly verifiable of any vehicle type.

4. Surface Ships

Surface ship dynamics are unconstrained in six degrees of freedom and are holonomic. The vertical component of motion is primarily determined by very large counterbalancing values of weight and displacement which keep the ship at the surface of the ocean. Vertical posture changes due to pitch and roll variations normally average to zero over long time scales, due to the hydrostatic righting moments produced by the current location of the center of buoyancy relative to the center of gravity. Equipment, personnel and overall ship trajectory are typically unaffected by the time rates of change of components of motion, either by design or seafaring practice. Changes in vertical motion are strongly affected by the changing buoyancy of the vehicle which varies as water displacement changes. As a result, a paramount criterion in ship design is that the vehicle be reliably stable and self-righting, under both normal and damaged conditions. Interactions of greatest interest between vehicle and environment usually pertain to the travel of the ship along the horizontal ocean surface. Nevertheless motion is greatly complicated by vehicle operation at the interface between two media: ocean and atmosphere. Except for sailing vessels and ships with low headway, aerodynamic forces tend to be weaker than hydrodynamic forces. Regardless of surface ship type, both sets of forces can be significant and both must be considered simultaneously.

Hydrodynamics and navigation of surface vessels are complex subjects but have been extensively studied, with a comprehensive compendium of knowledge in (Lewis 88) and more examples in (Fossen 94) (Covington 94) (Maloney 85). Predictable courses, predictable speeds, sideslip (lateral motion due to momentum during turns) and gradual smooth changes in vehicle velocity are all typical of surface ship behavior. Behavior of surface vehicle dynamic response can be tested and verified visually. Tow tank verification is also possible, but tow tank testing is expensive and is limited by two competing requirements. Test tank model designers attempt to maintain inverse proportionality constraints between the square root of model scale and maximum water speed (Froude number), along with the concurrent desirability of simultaneously maintaining drag coefficient (Reynolds number) similarity. Tradeoffs between these competing requirements are necessary when building and testing scale models. Wind and wave models can be represented by complex spectral functions that are computationally expensive and difficult to specify (Fossen 94). Nevertheless environmental disturbances can be separately computed and independently added to hydrodynamic forces based on the principle of superposition (Lewis 88) (Fossen 94). Additionally, linear models are available for wind and wave behavior which permit reasonably accurate real-time simulation (Fossen 94) (Covington 94). Models of similar or lesser complexity are also available for hovercraft vehicles (Amyot 89). In summary, modeling of surface ship dynamics is reasonably well defined, well studied and directly verifiable during testing.

5. Underwater Vehicles

Underwater vehicle dynamics may be as complex and difficult to model as any of these regimes, principally due to difficulties in observing and measuring actual underwater vehicle hydrodynamics response. Submerged vehicle motion is not constrained in the vertical direction. For some unmanned vehicles, posture must be restricted to only reach moderate pitch and roll angles. This constraint is imposed since pointing vertically or inverting can cause equipment damage or dangerous control response. Very large angles of attack between vehicle orientation and vehicle direction of motion are possible. The effects of forces and moments can all be cross-coupled between vertical, lateral and horizontal directions. Motion in world coordinates is only calculable after all effects in the body coordinate system are comprehensively predicted. Actual vehicle motion can be watched remotely only with very low precision or (more often) not at all. Tow tank testing imposes unrealistic external force constraints which are otherwise not present. The effects of the surrounding environment are relatively large and significant, so much so that the adjacent water tends to be accelerated along with the vehicle and can be thought of as an "added mass." Together these challenges make underwater vehicle physical response, guidance and control an extremely difficult dynamics problem.

There are over one hundred pertinent coefficients and variables relating to the linear and non-linear coupled effects of lift, drag, added mass and propulsion in the model of this dissertation. Although a number of these coefficients are of second-order effect or negligible importance, determination of primary coefficient values is very difficult and expensive. These problems are frequently compounded when the subject vehicle has an open frame with irregular surfaces, or when a towed tether is attached.

It is conceivable that an even more complex and fundamental model to calculate underwater vehicle dynamics might be derived than is presented here. Specifically, the Navier-Stokes fluid flow differential equations might be applied in a CFD vehicle-fluid coupled interaction model (Ren 93). Closed-form solutions for this approach do not exist, and numerical methods attempting to solve the Navier-Stokes partial differential equations in this domain tend to introduce more unknown parameters than they eliminate. This is a particular problem for UUVs which often have irregular shapes. Additionally, CFD problems are among those currently considered as computational "grand challenges" (Draper 94). Thus CFD methods are not currently suitable for real-time simulation of underwater vehicle hydrodynamics.

Many models exist for ground vehicles, air vehicles, space vehicles, and surface ships that appear suitable for real-time use in a virtual world. No complete analogous model for underwater vehicles was encountered during this research. A great many partial models of underwater dynamics exist, but all were found to suffer from incompleteness, confused nomenclature, oversimplification, or a formulation unsuitable for real-time simulation. No other models were found which combined cruise mode hydrodynamics (propellers, fins and predominantly forward velocity) with hovering mode hydrodynamics (thrusters, station-keeping, low forward motion and large angle of attack). *No rigorous general model was previously available from a single source which is computationally suitable for real-time simulation of submerged vehicle hydrodynamics*.

6. Comparison Summary

Examination of the salient characteristics of dynamics models in these many different robot environments reveals that the underwater case is very difficult to accurately specify, most difficult to verify and most critical for preventing catastrophic vehicle loss. Failure to properly predict the dynamics of ground vehicles, orbital space vehicles or surface ships at worst may result in a vehicle which stays in place and can be safely commanded. Failure of aircraft due to improper prediction of aerodynamics can be mitigated through well-developed analytic techniques, wind tunnel testing and remote human supervisory control. Failure to properly predict the dynamics of underwater vehicles can lead to overall system failure due to any number of subsequent related faults in control, sensing, navigation or power consumption. This critical vulnerability in underwater vehicle design is a contributing cause to the relative rarity of working underwater robots. Thus the rigorous and nearly complete (Healey 93) hydrodynamics model, which is compatibly described in (Fossen 94), has been fully extended and implemented here. This revised hydrodynamics model now fills a significant gap in the robotics and simulation literatures.

C. COORDINATE SYSTEMS AND KINEMATIC EQUATIONS OF MOTION

Proper definitions of coordinate systems are essential to specifying the physical behavior of vehicles in a fluid medium. There are two coordinate systems which must be understood independently and in relation to each other: world coordinates and body coordinates.

World coordinates are defined with respect to the surface of the earth, and so are sometimes referred to as *earth coordinates* or *inertial coordinates*. A variety of standardized world coordinate systems are now in common use. The world coordinate system of this model is defined by three orthogonal axes originating at an arbitrary local point at the ocean surface. North corresponds to x-axis, East corresponds to y-axis and increasing depth corresponds to z-axis as shown in Figure 6.1. These axes follow right-hand rule conventions, and are identical to (or compatible with) standard world coordinate systems defined in robotics, computer graphics, aircraft aerodynamics, naval architecture, navigation and the Distributed Interactive Simulation (DIS) protocol (Fu 87) (Foley, van Dam 90) (Cooke 92a, 92b) (Lewis 88) (Fossen 94) (Maloney 85) (IEEE 93, 94a, 94b). Conversions from a topocentric local earth coordinate frame to geocentric or geodetic coordinate systems are given in (Lin 93). Other coordinate systems are possible but remain undesirable if they do not match these important standardized conventions.

Body coordinates are defined with respect to the body of the vehicle of interest. The three axes of a vehicle are longitudinal pointing in the nominal forward direction of the vehicle, lateral pointing through the right hand side of the level vehicle, and downward through the nominal bottom of the vehicle. The origin of body coordinates for a submerged vehicle is at the half point along the symmetric longitudinal axis.

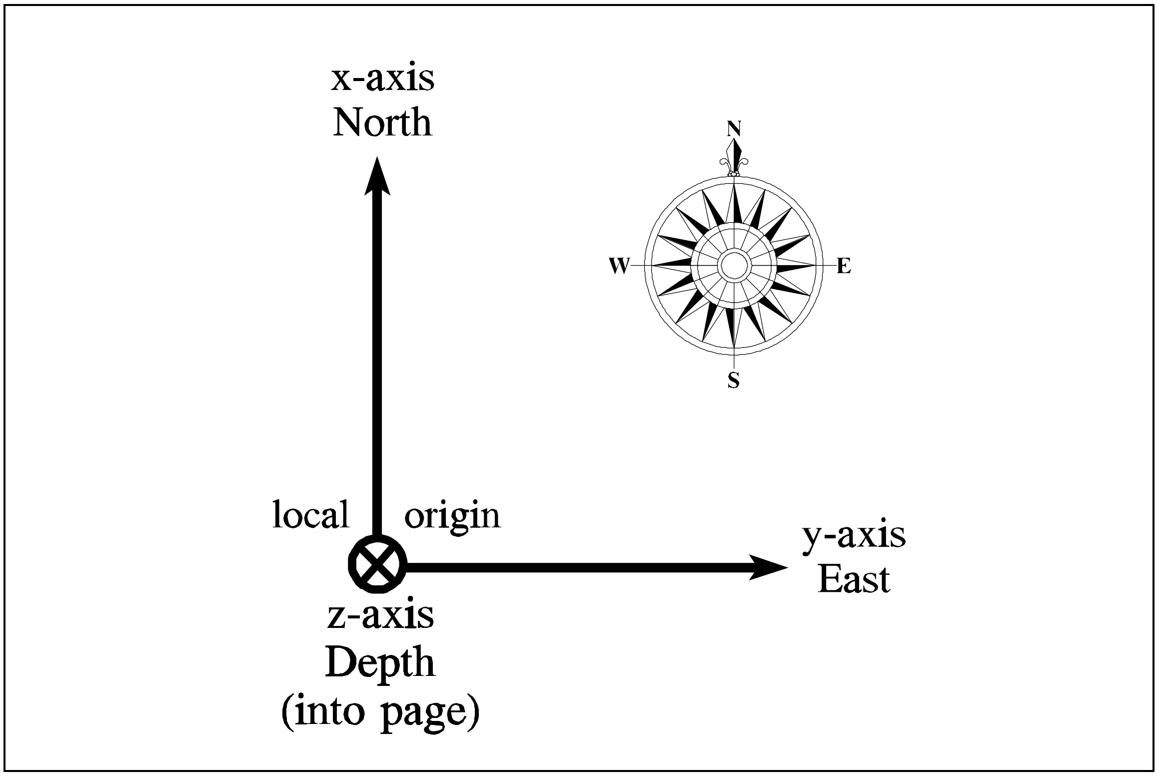

Figure 6.1. World coordinate system.

Typically this point is at or near the *center of buoyancy* (***CB***), which is the centroid of volumetric displacement of the submerged vehicle. A related location is the *center of gravity* (***CG***), which is the first moment centroid of vehicle mass. Ordinarily the center of gravity of a rigid body is the point at which net forces and moments are assumed to be applied. The center of gravity of a ship or submarine is always designed to be below the center of buoyancy to ensure static vehicle stability. The torque due to any vertical difference between the two centers ***CB*** and ***CG*** is called the *righting moment*. A nonzero righting moment results when the centers of buoyancy and gravity are not aligned vertically, tending to bring the submerged vehicle back to a neutral (typically level) pitch and roll posture. Any submerged vehicle that instead has center of gravity above center of buoyancy is inherently unstable and will tend to invert, even under static conditions.

Underwater vehicles often include free-flood spaces which can equalize with ocean pressure through small openings, all while remaining essentially contained by the hull. The water enclosed in these free-flood spaces directly contributes both to volumetric displacement and vehicle mass. Thus free-flood spaces affect buoyancy, mass, center of buoyancy, center of mass and vehicle hydrodynamics response. While submerged these effects are ordinarily static and not time-varying.

Interactions between a vehicle and the ocean environment are defined from the perspective of the vehicle, i.e. within the body coordinate system. This is because all actions and reactions between vehicle and environment are dependent on the orientation, shape, velocity and acceleration of the vehicle body, with the sole exceptions of gravity and ocean current. The direction of gravity can be sensed or estimated and is thus directly usable within the body coordinate frame of reference. Ocean current is reasonably assumed to act uniformly over the entire vehicle body. Therefore all vehicle-environment interactions can first be calculated from the perspective of the floating rigid body located inside a larger homogeneously moving ocean current frame of reference. Wind and surface wave action are normally assumed to have zero effect on submerged vehicles (if they do have an effect, then a surface ship model is likely more appropriate). Conversion from body coordinates to world coordinates consists of angular rotations to align body axes with world axes, correction for vehicle positional translation, and then addition of coordinate displacement due to ocean current motion.

Clear definition of coordinate systems greatly contributes to understanding the kinematics equations of motion. In order to reduce ambiguity, the use of (***x***, ***y***, ***z***) axis references are in world coordinates except when explicitly stated otherwise. Body axes are referred to as longitudinal, lateral and vertical, corresponding to (***x***, ***y***, ***z***) body coordinates when an algebraic description is necessary. Strictly defined variables for global coordinate frame translations and orientation rotations appear in coordinate system diagram Figure 6.2. Body coordinate frame linear and angular velocities (***u***, ***v***, ***w***, ***p***, ***q***, ***r***) are shown in Figure 6.3.

The global coordinate frame *Euler angle* orientation definitions of *roll* (ϕ), *pitch* (θ) and *yaw* (ψ) implicitly require that these rotations be performed in order. Robotics conventions usually specify physical order of rotations, while graphics conventions usually specify temporal order of rotations. Results are identical in each case. When converting from world to body coordinates using physical order (as might be specified in a three-axis gimbal system), the first rotation is for yaw (ψ) about the ***z***-axis, then pitch (θ) about the first intermediate ***y***-axis, then roll (ϕ) about the second intermediate ***x***-axis. Figure 6.4 illustrates these intermediate axes of rotation pertaining to Euler angle rotation (adapted from IEEE 94a). When converting from world to body coordinates using temporal order (as is common in computer graphics), the first rotation is roll (ϕ) about the world reference ***x***-axis, followed by pitch (θ) about the world reference ***y***-axis, and finally yaw (ψ) about the world reference ***z***-axis. Consistency of results using either method can be demonstrated by examining the mathematical order of the resulting rotation matrices, which is identical in each case. Naturally the orders of rotations are reversed if converting from body to world coordinate frame.

These Euler angle definitions are consistent with naval architecture definitions (Lewis 88). This is an important property since twelve different and unique Euler angle coordinate system definitions are possible (Fu 87), while only one Euler angle convention corresponds to naval architecture conventions.

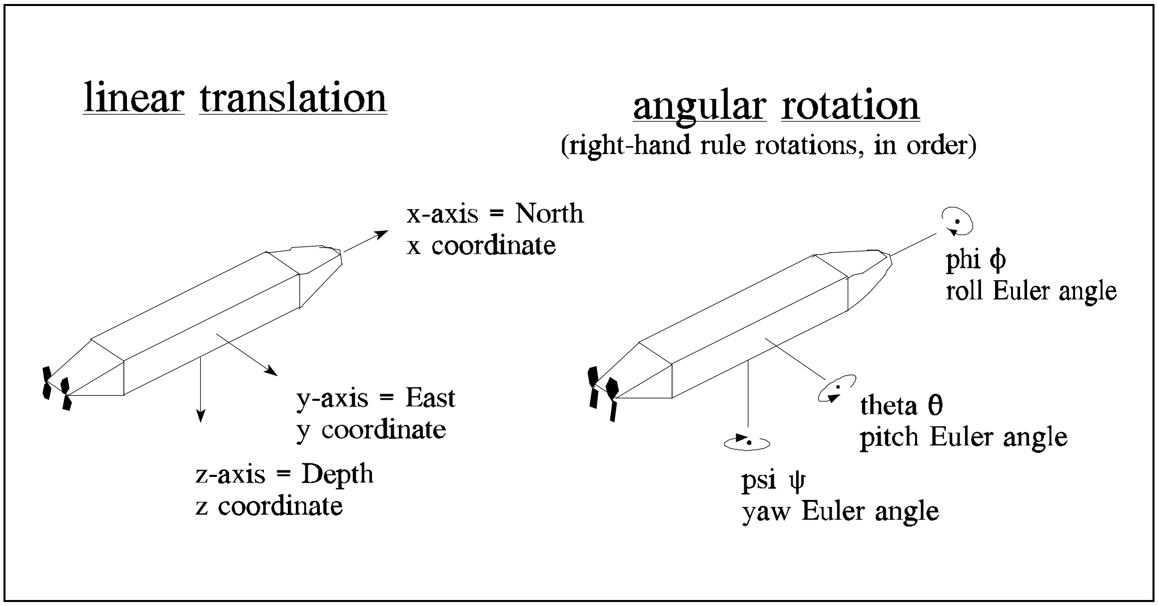

Figure 6.2. World coordinate system: translation and rotation conventions. World x-axis = North, y-axis = East, z-axis = Depth. World-to-body Euler rotations occur in order:
first yaw (ψ), then pitch (θ), then roll (ϕ).

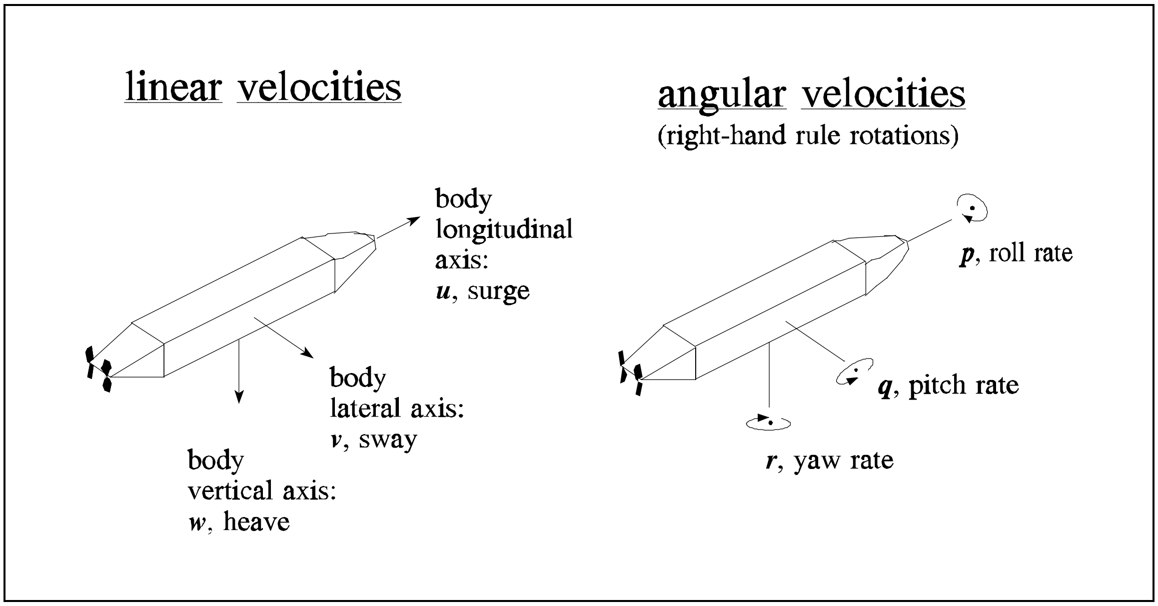

Figure 6.3. Body coordinate system: linear and angular velocity conventions. Note that roll Euler angle rate ≠ roll rate,
pitch Euler angle rate ≠ pitch rate, and
yaw Euler angle rate ≠ yaw rate.

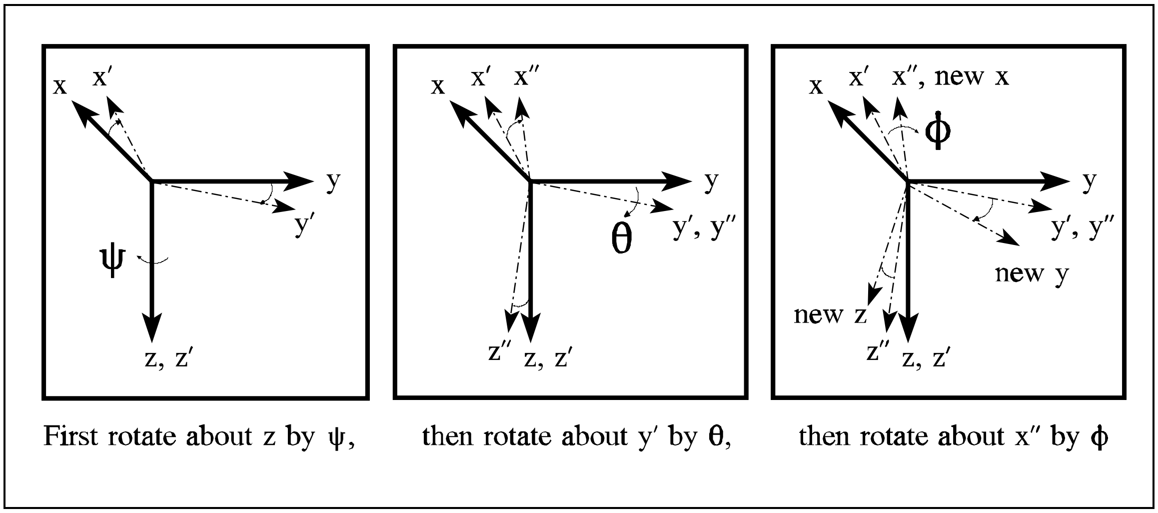

Figure 6.4. Intermediate rotation axes for Euler angle rotations from world coordinate frame to body coordinate frame, adapted from (IEEE 94a).

Normally Euler angles must be restricted from representing a vertical orientation or else mathematical singularities may result. Several techniques for avoiding Euler angle singularities in the vicinity of $\theta = \pm\pi/2$ are discussed in (Cooke 92a, 92b). Permitted ranges of the Euler angles follow:

$$-\pi < \phi \leq \pi \qquad \textbf{(6.1)}$$

$$-\frac{\pi}{2} < \theta < \frac{\pi}{2} \qquad \textbf{(6.2)}$$

$$0 \leq \psi < 2\pi \qquad \textbf{(6.3)}$$

Additionally, most underwater vehicles must be prevented from inverting horizontally or pointing vertically, in order to prevent internal vehicle damage and uncontrollable maneuvering instabilities. These restrictions add a further constraint on roll angle for normal operating conditions, but that constraint will not be applied in this model in order to be able to predict vehicle motion under all conditions.

The order of applying roll, pitch and yaw matrix rotations is fixed since these rotations are not commutative. The Euler angle rotation matrices for converting from body to world coordinates follow in Equation (6.4) (Fu 87) (Cooke 92b). Due to typographic errors in a number of other references, matrix multiplication results are also included in Equation (6.5). Finally it is essential to note that, as will be shown, body coordinate frame rotational velocities **p**, **q** and **r** are quite different from the world coordinate frame Euler angle rotation rates $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$.

$$[R] = [R]_{z,\psi}\ [R]_{y,\theta}\ [R]_{x,\phi} \tag{6.4}$$

$$= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$[R] = \begin{bmatrix} \cos\theta\cdot\cos\psi & \sin\phi\cdot\sin\theta\cdot\cos\psi - \cos\phi\cdot\sin\psi & \cos\phi\cdot\sin\theta\cdot\cos\psi + \sin\phi\cdot\sin\psi \\ \cos\theta\cdot\sin\psi & \sin\phi\cdot\sin\theta\cdot\sin\psi + \cos\phi\cdot\cos\psi & \cos\phi\cdot\sin\theta\cdot\sin\psi - \sin\phi\cdot\cos\psi \\ -\sin\theta & \sin\phi\cdot\cos\theta & \cos\phi\cdot\cos\theta \end{bmatrix} \tag{6.5}$$

Since the body to world rotation matrix ***[R]*** is an orthogonal matrix, it follows that **R** inverse equals ***[R]*** transpose.

$$[R]^{-1} = [R]^{T} \tag{6.6}$$

The three world coordinate frame translation rates can be obtained from the body coordinate frame translation rates by the following matrix equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = [\boldsymbol{R}] \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \textbf{(6.7)}$$

Inversely, body coordinate frame velocities can be determined from world coordinate frame velocities in a similar fashion:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = [\boldsymbol{R}]^T \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad \textbf{(6.8)}$$

The three world coordinate frame Euler angle rotation rates are obtained from body coordinate frame rotation rates by the following non-orthogonal linear transformations (Cooke 92b):

$$\dot{\phi} = p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \quad \textbf{(6.9)}$$

$$\dot{\theta} = q \cos(\phi) - r \sin(\phi) \quad \textbf{(6.10)}$$

$$\dot{\psi} = \frac{q \sin(\phi) + r \cos(\phi)}{\cos(\theta)} \quad \textbf{(6.11)}$$

These three conversions can be combined into matrix notation:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = [T] \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{6.12}$$

where

$$[T] = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{6.13}$$

However note that **[*T*]** is not orthogonal, so **[*T*]$^{-1}$** is not calculated by transposition:

$$[T]^{-1} \neq [T]^{T} \tag{6.14}$$

Instead inverse equations for obtaining body angular velocities from Euler angle rates are as follows:

$$p = \dot{\phi} - \dot{\psi}\sin(\theta) \tag{6.15}$$

$$q = \dot{\theta}\cos(\phi) + \dot{\psi}\sin(\phi)\cos(\theta) \tag{6.16}$$

$$r = -\dot{\theta}\sin(\phi) + \dot{\psi}\cos(\phi)\cos(\theta) \tag{6.17}$$

which yield the following matrix equations:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = [T]^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad \textbf{(6.18)}$$

$$[T]^{-1} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \qquad \textbf{(6.19)}$$

The preceding equations provide a complete set of component conversions between the world coordinate frame and body coordinate frame linear and angular velocities. All component velocities can be further grouped together in matrix notation. Combined velocity matrix definitions are as follows:

$$[V]_{body} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \qquad \textbf{(6.20)}$$

$$[V]_{world} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad \textbf{(6.21)}$$

Matrix conversion from body to world velocities is thus:

$$[V]_{world} = \left[\begin{array}{c|c} [R] & 0 \\ \hline 0 & [T] \end{array} \right] [V]_{body} \tag{6.22}$$

and inversely:

$$[V]_{body} = \left[\begin{array}{c|c} [R]^T & 0 \\ \hline 0 & [T]^{-1} \end{array} \right] [V]_{world} \tag{6.23}$$

These velocity relationships are the *kinematics equations of motion* (Greenwood 88). Equations (6.22) and (6.23) are equivalent ways of expressing Euler angle constraints between the inertial world coordinate frame and the rotating body coordinate frame. Each has six component equations linking twelve velocity components. When combined with the dynamics equations of motion, the kinematics equations of motion provide constraints essential to solving world coordinate system values of the vehicle state vector.

D. GENERAL REAL-TIME HYDRODYNAMICS MODEL FOR AN UNDERWATER VEHICLE

1. Definitions

A virtual world simulation component for hydrodynamics modeling of a submerged rigid body must account for six spatial degrees of freedom in real time. The six spatial degrees of freedom include *position* (3 position coordinates ***x***, ***y***, ***z***) and *orientation* (3 rotational Euler angles ϕ, θ, ψ). Together these six components describe vehicle *posture* ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, ϕ, θ, ψ). Accelerations and velocities each have these same six spatial degrees of freedom.

Six values for velocity and six values for posture comprise the vehicle *state vector*, since together they can fully specify in vehicle posture over time without redundancy. This state vector is in the world coordinate system. The overall goal of the hydrodynamics model is to calculate updated values of the vehicle state vector at each time step.

Much more information is needed to describe robot state. The hydrodynamics model needs a reasonably complete snapshot of robot state in order to properly predict interactions between robot and environment. All hydrodynamics state variables must be included, as well as a variety of sensor values, pertinent robot logic states, and variables for accelerations (due to forces and moments) as produced by effector values for propellers, rudders, planes and thrusters. The dynamics model is provided this partial snapshot of current robot state with each exchange of the robot telemetry record. After examining parameters controlled by the robot (e.g. robot orders for propellers and fins), the hydrodynamics model then calculates an updated state vector. With the updated state vector the hydrodynamics model is then able to calculate values expected from various robot sensors which ordinarily query the environment. By updating missing sensor values in the robot telemetry record with newly calculated sensor values, the hydrodynamics model provides virtual sensor response in the laboratory. Vehicle operation in virtual world or real world remains transparent to the robot. Further details on this data communication mechanism are included in Chapter IV.

2. Real Time

"Real time" in this context is defined by the requirement that a vehicle maneuvering within the virtual world describe essentially the same path and postures as the vehicle maneuvering in the real world. This requires that the robot hardware and software receives the same responsiveness from the virtual world as from the real world, since robot behavior is very closely coupled to real-time interactions and deadlines (Payton 91) (Badr 92).

> A *real-time system* is a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure... A *failed system* is a system which cannot satisfy one or more of the requirements laid out in the formal system specification. (Laplante 93)

Real-time systems can be further characterized by the criticality of their timing requirements, which are classified as hard or soft. *Hard real-time* system correctness is strictly dependent on the timeliness of results. *Soft real-time* systems may experience reduced effectiveness but will not fail due to missed deadlines. Alternatively, hard real-time systems are those which include the possibility of system loss or potential catastrophe if deadlines are not met, and soft real-time systems are those where "sooner is better than later" but lateness will not cause system failure. As a point of interest, *firm real-time* systems have been defined as those with hard deadlines that can survive despite the presence of low probabilities for missing a deadline. (Laplante 93) (Halang 91)

By these definitions it is clear that the system consisting of a robot interacting with a dynamics-based virtual world is a hard real-time system. Furthermore the robot itself operating in a real world environment is also a hard real-time system, since a temporal failure in navigation or depth control might result in vehicle destruction. However, in isolation, the dynamics component of the virtual world are able to provide accurate results regardless of the temporal scaling of interaction requests. Therefore the dynamics model *per se* can be classified as a soft real-time system. It only needs to be fast enough to support the hard real-time requirements of the networked robot processors. In general, hydrodynamics model responsiveness will be a function of algorithmic complexity, implementation efficiency, microprocessor performance and communications latency.

3. Forces, Moments and Accelerations

Forces and accelerations for the six state variables of posture can be grouped together in the matrix form of Newton's Second Law, initially expressed as

$$[F]_{world} = \frac{d}{dt}[M\,V]_{world} \qquad \textbf{(6.24)}$$

For a rigid body, translational forces are normally applied at the ***CG***. Moments are free vectors producing rotations to be applied about the origin of the vehicle body, since inertial integrals are calculated relative to that origin. Usual practice is to define ***CG*** measurements as being offset from ***CB***. Vehicle origin is not assumed coincident.

The key to properly estimating world coordinate frame velocities and position will be properly calculating time rate of change of velocities in the body coordinate frame, represented as:

$$[\dot{V}]_{body} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \qquad \textbf{(6.25)}$$

Time rate of change of body velocities $[\dot{V}]_{body}$ can also be referred to as *body-relative accelerations* $[A]_{body}$. However it must be clearly understood that these body accelerations are only with respect to the body coordinate frame, i.e. those which appear to a local observer moving with the body reference frame (Greenwood 88). Absolute acceleration components due to rotation and velocity changes between the body reference frame and world reference frame are specifically excluded from $[A]_{body}$.

Since physical interactions occur between the vehicle and the immediately surrounding water volume, force and moment calculations are most directly evaluated in the body-fixed coordinate frame. Moment of inertia terms in the mass matrix **[*M*]** can only be constant in a body coordinate frame, further making the body frame attractive for dynamics calculations. Mathematical rederivation of known acceleration relationships in a world coordinate reference frame is possible using a Lagrangian representation (Fossen 94). However such a form appears to be much less direct than the Newton-Euler formulations, particularly since the virtual world is centered around a robot vehicle which operates and interacts relative to the local body-fixed coordinate frame. Therefore it is desirable that all linear and angular acceleration and velocity relationships be specified exactly and completely in the body coordinate frame. Doing so yields six dynamics equations of motion relating the twelve unknowns of the vehicle state vector derivatives: six unknowns are body velocities, and six unknowns are body accelerations.

Although the vector sum of velocity components expressed in body coordinates equals the vector sum of velocity components expressed in world coordinates, an equivalent relationship does not hold for body and world acceleration vectors because the body coordinate frame is rotating. Specifically, differentiating Equation (6.22) with respect to time yields:

$$\frac{d}{dt}\,[V]_{world} = \frac{d}{dt}\left[\begin{array}{c|c} [R] & 0 \\ \hline 0 & [T] \end{array}\right][V]_{body} + \left[\begin{array}{c|c} [R] & 0 \\ \hline 0 & [T] \end{array}\right]\frac{d}{dt}\,[V]_{body} \qquad \textbf{(6.26)}$$

Substituting acceleration [$\boldsymbol{A}$] for time rate of change of [$\boldsymbol{V}$] (as always in the appropriate coordinate frames) results in

$$[A]_{world} = \frac{d}{dt}\left[\begin{array}{c|c} [R] & 0 \\ \hline 0 & [T] \end{array}\right][V]_{body} + \left[\begin{array}{c|c} [R] & 0 \\ \hline 0 & [T] \end{array}\right][A]_{body} \qquad \textbf{(6.27)}$$

Inspection of Equation (6.27) makes it clear that world coordinate frame accelerations $[\boldsymbol{A}]_{world}$ and rotated body coordinate frame accelerations $[\boldsymbol{A}]_{body}$ are not equivalent unless the transformation matrix between coordinate frames is unchanging, or all body velocities are zero (Greenwood 88). It is possible to examine accelerations acting upon the body from a perspective within the rotating body coordinate frame, but they cannot be directly integrated into world coordinate frame accelerations.

It is possible to numerically integrate the six dynamics equations with respect to time and determine new velocity values. This dynamics equation integration must be performed using body coordinate frame variables. Once new values for body velocities are thereby obtained, the six Euler kinematic constraint equations of motion (6.22) are utilized to produce linear and angular world velocities. Finally posture is determined within the world coordinate system using world velocities.

4. Time Dependencies

During operation of a vehicle in a virtual world, forces acting on the robot can be estimated from the vehicle state vector while velocities and body accelerations are analytically derived. During operation in the real world, forces can be similarly estimated while accurate velocity and body acceleration information may (or may not) be available from flow and inertial navigation sensors. In either world, good estimates of changes in body frame velocities are a primary robot requirement so that velocity and posture estimates can be cumulatively integrated over time. Accurately estimating body frame velocity changes at suitably short time intervals is the key to properly modeling vehicle hydrodynamics response. Thus the dynamics equations of motion must be written to produce time rates of change of velocities as the dependent variables, obtained through calculations that solely involve vehicle variables (such as posture, propellers, thrusters and plane surfaces) which are continuously known to the robot.

The Newton-Euler formulation of Newton's Second Law from Equation (6.24) can be expanded by the chain rule to produce

$$[F]_{body} = \frac{d}{dt}\left([M]_{body} \cdot [V]_{body}\right) = \frac{d[M]}{dt}[V] + [M]\frac{d[V]}{dt} \qquad \textbf{(6.28)}$$

Within the body coordinate frame the mass matrix **[*M*]** is unchanging. Differentiation of the velocity matrix **[*V*]** reveals effects that are due to the body coordinate frame rotating with angular velocity ω with respect to the world coordinate frame (Fossen 94):

$$[F]_{body} = [M]\left([\dot{V}]_{body} + \omega \times [V]_{body}\right) \qquad \textbf{(6.29)}$$

Since matrix multiplication is associative but not commutative, both sides of matrix Equation (6.29) can be multiplied by a single matrix as long as order of multiplication is carefully preserved. In this case both sides of Equation (6.29) are multiplied by the mass matrix inverse $[\boldsymbol{M}]^{-1}$. Transposing the result yields Equation (6.30):

$$[\dot{V}]_{body} = [M]^{-1} \cdot [F]_{body} - \omega \times [V]_{body} \qquad \textbf{(6.30)}$$

This form of the dynamics equation is very important from a time-integration perspective, since all accelerations are grouped together on the left-hand side. All terms on the right-hand sides of the dynamics equations are known or can be determined during vehicle operation in the virtual world. Thus calculation of the right-hand side can be used to determine updated values of the left-hand side.

Minimizing errors during the integration time step is essential for accurate real-time simulation of hydrodynamics models. This is due to the sensitivity of the hydrodynamics model to small perturbations, as well as the high degree of cross-coupling between forces acting on the three physical body axes of an submerged vehicle. If errors in determining body accelerations are minimized, then integration of body accelerations and subsequent coordinate frame transformations to yield velocities, positions and orientations will also minimize any accumulated errors inherent in velocity and posture estimation.

The local forces acting on an autonomous underwater vehicle are due to onboard effectors such as propellers, thrusters, planes and rudders. External ocean current forces are assumed to vary slowly with respect to vehicle time and act on the entire vehicle uniformly, having no effect on the interactions between the vehicle and the immediately adjacent water volume. Ocean current effects can therefore be added as a simple uniform translation. This vector addition is performed after fully calculating the effects of body accelerations and velocities, and after shifting back from a body coordinate system to the world coordinate system. Thus all forces can be completely determined or estimated in real time during vehicle operation. For constant-ballast vehicles, all elements of the body frame mass matrix $[\boldsymbol{M}]$ and corresponding inverse $[\boldsymbol{M}]^{-1}$ can be determined empirically through prior testing (to a close first approximation) and are not time-varying.

5. Velocities and Postures

Combination of force and mass matrices as described above gives a very accurate estimation of time rates of change of body velocities. The body velocity rate matrix is integrated first to provide linear and rotational velocities, then integrated again to provide posture. Initial integration to yield body velocities occurs in the body fixed coordinate frame:

$$[V]_{body\ (t0+\delta t)} = \int_{t0}^{t0+\delta t} [\dot{V}]_{body\ (t)}\, dt + [V]_{body\ (t0)} \tag{6.31}$$

Integration of the new body velocities to determine posture is preceded by a transformation from the body-fixed coordinate frame to the world coordinate frame. The following substitution pertains:

$$\int_{t0}^{t0+\delta t} [V]_{world\ (t)}\, dt = \int_{t0}^{t0+\delta t} \left([V]_{body\ (t)} \right)_{\substack{body \rightarrow world \\ coordinate\ shift}} dt \tag{6.32}$$

The final integration to determine posture is therefore:

$$\begin{aligned} [Posture]_{world\ (t0+\delta t)} = & \int_{t0}^{t0+\delta t} [V]_{world\ (t)}\, dt \\ & + \int_{t0}^{t0+\delta t} [Ocean\ currents]_{world\ (t)}\, dt \\ & + [Posture]_{world\ (t0)} \end{aligned} \tag{6.33}$$

6. Deriving Desired Form of Dynamics Equations of Motion

A full set of hydrodynamic equations of motion for a submerged vehicle are not usually written in the form suggested by Equations (6.30), (6.31), (6.32) and (6.33). Other derivations have been presented in the open literature (Gertler 76) (Smith 79) (Feldman 79) (Papoulias 89) (Watkinson 89) (Yuh 90) (Humphreys 91) (Baiardi 92) (Healey 93) (Fossen 94) and a variety of other sources, but are structured in such a way that similar time-dependent acceleration-related terms are present on both sides of the dynamics equations of motion. Because related body acceleration terms are not grouped together, direct time integration of both sides of the equations of motion is not mathematically valid in those representations. Furthermore these many references are all handicapped by variations in nomenclature and even a surprising variety of typographical errors, mathematical errors or omissions. None of these other models can be directly applied as a valid real-time underwater virtual world component. *Therefore the primary intended contributions of the hydrodynamic model developed here are clarity, correctness, generality, standardized nomenclature and suitability for real-time simulation.*

Given this broad outline showing how the dynamics equations of motion will be utilized, it is time to derive the desired forms of the hydrodynamics equations. We can reorganize all of the original (Healey 93) equations of motion to solely have mass-related, inertia-related and $[\dot{V}]_{body}$-related terms on the left-hand side. That rearrangement leaves lift and drag, buoyancy, weight, propulsion thrust, and other forces and moments on the right-hand side. At any given instant $\boldsymbol{t_0}$:

$$\begin{bmatrix} \boldsymbol{M} \\ \textit{mass, inertia} \\ \textit{and} \\ \textit{added-mass} \\ \textit{matrix} \\ \; \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{(t_0)} = \begin{bmatrix} \textit{velocity-related} \\ \textit{inertial and} \\ \textit{hydrodynamic} \\ (\textit{drag and lift}) \\ \textit{force functions} \\ \; \end{bmatrix}_{\left([V]_{body\ (t_0)}\right)} + \begin{bmatrix} \textit{buoyancy,} \\ \textit{weight,} \\ \textit{propulsion} \\ \textit{and other} \\ \textit{forces} \\ \; \end{bmatrix}_{(t_0)} \quad \textbf{(6.34)}$$

Calculating the inverse mass matrix and multiplying it against both sides of Equation (6.34) leaves only body accelerations on the left-hand side. Further classification of individual terms on the right-hand side as corresponding to Coriolis, centripetal and other forces can be found in (Greenwood 88) (Healey 92c). For the purpose of this derivation it is sufficient to group these accelerations together without further discussion. Such an arrangement prepares the dynamics equations of motion for temporal integration in the body-fixed coordinate frame as follows:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{(t_0)} = \begin{bmatrix} M^{-1} \\ mass\ matrix \\ inverse \\ \\ \end{bmatrix} \begin{bmatrix} dynamics \\ equations \\ of\ motion \\ right-hand \\ sides \\ \end{bmatrix}_{(t_0)} \qquad \textbf{(6.35)}$$

The body frame velocity matrix $\mathbf{[}\boldsymbol{V}\mathbf{]}_{\boldsymbol{body}}$ can now be updated by numerical integration. For example, Euler integration (Hamming 86) (Green 91) (Press 92) yields:

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}_{(t_0 + \delta t)} = \delta t \cdot \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{(t_0)} + \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}_{(t_0)} \qquad \textbf{(6.36)}$$

A slightly more precise estimation of the velocity matrix can be achieved by averaging body acceleration at the beginning and end of each time step prior to integrating with respect to time. This method called second-order Runge-Kutta or *Heun* integration (Fossen 94), and is also the approach used for velocity estimation in the source code implementing this work (Brutzman 94e).

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}_{(t_0 + \delta t)} = \delta t \cdot \frac{1}{2} \cdot \left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{(t_0 + \delta t)} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{(t_0)} \right) + \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}_{(t_0)} \qquad \textbf{(6.37)}$$

where $[\dot{u}\ \dot{v}\ \dot{w}\ \dot{p}\ \dot{q}\ \dot{r}]^T_{(t+\delta t)}$ in Equation (6.37) is itself an estimate obtained by Euler integration of $[\dot{u}\ \dot{v}\ \dot{w}\ \dot{p}\ \dot{q}\ \dot{r}]^T_{(t)}$.

Conversion of these body velocities to world velocities is performed using the transformation of Equation (6.22). Subsequent integration of world velocities into world posture is performed by Euler integration as follows:

$$\begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}_{(t_0 + \delta t)} = \delta t \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{(t_0)} + \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix}_{(t_0)} \qquad \textbf{(6.38)}$$

Final addition of ocean current effects completes the calculation of world coordinate system posture, as previously specified in Equation (6.33).

Increasingly accurate temporal resolution is possible using smaller time steps, chosen adaptively if necessary. Further numerical analysis considerations and recommendations appear in (Press 92) (Green 91) and (Hamming 86). In practice, a fixed time step of 0.1 seconds has worked well for model resolution, real-time robot hardware control response, network latency, remote interaction, computer graphics rendering update rate, and human observation. Care must be taken if higher-order integration methods are employed to ensure that hydrodynamics model responsiveness does not degrade past the real-time requirements of the robot operating in the virtual world.

To summarize: the dynamics equations of motion are not mathematically rewritten in world coordinates, but are kept in body coordinates. Integrating the dynamics equations of motion provides body velocity values at the next time step. These new body coordinate frame velocities are combined with the kinematics equations of motion to produce world coordinate frame velocities. World velocities are then integrated and added to ocean current effects to produce updated world postures. Algorithmic complexity is sufficiently low to permit rapid model response within the same time period that the robot normally uses to query vehicle sensors.

7. Nomenclature Tables for Variables and Coefficients

The many details pertaining this approach still need to be filled in using a complete six-degree-of-freedom set of dynamics equations of motion. First, however, it must be noted that small yet persistent nomenclature inconsistencies were encountered in all of the dozens of hydrodynamics references studied. This is a serious problem for newcomers to the hydrodynamics literature, since both names and definitions of key terms may vary. This lack of standardization results in troubling mathematical incompatibilities throughout an entire body of scientific literature. Clearly an important prerequisite for describing any general hydrodynamics model is to use well-defined (and hopefully standardized) nomenclature. Of all the hydrodynamics models studied in this work, (Healey 93) and (Fossen 94) appear to be the most general and most applicable for real-time simulation of autonomous underwater vehicle response. The nomenclature of (Healey 93) closely follows that of the standard reference work on ship control (Lewis 88). The same nomenclature is followed here.

Since usage of a rigorously standardized nomenclature is only partially possible, this work will attempt to follow accepted conventions wherever possible while precisely defining all variables and coefficients, both mathematically and descriptively. Coefficient subscripts from previous models have been corrected when necessary to explicitly indicate accompanying variable factors. Such an approach permits comparison of this hydrodynamics model with any other work, and hopefully provides clarity in a subject area that unfortunately includes wide variation.

The following tables define and describe the state variables and hydrodynamics coefficients used. Symbol, name, description, units and coefficient value (for the NPS AUV) are included. Variable and coefficient names in implementation software source code (Brutzman 94e) match exactly to further encourage clarity and correctness.

Close examination of the dynamics equations of motion reveals that nearly all of the hydrodynamics coefficients are dimensionless, having been normalized with respect to vehicle length L. This convention permits rough comparison of the relative effects of individual coefficients with other vehicles or between different body axis orientations.

Coefficients presented here have been tested for a large variety of scenarios (Brutzman 94e). Nevertheless the complexities of hydrodynamics testing and intricacies of this model preclude complete validation, verification and accreditation. Constant coefficients are included for dynamics effects that occur in both cruise mode and hover mode. For vehicles that are capable of much higher speeds, coefficients are expected to become variable as a function of Reynolds Number, which quantifies the transition from laminar to fully developed turbulent fluid flow. Examination of Reynolds number effects on hydrodynamics coefficients appears in (Humphreys 89) (Ruth, Humphreys 90) (Humphreys 91). Further testing and refinement of hydrodynamics coefficients for various vehicles is an important subject for future work.

Although lengthy, proper definition of the numerous state variables and hydrodynamics coefficients is essential when producing and understanding a hydrodynamic model capable of providing precise response within an underwater virtual world. The complete tables are presented here as an integral and essential part of the hydrodynamics model, in order to provide context for the derivations of the equations of motion which follow. A similarly exhaustive set of definitions for submarine simulation is included in (Feldman 79). NPS AUV II coefficient values are from (Warner 91) (Bahrke 92) (Marco 95) and laboratory testing. All angular definitions conform to the right-hand rule. Readers interested in comprehending the final six dynamics equations of motion are urged to closely examine the precise variable and coefficient definitions provided in these nomenclature tables.

Table 6.1. Hydrodynamics and Control System Variables.

Symbol	Name	Description	Coordinate system	Units
t	time	Clock time (real or simulated)	-	seconds
δ***t***	time step	Loop interval (robot or dynamics model)	-	seconds
x	x	Position along North-South axis, North positive.	world	feet
y	y	Position along East-West axis, East positive.	world	feet
z	z	Depth, downward direction is positive.	world	feet
ϕ	roll Euler angle	Roll Euler angle rotation about North-South axis, preceding pitch and yaw rotations. Positive sense clockwise as seen from stern to bow of vehicle.	world	radians
θ	pitch Euler angle	Pitch Euler angle rotation about East-West axis, following rotation for roll and preceding rotation for yaw. Positive sense is clockwise as seen from port side of vehicle.	world	radians
ψ	yaw Euler angle	Yaw Euler angle rotation about vertical (depth) axis, following rotations for roll and pitch. Positive sense is clockwise as seen from above.	world	radians

Symbol	Name	Description	Coordinate system	Units
$\dot{x}$	x dot	Linear velocity along North-South axis.	world	ft/sec
$\dot{y}$	y dot	Linear velocity along East-West axis.	world	ft/sec
$\dot{z}$	z dot	Linear velocity along Depth axis.	world	ft/sec
$\dot{\phi}$	phi dot	Roll Euler angle rate component, about North-South axis. Not equivalent to ***p***.	world	radians/sec
$\dot{\theta}$	theta dot	Pitch Euler angle rate component, about East-West axis. Not equivalent to ***q***.	world	radians/sec
$\dot{\psi}$	psi dot	Yaw Euler angle rate component, about vertical (depth) axis. Not equivalent to ***r***.	world	radians/sec
u	surge	Linear velocity along longitudinal axis.	body	ft/sec
v	sway (sideslip)	Linear velocity along lateral axis.	body	ft/sec
w	heave	Linear velocity along vertical axis.	body	ft/sec
p	roll rate	Angular velocity component about longitudinal axis. Not equivalent to $\dot{\phi}$.	body	radians/sec
q	pitch rate	Angular velocity component about lateral axis. Not equivalent to $\dot{\theta}$.	body	radians/sec
r	yaw rate	Angular velocity component about vertical axis. Not equivalent to $\dot{\psi}$.	body	radians/sec

Symbol	Name	Description	Coordinate system	Units
$\dot{u}$	u dot	Time rate of change of surge velocity (along longitudinal axis)	body	ft/sec^2
$\dot{v}$	v dot	Time rate of change of sway velocity (along lateral axis)	body	ft/sec^2
$\dot{w}$	w dot	Time rate of change of heave velocity (along vertical axis)	body	ft/sec^2
$\dot{p}$	p dot	Time rate of change of roll angular velocity (about longitudinal axis)	body	radians/sec^2
$\dot{q}$	q dot	Time rate of change of pitch angular velocity (about lateral axis)	body	radians/sec^2
$\dot{r}$	r dot	Time rate of change of yaw angular velocity (about vertical axis)	body	radians/sec^2
δ_{rb}	bow rudders angle	Bow rudder deflection angle. Usually bow and stern rudders orders go to exactly opposite positions. Positive sense is clockwise as seen from above. Positive bow rudders angle with positive surge ***u*** produces positive change in yaw.	body	radians

Symbol	Name	Description	Coordinate system	Units
δ_{rs}	stern rudders angle	Stern rudder deflection angle. Usually bow and stern rudders go to exactly opposite positions). Positive sense is clockwise as seen from above. Positive stern rudders angle with positive surge ***u*** produces negative change in yaw.	body	radians
δ_{pb}	bow planes angle	Bow planes deflection angle. Usually bow and stern planes go to exactly opposite positions). Positive sense is clockwise as seen from the port side of the vehicle. Positive bow planes angle with positive surge ***u*** produces positive change in pitch.	body	radians
δ_{ps}	stern planes angle	Stern planes deflection angle. Usually bow and stern planeS go to exactly opposite positions). Positive sense is clockwIse as seen from the port side of the vehicle. Positive stern planes angle with positive surge ***u*** produces negative change in pitch.	body	radians
$\boldsymbol{n_{porT}}$	Port rpm	Port propeller ordered turns	body	rpm
$\boldsymbol{n_{stbd}}$	Stbd rpm	Starboard propeller ordered turns	body	rpm

Symbol	Name	Description	Coordinate system	Units
$V_{bow\text{-}vertical}$		Volts, bow vertical cross-body thruster (±24 V corresponds to ±2.0 lb)	body	volts
$V_{stern\text{-}vertical}$		Volts, stern vertical cross-body thruster (±24 V corresponds to ±2.0 lb)	body	volts
$V_{bow\text{-}lateral}$		Volts, bow lateral cross-body thruster (±24 V corresponds to ±2.0 lb)	body	volts
$V_{stern\text{-}lateral}$		Volts, stern lateral cross-body thruster (±24 V corresponds to ±2.0 lb)	body	volts

Table 6.2. Hydrodynamics Model Coefficients.

Coefficient	**Name**	**Description**	**Value for NPS AUV II**
W	weight	Submerged weight of vehicle, including water in contained free-flood spaces, neutral ballast.	435 lb
B	buoyancy	Weight of water displaced by vehicle, including water in contained free-flood spaces. Can vary with depth (due to hull compression) and with changes in water density ρ.	435 lb
L	length	Vehicle length, also known as characteristic length. Dynamics equations of motion are written to explicitly utilize ***L*** as a normalization coefficient. This approach makes most other coefficients dimensionless and quantitatively independent of vehicle dimensions, permitting comparison of relative effects between different forces and dissimilar vehicles.	7.302 ft
g		Acceleration due to gravity	32.174 ft/sec^2
ρ	rho	Mass density of fresh water: Mass density of sea water (representative):	1.94 $slugs/ft^3$ 1.99 $slugs/ft^3$
m	mass	Vehicle mass, including water contained in enclosed free-flood spaces, neutral ballast.	***W / g*** = 13.52 $(lb \cdot sec^2)/ft$
I_x		Mass moment of inertia coefficient about body longitudinal axis, Equation (6.55)	2.7 $ft \cdot lb \cdot sec^2$

Coefficient	Name	Description	Value for NPS AUV II
I_y		Mass moment of inertia coefficient about body lateral axis, Equation (6.56)	42.0 ft·lb·sec^2
I_z		Mass moment of inertia coefficient about body vertical axis, Equation (6.57)	45.0 ft·lb·sec^2
I_{xy} $= I_{yx}$		Cross product of inertia coefficient, due to asymmetric mass distribution about body longitudinal/lateral axes, Equation (6.58)	0.0 ft·lb· sec^2
I_{xz} $= I_{zx}$		Cross product of inertia coefficient, due to asymmetric mass distribution about body longitudinal/vertical axes, Equation (6.59)	0.0 ft·lb· sec^2
I_{yz} $= I_{zy}$		Cross product of inertia coefficient, due to asymmetric mass distribution about body lateral/vertical axes, Equation (6.60)	0.0 ft·lb· sec^2
CG	center of gravity	Mass centroid of vehicle. The CG is the apparent point where forces and moments are applied.	(x_G, y_G, z_G)
x_G		Center of gravity location along body longitudinal axis, measured in body coordinates from nominal vehicle centroid	0.125 in = 0.010 ft
y_G		Center of gravity location along body lateral axis, measured in body coordinates from nominal vehicle centroid	0.0 ft

Coefficient	Name	Description	Value for NPS AUV II
z_G		Center of gravity location along body vertical axis, measured in body coordinates from nominal vehicle centroid. Note z_G is below center of buoyancy z_B by design for passive roll/pitch stability.	1.07 in = 0.089 ft
CB	center of buoyancy	Volumetric centroid of the vehicle.	(x_B, y_B, z_B)
x_B		Center of buoyancy location along body longitudinal axis	0.125 in = 0.010 ft
y_B		Center of buoyancy location along body lateral axis	0.0 ft
z_B		Center of buoyancy location along body vertical axis. Note z_B is above center of gravity z_G by design for passive roll/pitch stability.	0.0 ft
$x_{bow\text{-}vertical}$		Distance from nominal vehicle centroid to centerline of bow vertical thruster tunnel along body longitudinal axis.	1.41 ft
$x_{stern\text{-}vertical}$		Distance from nominal vehicle centroid to centerline of stern vertical thruster tunnel along body longitudinal axis. Note negative.	- 1.41 ft
$x_{bow\text{-}lateral}$		Distance from nominal vehicle centroid to centerline of bow lateral thruster tunnel along body longitudinal axis.	1.92 ft

Coefficient	Name	Description	Value for NPS AUV II
$x_{stern\text{-}lateral}$		Distance from nominal vehicle centroid to centerline of stern lateral thruster tunnel along body longitudinal axis. Note negative.	- 1.92 ft
$y_{port\text{-}propeller}$		Port propeller shaft offset from longitudinal centerline of vehicle	- 3.75 in = - 0.313 ft
$y_{stbd\text{-}propeller}$		Starboard propeller shaft offset from longitudinal centerline of vehicle	3.75 in = 0.313 ft
$h\ (x)$		Width of vehicle at body center along the y-axis, at a given position x measured on the longitudinal body axis	vehicle geometry tabular data
$b\ (x)$		Height of vehicle at body center along the z-axis, at a given position x measured on the longitudinal body axis	vehicle geometry tabular data
$U_{cf}\ (x)$		Total cross-flow velocity across body at a given body position x along longitudinal axis	see Equation (6.47)

Coefficient	Name	Description	Value for NPS AUV II
Surge force coefficients			
steady-state speed per maximum propeller rpm		Average forward velocity based on combined propeller revolutions per minute (rpm), typically measured at maximum steady-state speed. Analogous to turns-per-knot (TPK) ratio for ships with fixed-pitch propellers.	$\left(\frac{2 \text{ ft/sec}}{700 \text{ rpm}}\right)$ for twin propellers, steady state
X_{prop}		No longer used, since X_{prop} term is not a true coefficient. X_{prop} is now decomposed in Equation (6.43) to explicitly show individual contributing propulsion-related variables, which are then included in the revised surge equation of motion (6.48).	Not used. Previous values are no longer applicable.

Coefficient	Name	Description	Value for NPS AUV II
$X_{\dot{u}}$		Coefficients describing surge forces from resolved lift, drag and fluid inertia along body longitudinal axis. These occur in response to individual (or multiple) velocity, acceleration and plane surface components, as indicated by the corresponding subscripts. For example: $X_{\dot{u}}$ describes the drag contribution in the longitudinal $\boldsymbol{X}$ direction due to time rate of change of surge velocity ($\dot{\boldsymbol{u}}$) Note that any coefficient may be non-zero, depending principally on the geometry of the vehicle being modeled.	-2.82 E-3
$X_{\dot{v}}$			0.0
$X_{\dot{w}}$			0.0
$X_{\dot{p}}$			0.0
$X_{\dot{q}}$			0.0
$X_{\dot{r}}$			0.0
X_{uu}			0.0
X_{vv}			0.0
X_{ww}			0.0
X_{pp}			0.0
X_{qq}			0.0
X_{rr}			0.0
$X_{uq\delta b}$			0.0
$X_{uq\delta s}$			0.0
$X_{ur\delta r}$			0.0
$X_{uv\delta r}$			0.0
$X_{uw\delta b}$			0.0
$X_{uw\delta s}$			0.0

Coefficient	Name	Description	Value for NPS AUV II
$X_{u\|u\|\delta b \delta b}$		Drag force due to square of deflection angle of bow planes (δ_{pb}), stern planes (δ_{ps}) and rudders (δ_{rb}, δ_{rs}) respectively due to square of surge ***u***	-1.018 E-2
$X_{u\|u\|\delta s \delta s}$			-1.018 E-2
$X_{u\|u\|\delta r \delta r}$			-1.018 E-2
X_{pr}		Fluid inertia force due to paired interactions as indicated by subscripted velocities, typically nonzero only as a result of asymmetries in the vehicle hull form	0.0
X_{wq}			0.0
X_{vp}			0.0
X_{vr}			0.0
C_{d0}		Drag coefficient along body longitudinal axis	0.00778
Note: for remaining coefficients, only non-negligible NPS AUV values are listed.			
Sway force coefficients			
$Y_{\dot{v}}$		Coefficients describing sway forces from resolved lift, drag and fluid inertia along body lateral axis. These occur in response to individual (or multiple) velocity, acceleration and plane surface components, as indicated by the corresponding subscripts.	-3.43 E-2
$Y_{\dot{r}}$			-1.78 E-1
Y_{uv}			-1.07 E-1
Y_{ur}			0.0
$Y_{u\|u\|\delta rb}$			+1.18 E-2
$Y_{u\|u\|\delta rs}$			+1.18 E-2
C_{dy}		Drag coefficient along body lateral axis	0.5

Coefficient	Name	Description	Value for NPS AUV II
Heave force coefficients			
$Z_{\dot{w}}$		Coefficients describing heave forces from resolved lift, drag and fluid inertia along body vertical axis. These occur in response to individual (or multiple) velocity, acceleration and plane surface components, as indicated by the corresponding subscripts.	-9.43 E-2
$Z_{\dot{q}}$			-2.53 E-3
Z_{uw}			-7.844 E-1
Z_{uq}			-7.013 E-2
$Z_{u\|u\|\delta pb}$			-2.11 E-2
$Z_{u\|u\|\delta ps}$			-2.11 E-2
C_{dy}		Drag coefficient along body vertical axis	0.6
Roll moment coefficients			
$K_{\dot{p}}$		Fluid inertia moment about longitudinal body axis due to time rate of change of roll rate ($\dot{p}$)	-2.4 E-4
$K_{\|u\|p}$		Fluid inertia moment about longitudinal body axis due to existing roll ***p*** and magnitude of surge ***u***	-5.4 E-3
$K_{p\|p\|}$		Drag moment about longitudinal body axis due to signed square of existing roll ***p*** corresponding to turbulent flow	-2.02 E-2 estimate

Coefficient	Name	Description	Value for NPS AUV II
K_p		Drag moment about longitudinal body axis due to existing roll ***p*** corresponding to laminar flow, approximately equals $K_{p\|p\|}$ at 1°/sec	$K_{p\|p\|}\left(\frac{\pi}{180^\circ}\right)$ estimate
Pitch moment coefficients			
$M_{\dot{w}}$		Fluid inertia moment about lateral body axis due to time rate of change of heave rate ($\dot{w}$)	-2.53 E-3
$M_{\dot{q}}$		Fluid inertia moment about lateral body axis due to time rate of change of pitch rate ($\dot{q}$)	-6.25 E-3
M_{uw}		Fluid inertia moment about lateral body axis due to existing heave ***w*** and surge ***u***	0.0
M_{uq}		Fluid inertia moment about lateral body axis due to existing pitch ***q*** and surge ***u***	-1.53 E-2
$M_{u\|u\|\delta pb}$		Drag moment force about lateral body axis due to bow plane deflection δ_{pb} and signed square of surge ***u***	-0.283·L· $Z_{u\|u\|\delta pb}$
$M_{u\|u\|\delta ps}$		Drag moment about lateral body axis due to stern plane deflection δ_{ps} and signed square of surge ***u***	+0.377·L· $Z_{u\|u\|\delta ps}$
$M_{q\|q\|}$		Drag moment about lateral body axis due to signed square of existing pitch ***q*** corresponding to turbulent flow	-7.0 E-3 estimate

Coefficient	Name	Description	Value for NPS AUV II
M_q		Drag moment about lateral body axis due to existing pitch ***q*** corresponding to laminar flow, approximately equals $M_{q\|q\|}$ at $1°$/sec	$M_{q\|q\|}\left(\frac{\pi}{180°}\right)$ estimate
Yaw moment coefficients			
$N_{\dot{v}}$		Fluid inertia moment about vertical body axis due to time rate of change of sway ($\dot{v}$)	-1.78 E-3
$N_{\dot{r}}$		Fluid inertia moment about vertical body axis due to time rate of change of yaw ($\dot{r}$)	-4.7 E-4
N_{uv}		Fluid inertia moment about vertical body axis due to existing sway ***v*** and surge ***u***	0.0
N_{ur}		Fluid inertia moment about vertical body axis due to existing yaw ***r*** and surge ***u***	-3.90 E-3
$N_{u\|u\|\delta rb}$		Drag moment about vertical body axis due to bow rudder deflection δ_{bs} and signed square of surge ***u***	$+0.283 \cdot L \cdot Y_{u\|u\|\delta rb}$
$N_{u\|u\|\delta rs}$		Drag moment about vertical body axis due to stern rudder deflection δ_{rs} and signed square of surge ***u***	$+0.377 \cdot L \cdot Y_{u\|u\|\delta rs}$
$N_{r\|r\|}$		Drag moment about vertical body axis due to signed square of existing yaw ***r*** corresponding to turbulent flow	-5.48 E-3 estimate

Coefficient	Name	Description	Value for NPS AUV II
N_r		Drag moment about vertical body axis due to existing yaw ***r*** corresponding to laminar flow, approximately equals $N_{r\|r\|}$ at $1°$/sec	$N_{r\|r\|}\left(\frac{\pi}{180°}\right)$ estimate
N_{prop}		Propeller yaw moment for NPS AUV II is normally zero due to twin propellers that are identically paired, offsetting and counterrotating. However N_{prop} yaw moments are not zero if paired propeller rpm values differ. Actual moments equal **($F_{propeller} \cdot y_{propeller}$)** for each propeller, now included in yaw equation of motion **(6.53)**.	Not used. Previous values are no longer applicable.

8. Modifications to Previous Dynamics Equations of Motion

Given these nomenclature definitions, the next task is to modify the dynamics equations of motion to group only body-acceleration-related terms on the left-hand sides, and group velocity-related terms on the right-hand sides. The algebraic transformations are similar for each of the six equations of motion. However the surge equation requires a number of important modifications and will be derived in detail. The surge equation describes the relationships between all forces affecting the linear body acceleration of the vehicle along the body longitudinal axis. The original surge motion equation of (Healey 93, appendix) includes accelerations on both sides and appears as follows:

Previous Surge Equation of Motion **(6.39)**

$$
\begin{aligned}
m\ [\dot{u} - vr + wq - x_G(q^2+r^2) + y_G(pq-\dot{r}) + z_G(pr+\dot{q})] \\
= \frac{\rho}{2}L^4\ [X_{pp}p^2 + X_{qq}q^2 + X_{rr}r^2 + X_{pr}pr] \\
+ \frac{\rho}{2}L^3\ [X_{\dot{u}}\dot{u} + X_{wq}wq + X_{vp}vp + X_{vr}vr \\
+ uq(X_{q\delta s}\delta_s + X_{q\delta b/2}\delta_{bp} + X_{q\delta b/2}\delta_{bs} + X_{r\delta r}ur\delta_r) + X_{r\delta_r}ur\delta_r] \\
+ \frac{\rho}{2}L^2\ [X_{vv}v^2 + X_{ww}w^2 + X_{V\delta r}uv\delta_r\ + uw(X_{w\delta s}\delta_s + X_{w\delta b/2}\delta_{bs} + X_{w\delta b/2}\delta_{bp}) \\
+ u^2(X_{\delta s\delta s}\delta_s^2 + X_{\delta b\delta b/2}\delta_{\delta b}^2 + X_{\delta r\delta r}\delta_r^2)] \\
+ (W-B)\ \sin\theta \\
+ \frac{\rho}{2}L^3\ X_{q\delta sn}uq\delta_s\epsilon(\eta)\ + \frac{\rho}{2}L^2\ [X_{w\delta sn}uw\delta_{sn} + X_{\delta s\delta sn}u^2\delta_s^2]\cdot\epsilon(\eta) \\
+ \frac{\rho}{2}L^2\ u^2X_{prop}
\end{aligned}
$$

The (Healey 93) equations of motion described and extended the earlier U.S. Navy Swimmer Delivery Vehicle hull number 9 (SDV-9) equations of motion (Smith 78, declassified), which were determined both empirically and theoretically. The ε **(η)** term in Equation (6.39) approximates a second-order speed-related SDV-9 propulsion response as observed in tow tank testing. Tow tank testing is atypical for most underwater vehicles. Similarly, $\delta_{b/2}$ terms are related to an nonstandard control arrangement in the SDV-9 that included independent control of port and starboard bow planes. A split bow planes control configuration is not unusual, but more often plane surfaces are controlled in pairs. The effects of individual planes have been combined as pairs in this model for simplicity. Therefore the ε **(**η**)** and $\delta_{b/2}$ terms are not included in the general model derived here.

Despite these reasonable simplifications it is worth noting that many existing underwater vehicles have asymmetries and unique characteristics which may not be fully captured by these general dynamics equations of motion. Additional modifications to the equations of motion may be necessary in some applications for proper characterization of different vehicle designs (such as individually controlled bow planes). For example, individual control of plane surface pairs will be necessary if active control of vehicle roll during cruise mode is attempted.

$\boldsymbol{X_{prop}}$ as defined in the original surge equation of motion of (Healey 93) composes a number of important variables including commanded speed, actual speed and drag. The $\boldsymbol{X_{prop}}$ formulation is not intuitive from the perspective of a general description of forces. Furthermore the composition of several variables as an apparent constant is very misleading. The following derivation algebraically reveals and rearranges the component variables making up the $\boldsymbol{X_{prop}}$ term. This reformulation permits distinguishing between propulsive force and drag force contributions occurring along the body longitudinal direction. Again from (Healey 93):

$$X_{prop} = C_{d0}\,(\eta\,|\eta| - 1) \tag{6.40}$$

$$\eta = \left(\frac{2\,ft/\sec}{700\;rpm}\right)\frac{n}{u} \tag{6.41}$$

where $\left(\frac{2\,ft/\sec}{700\,rpm}\right)$ can be referred to as *steady-state speed per maximum propeller rpm* ratio.

Combining (6.40) with (6.41) and expanding the last complete term contained in the (Healey 93) surge equation (6.39):

$$\begin{aligned}
\frac{\rho}{2}L^2u^2X_{prop} &= \frac{\rho}{2}L^2u|u|C_{d0}\ (\eta|\eta|-1) \\
&= \frac{\rho}{2}L^2u|u|C_{d0}\left[\frac{2\,ft/\sec}{700\,rpm}\,\frac{n}{u}\ \cdot\ \frac{2\,ft/\sec}{700\,rpm}\,\frac{|n|}{|u|}\ -\ 1\right] \\
&= \frac{\rho}{2}L^2C_{d0}\left[\left(\frac{2\,ft/\sec}{700\,rpm}\right)^2 n|n|\,\frac{u}{u}\,\frac{|u|}{|u|}\ -\ u|u|\right] \\
&= \frac{\rho}{2}L^2C_{d0}\left[\left(\frac{2\,ft/\sec}{700\,rpm}\right)^2 n|n|\ -\ u|u|\right]
\end{aligned} \quad \textbf{(6.42)}$$

The propulsion contribution (due to propeller rpm ***n***) and opposing drag contribution (due to forward surge velocity ***u***) are now evident. When the vehicle has two propellers, a pair of forward forces contribute to the expected speed per rpm, and the preceding $\boldsymbol{X_{prop}}$ term shown in Equation (6.42) is expanded to become:

$$\frac{\rho}{2}L^2u^2X_{prop} = \frac{\rho}{2}L^2C_{d0}\left[\left(\frac{2\,ft/\sec}{700\,rpm}\right)^2 \frac{1}{2}\left(n_{port}|n_{port}| + n_{stbd}|n_{stbd}|\right) - u|u|\right] \quad \textbf{(6.43)}$$

Force from a single propeller out of a pair is as follows. Corresponding yaw moment contributions by each of the propellers have been added to the yaw Equation (6.53).

$$\underset{(out\ of\ a\ pair)}{F_{single-propeller}} = \frac{\rho}{2}L^2C_{d0}\left[\left(\frac{2\,ft/\sec}{700\,rpm}\right)^2 \frac{1}{2}\left(n_{prop}|n_{prop}|\right) - u|u|\right] \quad \textbf{(6.44)}$$

Examination of Equations (6.43) and (6.44) reveals that, as forward velocity ***u*** increases, the effective forward thrust due to propeller rpm ***n*** decreases according to the expected signed square law, similar to a pump curve of shaft rpm versus pressure head. Note that these equations also accurately describe drag forces against forward motion when a moving vehicle's propellers are turned off. Extensive test tank experimental data is not needed for measuring this predominant relationship between propeller thrust and forward speed. A straightforward measurement of steady-state speed for maximum propeller rpm precisely quantifies this relationship.

Cross-body thruster propulsion terms have also been added to the dynamics equations of motion. Steady-state thruster force is closely proportional to the signed square of ordered motor voltage for the cross-body thrusters designed and constructed for the NPS AUV II (Cody 92) (Healey 94b). This signed square relationship between control voltage and effective thruster force is shown in Equation (6.45). The sign convention for thruster voltages is that positive voltage results in a force which pushes the vehicle in the positive direction of the body lateral or depth axes. More precise modeling of thruster nonlinearities and sinusoidal-exponential time response is possible using generalized tunnel thruster dynamics models (Cody 92) (Healey 94b) (Brown 93) (Belton 93) (Fossen 94). Dynamics-based models of thruster response must be used instead if thruster temporal response is significant. Similar results have been found for other thrusters that include thrust controller circuitry (Sagatun 91) (Marks 92). A nontemporal signed square voltage model was found to be reasonably accurate for the overall effects of the NPS AUV thrusters. Open loop test tank experiments can quantify installed thruster performance versus time with little difficulty.

Since an accurate force equation is available to model the four individual thrusters, force and moment terms can be added directly to the sway, heave, pitch and yaw equations of motion. Physical offsets of thruster centerline away from the vehicle centroid are multiplied against forces to obtain corresponding moments, as shown in Equation (6.46). Opposing moments due to forward and aft thrusters are accounted for by positive and negative thruster tube offset distances, respectively. This eliminates the need for the previous $\boldsymbol{N_{prop}}$ formulation.

$$F_{thruster} = \left(\frac{2.0 \ lb}{24^2 \ volts} \right) V_{thruster} \left| V_{thruster} \right| \tag{6.45}$$

$$Moment_{thruster} = \left(\frac{2.0 \ lb}{24^2 \ volts} \right) V_{thruster} \left| V_{thruster} \right| \cdot x_{\substack{offset \\ distance}} \tag{6.46}$$

The addition of thruster forces and moments is required to extend the (Healey 93) model to remain valid at low forward speeds (i.e. hovering mode). Corresponding damping moments must also be included to model the resistance of water against rotational motion in these directions. Previously existing drag terms each include surge $\boldsymbol{u}$ as a factor, and each approaches zero at the low forward speeds associated with hovering. Therefore new rotational damping drag terms must be included to account for skin friction, particularly at low speeds. $\boldsymbol{K_{pp}}$, $\boldsymbol{M_{qq}}$, and $\boldsymbol{N_{rr}}$ are coefficients for quadratic terms corresponding to turbulent boundary layer skin friction. $\boldsymbol{K_p}$, $\boldsymbol{M_q}$, and $\boldsymbol{N_r}$ are coefficients for linear terms corresponding to laminar boundary layer skin friction. As suggested by (Sagatun 91) (Fossen 94), all six of these skin friction damping terms have been added to rotational dynamics equations of motion (6.51) through (6.53) respectively. $\boldsymbol{K_{prop}}$ and $\boldsymbol{M_{prop}}$ terms are no longer needed, for reasons analogous to those presented for $\boldsymbol{N_{prop}}$ previously.

One additional function needed for the dynamics equations of motion is $\boldsymbol{U_{cf}}$, a normalizing quantity for cross-body fluid flow with respect to body distance $\boldsymbol{x}$ along the vehicle longitudinal axis. From (Healey 93):

$$U_{cf}(x) = \sqrt{(v + xr)^2 + (w - xq)^2} \tag{6.47}$$

Related functions ***h(x)*** and ***b(x)*** in Table 6.2 and the dynamics equations of motion are provided for the NPS AUV by a table of cross-sectional measurements (Marco 95).

This is an example of *strip theory* which divides the body of a submerged vehicle into multiple parallel strips, estimates hydrodynamic coefficients for damping and added mass over each strip, and then sums the contribution over each strip to produce overall coefficient estimates (Fossen 94). Alternative methods of calculating cross-body flow forces and moments appear in (Humphreys 91).

Some additional explanation is necessary for time-varying forces. So-called "added mass" forces are related to the resistance of the surrounding fluid to vehicle body acceleration. This physical behavior is predictable and reasonably intuitive: acceleration of the immediately adjacent water volume requires a corresponding force, and is thereby referred to as an "added mass" effect. These forces are only proportional to vehicle accelerations and not vehicle velocities. This characteristic of a rigid body interacting with a fluid medium helps to explain why the body frame mass matrix **[*M*]** (which corresponds to vehicle mass, moments of inertia and "added mass") is time invariant.

Replacement of the $\boldsymbol{X_{prop}}$ and similar terms, removal of the ε **(**η**)** and $\delta_{b/2}$ terms, including added mass terms, standardizing explicit nomenclature for hydrodynamics coefficients, and grouping body accelerations on the opposite sides from velocities now produces the desired form of the surge equation. Transformation of the remaining five equations of motion for sway, heave, roll, pitch and yaw is similarly performed by direct algebraic manipulation from those versions presented in (Healey 93). Thruster forces, thruster moments, propeller yaw moments and damping drag moments have been added where appropriate.

9. Dynamics Equations of Motion

The critical contribution of this chapter is the unambiguous definition of variables and coefficients, and a revised set of underwater vehicle dynamics equations of motion. These equations and the accompanying hydrodynamics model are implemented verbatim in the accompanying virtual world source code (Brutzman 94e). Final and complete forms for all six dynamics equations of motion follow.

Surge Equation of Motion (6.48)

$$\left(m - \frac{\rho}{2}L^3 X_{\dot{u}}\right)\dot{u} + m z_G \dot{q} - m y_G \dot{r}$$

$$= m\left[vr - wq + x_G(q^2 + r^2) - y_G pq - z_G pr\right]$$

$$+ \frac{\rho}{2}L^4\left[X_{pp}p^2 + X_{qq}q^2 + X_{rr}r^2 + X_{pr}pr\right]$$

$$+ \frac{\rho}{2}L^3\left[\begin{array}{l} X_{wq}wq + X_{vp}vp + X_{vr}vr + uq\left(X_{uq\delta b}\delta_{pb} + X_{uq\delta s}\delta_{ps}\right) \\ \qquad + ur\left(X_{ur\delta r}\delta_{rb} + X_{ur\delta r}\delta_{rs}\right) \end{array}\right]$$

$$+ \frac{\rho}{2}L^2\left[\begin{array}{l} X_{vv}v^2 + X_{ww}w^2 + uv\,X_{uv\delta r}\delta_{rs} + uw\left(X_{uw\delta b}\delta_{pb} + X_{uw\delta s}\delta_{ps}\right) \\ + u|u|\left(X_{u|u|\delta b\delta b}\delta_{pb}^2 + X_{u|u|\delta s\delta s}\delta_{ps}^2 + X_{u|u|\delta r\delta r}\delta_{rb}^2 + X_{u|u|\delta r\delta r}\delta_{rs}^2\right) \end{array}\right]$$

$$- (W - B)\sin(\theta)$$

$$+ \frac{\rho}{2}L^2 C_{d0}\left[\left(\frac{2\,ft/\sec}{700\,rpm}\right)^2 \frac{1}{2}\left(n_{port}|n_{port}| + n_{stbd}|n_{stbd}|\right) - u|u|\right]$$

Sway Equation of Motion (6.49)

$$\left(m - \frac{\rho}{2}L^3 Y_{\dot{v}}\right)\dot{v} + \left(-m z_G - \frac{\rho}{2}L^4 Y_{\dot{p}}\right)\dot{p} + \left(m x_G - \frac{\rho}{2}L^4 Y_{\dot{r}}\right)\dot{r}$$

$$= m\left[-ur + wp - x_G pq + y_G(p^2 + r^2) - z_G qr\right]$$

$$+ \frac{\rho}{2}L^4\left[Y_{pq}pq + Y_{qr}qr\right]$$

$$+ \frac{\rho}{2}L^3\left[Y_{up}up + Y_{ur}ur + Y_{vq}vq + Y_{wp}wp + Y_{wr}wr\right]$$

$$+ \frac{\rho}{2}L^2\left[Y_{uv}uv + Y_{vw}vw + u|u|\left(Y_{u|u|\delta rb}\delta_{rb} + Y_{u|u|\delta rs}\delta_{rs}\right)\right]$$

$$- \frac{\rho}{2}\int_{x\ tail}^{x\ nose}\left[C_{dy}\,h(x)\,(v + xr)^2 + C_{dz}\,b(x)\,(w - xq)^2\right]\frac{(v + xr)}{U_{cf}\,(x)}\,dx$$

$$+ (W - B)\cos(\theta)\sin(\phi)$$

$$+ \left(\frac{2\,lb}{24^2\,volts}\right)\left[V_{bow-lateral}|V_{bow-lateral}| + V_{stern-lateral}|V_{stern-lateral}|\right]$$

Heave Equation of Motion (6.50)

$$\left(m - \frac{\rho}{2}L^3 Z_{\dot{w}}\right)\dot{w} + m y_G \dot{p} + \left(-m x_G - \frac{\rho}{2}L^4 Z_{\dot{q}}\right)\dot{q}$$

$$= m\left[uq - vp - x_G pr - y_G qr + z_G(p^2 + q^2)\right]$$

$$+ \frac{\rho}{2}L^4\left[Z_{pp}p^2 + Z_{pr}pr + Z_{rr}r^2|\right]$$

$$+ \frac{\rho}{2}L^3\left[Z_{uq}uq + Z_{vp}vp + Z_{vr}vr\right]$$

$$+ \frac{\rho}{2}L^2\left[Z_{uw}uw + Z_{vv}v^2 + u|u|\left(Z_{u|u|\delta b}\delta_{pb} + Z_{u|u|\delta s}\delta_{ps}\right)\right]$$

$$- \frac{\rho}{2}\int_{x\ tail}^{x\ nose}\left[C_{dy}h(x)(v + xr)^2 + C_{dz}b(x)(w - xq)^2\right]\frac{(w - xq)}{U_{cf}(x)}\,dx$$

$$+ (W - B)\cos(\theta)\cos(\phi)$$

$$- \left(\frac{2\ lb}{24^2\ volts}\right)\left[V_{bow-vertical}|V_{bow-vertical}| + V_{stern-vertical}|V_{stern-vertical}|\right]$$

Roll Equation of Motion (6.51)

$$\left(m z_g - \frac{\rho}{2}L^4 K_{\dot{v}}\right)\dot{v} + m y_G \dot{w} + \left(I_x - \frac{\rho}{2}L^5 K_{\dot{p}}\right)\dot{p} - I_{xy}\dot{q} + \left(-I_{xz} - \frac{\rho}{2}L^5 K_{\dot{r}}\right)\dot{r}$$

$$= \left[-(I_z - I_y)qr - I_{xy}pr + I_{yz}(q^2 - r^2) + I_{xz}pq\right]$$

$$- m\left[y_G(-uq + vp) - z_G(ur - wp)\right]$$

$$+ \frac{\rho}{2}L^5\left[K_{pq}pq + K_{qr}qr + K_{p|p|}p|p| + K_p p\right]$$

$$+ \frac{\rho}{2}L^4\left[K_{|u|p}|u|p + K_{ur}ur + K_{vq}vq + K_{wp}wp + K_{wr}wr\right]$$

$$+ \frac{\rho}{2}L^3\left[K_{uv}uv + K_{vw}vw - u|u|\left(K_{u|u|\delta b}\delta_{pb} + K_{u|u|\delta s}\delta_{ps}\right)\right]$$

$$+ (y_G W - y_B B)\cos(\theta)\cos(\phi) - (z_G W - z_B B)\cos(\theta)\sin(\phi)$$

Pitch Equation of Motion (6.52)

$$m z_G \dot{u} + \left(-m x_g - \frac{\rho}{2} L^4 M_{\dot{w}}\right) \dot{w} + -I_{xy} \dot{p} + \left(I_y - \frac{\rho}{2} L^5 M_{\dot{q}}\right) \dot{q} - I_{yz} \dot{r}$$

$$= \left[-(I_x - I_z) pr + I_{xy} qr - I_{yz} pq - I_{xz} (p^2 - r^2)\right]$$

$$+ m \left[x_G(-uq + vp) - z_G(-vr + wq)\right]$$

$$+ \frac{\rho}{2} L^5 \left[M_{pp} p^2 + M_{pr} pr + M_{r|r|} r|r| + M_{q|q|} q|q| + M_q q\right]$$

$$+ \frac{\rho}{2} L^4 \left[M_{uq} uq + M_{vp} vp + M_{vr} vr\right]$$

$$+ \frac{\rho}{2} L^3 \left[M_{uw} uw + M_{vv} v^2 + u|u| \left(M_{u|u|\delta b} \delta_{pb} + M_{u|u|\delta s} \delta_{ps}\right)\right]$$

$$+ \frac{\rho}{2} \int_{x\ tail}^{x\ nose} \left[C_{dy}\, h(x)\, (v + xr)^2 + C_{dz}\, b(x)\, (w - xq)^2\right] \frac{(w - xq)}{U_{cf}(x)}\, x\, dx$$

$$- (x_G W - x_B B) \cos(\theta) \cos(\phi) - (z_G W - z_B B) \sin(\theta)$$

$$- \left(\frac{2\ lb}{24^2\ volts}\right) \begin{bmatrix} V_{bow-vertical} |V_{bow-vertical}| \cdot x_{bow-vertical} + \\ V_{stern-vertical} |V_{stern-vertical}| \cdot x_{stern-vertical} \end{bmatrix}$$

Yaw Equation of Motion (6.53)

$$
m y_G \dot{u} + \left(m x_g - \frac{\rho}{2} L^4 N_{\dot{v}}\right)\dot{v} + \left(-I_{xz} - \frac{\rho}{2} L^5 N_{\dot{p}}\right)\dot{p} + -I_{yz}\dot{q} + \left(I_z - \frac{\rho}{2} L^5 N_{\dot{r}}\right)\dot{r}
$$

$$
\begin{aligned}
= \quad & \left[-(I_y - I_x) pq + I_{xy}(p^2 - q^2) + I_{yz} pr - I_{xz} qr\right] \\
& - m\left[x_G(ur - wp) - y_G(-vr + wq)\right] \\
& + \frac{\rho}{2} L^5 \left[N_{pq}\, pq + N_{qr}\, qr + N_{r|r|}\, r|r| + N_r\, r\right] \\
& + \frac{\rho}{2} L^4 \left[N_{up}\, up + N_{ur}\, ur + N_{vq}\, vq + N_{wp}\, wp + N_{wr}\, wr\right] \\
& + \frac{\rho}{2} L^3 \left[N_{uv}\, Uv + N_{vw}\, vw + u|u|\left(N_{u|u|\delta rb}\delta_{rb} - N_{u|u|\delta rs}\delta_{rs}\right)\right] \\
& - \frac{\rho}{2} \int_{x\ tail}^{x\ nose} \left[C_{dy}\, h(x)\,(v + xr)^2 + C_{dz}\, b(x)\,(w - xq)^2\right] \frac{(v + xr)}{U_{cf}(x)}\, x\, dx \\
& + (x_G W - x_B B)\cos(\theta)\sin(\phi) + (y_G W - y_B B)\sin(\theta) \\
& + \left(\frac{2\ lb}{24^2\ volts}\right)\begin{bmatrix} V_{bow-lateral}\, |V_{bow-lateral}| \cdot x_{bow-lateral} + \\ V_{stern-lateral}\, |V_{stern-lateral}| \cdot x_{stern-lateral} \end{bmatrix} \\
& - F_{port-propeller} \cdot y_{port-propellor} - F_{stbd-propeller} \cdot y_{stbd-propeller}
\end{aligned}
$$

10. Mass and Inertia Matrix [M]

Matrix equations can now be written from the dynamics equations of motion (6.48) through (6.53), grouping significant terms together appropriately. The left-hand sides are simply written in matrix form as the product of the body coordinate frame mass matrix [***M***] and the time rate of change of velocities matrix $[\dot{V}]_{body}$. The force matrix [***F***] is a (6 × 1) matrix comprised of the right-hand sides of the six dynamics equations of motion.

The body coordinate frame mass matrix [***M***] is determined from the coefficients corresponding to linear and rotational components of $[\dot{V}]_{body}$ on the left-hand side of the given equations of motion **(6.48)** through **(6.53)**. When expressed properly, this mass matrix is time-invariant and does not include any velocity-related terms. All possible added mass terms are included here for completeness, even though many of the terms are likely to equal zero (Fossen 94).

Mass Matrix **(6.54)**

$$
[M] = \begin{bmatrix}
m-\frac{\rho}{2}L^3X_{\dot{u}} & -\frac{\rho}{2}L^3X_{\dot{v}} & -\frac{\rho}{2}L^3X_{\dot{w}} & -\frac{\rho}{2}L^4X_{\dot{p}} & mz_G-\frac{\rho}{2}L^4X_{\dot{q}} & -my_G-\frac{\rho}{2}L^4X_{\dot{r}} \\
-\frac{\rho}{2}L^3Y_{\dot{u}} & m-\frac{\rho}{2}L^3Y_{\dot{v}} & -\frac{\rho}{2}L^3Y_{\dot{w}} & -mz_G-\frac{\rho}{2}L^4Y_{\dot{p}} & -\frac{\rho}{2}L^4Y_{\dot{q}} & mx_G-\frac{\rho}{2}L^4Y_{\dot{r}} \\
-\frac{\rho}{2}L^3Z_{\dot{u}} & -\frac{\rho}{2}L^3Z_{\dot{v}} & m-\frac{\rho}{2}L^3Z_{\dot{w}} & my_G-\frac{\rho}{2}L^4Z_{\dot{p}} & -mx_G-\frac{\rho}{2}L^4Z_{\dot{q}} & -\frac{\rho}{2}L^4Z_{\dot{r}} \\
-\frac{\rho}{2}L^4K_{\dot{u}} & -mz_G-\frac{\rho}{2}L^4K_{\dot{v}} & my_G-\frac{\rho}{2}L^3K_{\dot{w}} & I_x-\frac{\rho}{2}L^5K_{\dot{p}} & -I_{xy}-\frac{\rho}{2}L^5K_{\dot{q}} & -I_{xz}-\frac{\rho}{2}L^5K_{\dot{r}} \\
mz_G-\frac{\rho}{2}L^4M_{\dot{u}} & -\frac{\rho}{2}L^4M_{\dot{v}} & -mx_G-\frac{\rho}{2}L^4M_{\dot{w}} & -I_{xy}-\frac{\rho}{2}L^5M_{\dot{p}} & I_y-\frac{\rho}{2}L^5M_{\dot{q}} & -I_{yz}-\frac{\rho}{2}L^5M_{\dot{r}} \\
-my_G-\frac{\rho}{2}L^4N_{\dot{u}} & mx_G-\frac{\rho}{2}L^4N_{\dot{v}} & -\frac{\rho}{2}L^4N_{\dot{w}} & -I_{xz}-\frac{\rho}{2}L^5N_{\dot{p}} & -I_{yz}-\frac{\rho}{2}L^5N_{\dot{q}} & I_z-\frac{\rho}{2}L^5N_{\dot{r}}
\end{bmatrix}
$$

The spatial distribution of mass within a body has several important effects which are quantified as moments of inertia. Calculation of inertial moments are as shown in Equations (6.55) through (6.60). In practice these calculations are performed as weighted sums, measured from vehicle origin to centers of mass for individual internal vehicle components. If the vehicle has a variable ballast system, changes of mass and inertial moment must be accounted for and then the body frame mass matrix **[*M*]** becomes slowly time-varying.

$$I_x = \int \left(y^2 + z^2\right) dm \tag{6.55}$$

$$I_y = \int \left(x^2 + z^2\right) dm \tag{6.56}$$

$$I_z = \int \left(x^2 + y^2\right) dm \tag{6.57}$$

$$I_{xy} = I_{yx} = \int xy \, dm \tag{6.58}$$

$$I_{xz} = I_{zx} = \int xz \, dm \tag{6.59}$$

$$I_{yz} = I_{zy} = \int yz \, dm \tag{6.60}$$

Mass matrix inversion can be accomplished via any of several algorithms (Press 92) (Hamming 86). Note that since the body frame mass matrix [***M***] is ordinarily time-invariant, the inverse mass matrix $[\boldsymbol{M}]^{-1}$ does not have to be determined repeatedly. Thus the computational efficiency of this large matrix inversion calculation has no effect on the real-time responsiveness of the hydrodynamics model algorithm. If total vehicle mass or inertial moment changes due to variable ballast or significant moving internal components, the matrix inversion calculation will have to be occasionally repeated and may impact real-time response.

11. Summary of Hydrodynamics Model Algorithm

All of the components of the general underwater vehicle real-time hydrodynamics model have been presented. Figure 6.5 summarizes the hydrodynamics model algorithm.

- Estimate and invert mass matrix **[*M*]** using **equation (6.54)**
- Initialize hydrodynamics model variables for posture **[*P*]**, velocities **[*V*]** and time rates of change of velocities using **Table 6.1**
- Loop until robot is done:
  - receive updated state vector from robot, including ordered effector values for rudders, planes, propellers, thrusters and elapsed time
  - Calculate new values for time rate of change of body velocities, using the current vehicle state vector and equation of motion right-hand sides using **Table 6.2, equations (6.24), (6.30), (6.35), and (6.48) through (6.53)**
  - Update velocities $[V]_{body}$ using **equation (6.31)**
  - Perform transformation to $[V]_{world}$ using **equations (6.5), (6.9), (6.10), (6.11), and (6.22)**
  - Update posture **[*P*]** using newly-calculated velocities $[V]_{world}$, ocean current estimate and previous posture using **equation (6.33)**
  - Return newly-calculated hydrodynamics values to robot via telemetry update of the robot state vector. Most calculated velocities and accelerations correspond to real-world values provided by inertial, flow and pressure sensors.
  - Wait for next updated state robot vector. Continue loop upon receipt. Shutdown when model is no longer required by robot.

Figure 6.5. Underwater vehicle real-time hydrodynamics modeling algorithm.

E. EULER ANGLE METHODS COMPARED TO QUATERNION METHODS

The hydrodynamics model presented here is based on Euler angle representations of vehicle orientation. Another possible representation method of interest is the unit quaternion. The use of quaternions is most notable for a lack of singularity when pointing vertically, and also for well-developed mathematics that permits rapid and efficient orientation update rates (Cooke 92b) (Kolve 93) (Chou 92) (Funda 90) (Shoemake 85). This section briefly describes quaternion mathematics as a possible alternative to Euler angle orientation calculations in the underwater vehicle hydrodynamics model.

The underlying mathematical reason that an Euler angle rotation matrix is unable to satisfactorily represent a vehicle pointing vertically (along the ***z***-axis) is that extraction of Euler angles provides a unique value for pitch ($\theta = \pm\pi$) but can only provide the sum ($\phi + \psi$, nose up) or difference ($\phi + \psi$, nose down) of roll and yaw, not unique values for each. Thus three parameters are inadequate to unambiguously represent all possible orientations as desired. Sir William Rowan Hamilton deduced and developed quaternion algebra in 1843 after searching many years for a generalization of complex numbers. He determined that four parameters are necessary to represent all possible orientations without potential mathematical singularity (Cooke 92b).

Consider the unit sphere as illustrated in Figure 6.6. Three parameters are necessary to describe a unit vector directed from the center to any point on the sphere surface. A fourth parameter can then be used to describe a value for rotation about this axis. This combination of unit vector and axial rotation uniquely defines all possible orientations, provided rotation values are specified to have a range $[0..2\pi)$ (Euler's Theorem).

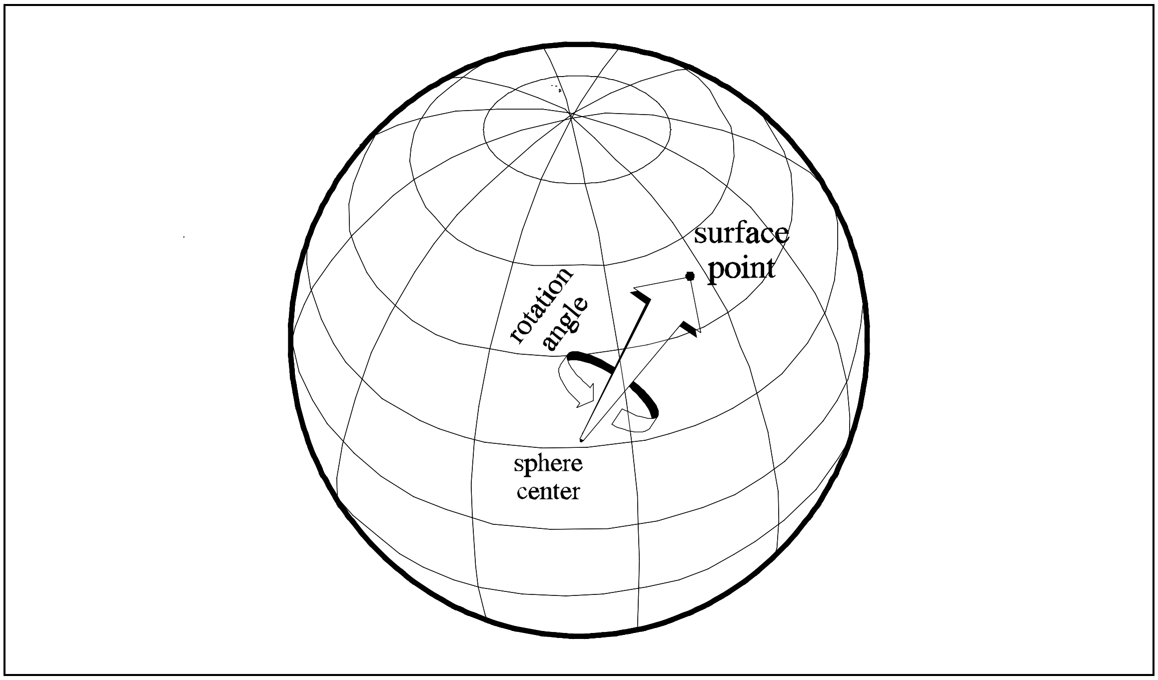

Figure 6.6. Quaternion representation.

There are several ways to represent quaternion values, described in detail in (Cooke 92a, 92b) (Kolve 93) (Chou 92) (Funda 90) (Maillot 90) and (Shoemake 85). The simplest representation is to scale three orthogonal unit vectors $\hat{i}$, $\hat{j}$, and $\hat{k}$ to indicate a point in three space, and then combine those three terms with another value for rotation about the described axis as follows:

$$Q = W + \hat{i}\,X + \hat{j}\,Y + \hat{k}\,Z \qquad \textbf{(6.61)}$$

The Euler parameter representation follows an Euler angle approach to state that three angles **A**, **B** and **C** can provide a rotation matrix that will align a rotation axis with the world coordinate frame. A fourth angle **D** describes rotation about this axis.

Rather than use **A**, **B**, **C** and **D** directly, a unit quaternion $\boldsymbol{Q}$ is represented using the following substitutions:

$$Q = \langle q_0,\ q_1,\ q_2,\ q_3 \rangle \tag{6.62}$$

where

$$\begin{aligned} q_0 &= \cos\left(\frac{\boldsymbol{D}}{2}\right) \\ q_1 &= \cos(\boldsymbol{A})\ \sin\left(\frac{\boldsymbol{D}}{2}\right) \\ q_2 &= \cos(\boldsymbol{B})\ \sin\left(\frac{\boldsymbol{D}}{2}\right) \\ q_3 &= \cos(\boldsymbol{C})\ \sin\left(\frac{\boldsymbol{D}}{2}\right) \end{aligned} \tag{6.63}$$

The four component values of quaternion $\boldsymbol{Q}$ are called *Euler parameters*. Expressing quaternions using Euler parameter form is desirable due to improved computational efficiency during arithmetic operations. Normalization may be periodically required after numerical calculations to ensure that magnitude of each unit quaternion vector remains equal to unity (Cooke 92b).

One important property of unit quaternions as described above is especially useful. Multiplication of two unit quaternions produces a new unit quaternion which represents the results of two successive corresponding rotations.

$$\begin{aligned} \boldsymbol{Q}\cdot\boldsymbol{Q}_1 &= \left(W + \hat{\imath}\,X + \hat{\jmath}\,Y + \hat{k}\,Z\right)\left(W_1 + \hat{\imath}\,X_1 + \hat{\jmath}\,Y_1 + \hat{k}\,Z_1\right) \\ &= \left(WW_1 - XX_1 - YY_1 - ZZ_1\right) \\ &\quad + \hat{\imath}\left(XW_1 + WX_1 - ZY_1 + YZ_1\right) \\ &\quad + \hat{\jmath}\left(YW_1 + ZX_1 - WY_1 + XZ_1\right) \\ &\quad + \hat{k}\left(ZW_1 + YX_1 - XY_1 + WZ_1\right) \end{aligned} \tag{6.64}$$

Angular velocity of a rigid body can be converted from body coordinate frame angular velocities to quaternion rates as follows:

$$\begin{aligned} \dot{q}_0 &= -\tfrac{1}{2}\left(q_1 p + q_2 q + q_3 r\right) \\ \dot{q}_1 &= \tfrac{1}{2}\left(q_0 p + q_2 r - q_3 q\right) \\ \dot{q}_2 &= \tfrac{1}{2}\left(q_0 q + q_3 p - q_1 r\right) \\ \dot{q}_3 &= \tfrac{1}{2}\left(q_0 r + q_1 q - q_2 p\right) \end{aligned} \qquad \textbf{(6.65)}$$

Given an initial orientation represented by a quaternion $\boldsymbol{Q}$, orientation updates are obtained by periodically integrating quaternion $\boldsymbol{Q}$ using quaternion rate $\dot{Q}$ and time step ($\delta \boldsymbol{t}$) via any numerical integration method.

Euler angles, if needed, are then extracted from the updated quaternion $Q_{(t_0 + \delta t)}$ as follows (Cooke 92b):

$$\theta = \sin^{-1}\left(-2\ (q_1 q_3 - q_0 q_2)\right) \qquad \textbf{(6.66)}$$

$$\psi = \cos^{-1}\left(\frac{q_0^2 + q_1^2 - q_2^2 - q_3^2}{\cos(\theta)}\right) \cdot \text{sign}\,(q_1 q_2 + q_0 q_3) \qquad \textbf{(6.67)}$$

$$\phi = \cos^{-1}\left(\frac{q_0^2 - q_1^2 - q_2^2 + q_3^2}{\cos(\theta)}\right) \cdot \text{sign}\,(q_2 q_3 + q_0 q_1) \qquad \textbf{(6.68)}$$

Note that the vertical restrictions on the range of pitch angle θ from Equation (6.2) remain unchanged in Equations (6.67) and (6.68) when converting from the quaternion representation back to Euler angles. Further mathematical manipulations of the quaternion will not produce values for ϕ or ψ. However, unlike the singularity in Euler rates at $\theta = \pm\pi/2$, there is no corresponding singularity in the quaternion rates of Equation (6.65).

The principal drawback to using quaternions in an underwater virtual world hydrodynamics model is greater computational complexity when calculating Euler angles, which are needed for networked posture update reports. The principal advantage of quaternion arithmetic is that computational complexity is less than Euler angle methods when solely calculating rotational updates (Cooke 92b). In the current implementation of the virtual world, Euler angles are required at every time step, in order to produce sensor values in the vehicle state vector and in order to provide DIS network updates. Thus Euler angle methods are used in the hydrodynamics model implementation (Brutzman 94e). These requirements might change if another vehicle without sucH sensors were modeled. If no virtual vehicle yaw, pitch or roll sensors are being modeled, or if DIS network updates are infrequent, the periodic computational drawback of quaternion conversions to Euler angles might become negligible. The mathematical methodology presented in this section demonstrated how to utilize quaternions for recording and updating orientation rotations in the hydrodynamics model, as an alternative to Euler angle methods. Detailed comparisons of computational efficiency including network considerations appear in (Cooke 92a, 92b).

F. DISTRIBUTED INTERACTIVE SIMULATION (DIS) AND NETWORK CONSIDERATIONS

Distributed Interactive Simulation (DIS) is the IEEE standard protocol (IEEE 93) used for communicating between networked entities sharing the same virtual environment. In order for a robot operating in a virtual world to be visible to other entities, DIS Protocol Data Units (PDUs) are sent out at regular intervals. The purpose and implementation of the virtual world DIS interface are presented separately in the network considerations chapter. This section examines the specific requirements of the hydrodynamics model that pertain to DIS.

The purpose of the hydrodynamics model is to provide valid real-time response to a networked robot operating in a virtual world. The hydrodynamics model is complex and sophisticated. A wide variety of subtle physical responses are possible. One current focus of research interest is examining the precise interactions that occur between robot and hydrodynamics models. Fine-grained reproduction of every interaction is therefore desirable for scientific purposes, if supportable by the network and virtual world viewer programs. Reproduction of AUV state at the same rate as interactions between the robot and the hydrodynamics model is correspondingly useful for visualization of both robot vehicle performance and hydrodynamics model performance. Currently this interaction rate is ten times per second (10 Hz).

The DIS protocol requires that entities announce their position at intervals not to exceed 5 seconds so that other entities are aware of their "live" presence (IEEE 93). In practice an interval of one to three seconds is typically used for entities such as ground vehicles which usually move with constant linear velocity. Highly dynamic vehicles such as jet aircraft may announce posture data many times per second in order to permit smooth refresh rates of rapidly varying postures (Towers 94). In order to reduce unnecessary network traffic, adaptive time steps between PDUs are recommended which only broadcast new values when predicted dead-reckoning error exceeds a reasonable threshold (or when the 5 second keep-alive deadline is reached). Choice of dead reckoning algorithm and other parameters can also reduce network loading (Lin 94). In general, minimizing PDU traffic is important to reduce network bandwidth, and also to reduce the processing load on each DIS receiver. These bandwidth considerations grow in importance when the number of actively participating entities becomes large, and also when using multicast DIS which can have world-wide Internet scope (Macedonia, Brutzman 94).

Although linear and rotational velocities and accelerations of an underwater vehicle are orders of magnitude lower than jet aircraft, underwater vehicle behavior is highly dynamic nevertheless. Example missions demonstrating highly complex interrelationships among vehicle state variables appear in the experimental results chapter and software distribution (Brutzman 94e). For some missions, frequent posture updates are necessary to closely evaluate vehicle interaction with hazardous environments in close quarters (such as a minefield). Precise posture information is also necessary to indicate interactions of propulsor flow and sonar sensors with the environment. Currently thrust, control plane and sonar values are embedded as "articulated parameters" within individual DIS entity state PDUs for the NPS AUV. Future versions of the DIS standard are expected to provide new PDU types specifically designed for announcing sonar transmissions, but hydrodynamics flow vectors (proportional to propulsor values) will continue to be inferred from the vehicle entity state PDU articulated parameter values.

Entity state PDUs must contain posture values and can optionally include linear velocity, angular velocity, and linear acceleration. Dead reckoning algorithm velocities and accelerations may be in world or body coordinates. Body accelerations are not explicitly defined, but (Towers 94) presents two dead reckoning algorithms pertaining to each of two possible body acceleration definitions. Of particular note are experimental results which show that average processing time of world coordinate frame PDUs is only 80% relative to body coordinate frame PDUs (Towers 94). On the other hand, a computational drawback in the use of world coordinate frame PDUs here is the fact that the underwater vehicle hydrodynamics model does not directly provide accelerations in the world coordinate frame. The current DIS implementation in the underwater virtual world utilizes world coordinate frame PDUs because they are more efficient for receivers and less expensive to render. Future work of interest includes implementing a selectable alternative encoding of entity state velocities and accelerations in body frame coordinates, and then empirically evaluating whether virtual world efficiency is degraded by shifting PDUs to body coordinates. Dead reckoning algorithm efficiency and evaluation is further discussed in (Lin 94).

G. OBJECT-ORIENTED NETWORKED RIGID BODY DYNAMICS CLASS HIERARCHY

Physically based modeling includes dynamics (modeling forces and accelerations) as well as kinematics (modeling velocity effects only). Dynamics considerations are a superset of kinematics. The implementation of the underwater vehicle hydrodynamics model was designed to incorporate the principles of object-oriented programming (encapsulation, inheritance and polymorphism) and structured programming (top-down design, modularity and data abstraction) as appropriate (Booch 91) (Barr 91) (Stroustrup 91) (Frakes 91) (Barzel 92) (Pohl 93) (Bailey 94). The many good design and software engineering principles found in these references were valuable in managing the complexity of the hydrodynamics model, and also in building a general dynamics model that can be easily adapted to other underwater vehicles (or even other vehicle types). Although no single software engineering methodology was rigidly adhered to, the resulting model implementation (written in ***C++***) enjoys most of the benefits which motivate these various references. Model structure is briefly presented here and further described in (Brutzman 94e).

Structuring the model design problem was the key to comprehensible implementation. A straightforward hierarchy follows. Posture is common to all vehicles and can be represented either by Euler angles, by Euler angles embedded in a homogenous transformation matrix (Fu 87) (Foley, van Dam 90), or by quaternions (Cooke 92b). A rigid body is subject to kinematics equations of motion which combine velocities with postures in strictly defined ways regardless of vehicle type or environmental dimensionality. A networked rigid body which communicates with other entities via DIS needs to calculate postures, optional linear and rotational velocities, and (again optional) linear accelerations. Such a DIS-networked rigid body has identical capabilities regardless of vehicle type. An entity dynamics component for a real-time networked virtual world combines the functionality of rigid bodies and DIS networking with the dynamics equations of motion (forces and accelerations) unique to a specific vehicle type. This structured hierarchy of relationships between posture representations, rigid bodies, DIS networking and dynamics equations of motion led to the general model class diagram which appears in Figure 6.7.

The compartment boxes in Figure 6.7 delineate the functionality of class components. The first compartment is class name. The second compartment indicates member data fields, which are the data structures encapsulated by the object. The third compartment indicates object methods (functions) which effectively occur instantaneously. The fourth compartment includes methods (functions) which are time-consuming, either from the perspective of simulation clock duration or actual delay due to network latency. Adapted from the Object-Oriented Simulation Pictures (OOSPICs) design and testing methodology (Bailey 94), this diagraming approach is very useful because it simplifies presentation of key object relationships and clarifies hierarchy design. Of particularly value is the explicit specification of temporal relationships, which are critical to success in a real-time system and are often overlooked in complex system design. An example object template which adapts the OOSPICs methodology from ***MODSIM*** programming language to ***C++*** appears as Figure 6.8. A key for OOSPIC arrow conventions is included in Figure 6.9 (Bailey 94). Software source code throughout the hydrodynamics class library implementation (Brutzman 94e) follows the structural layout presented in the OOSPIC diagram of Figure 6.8.

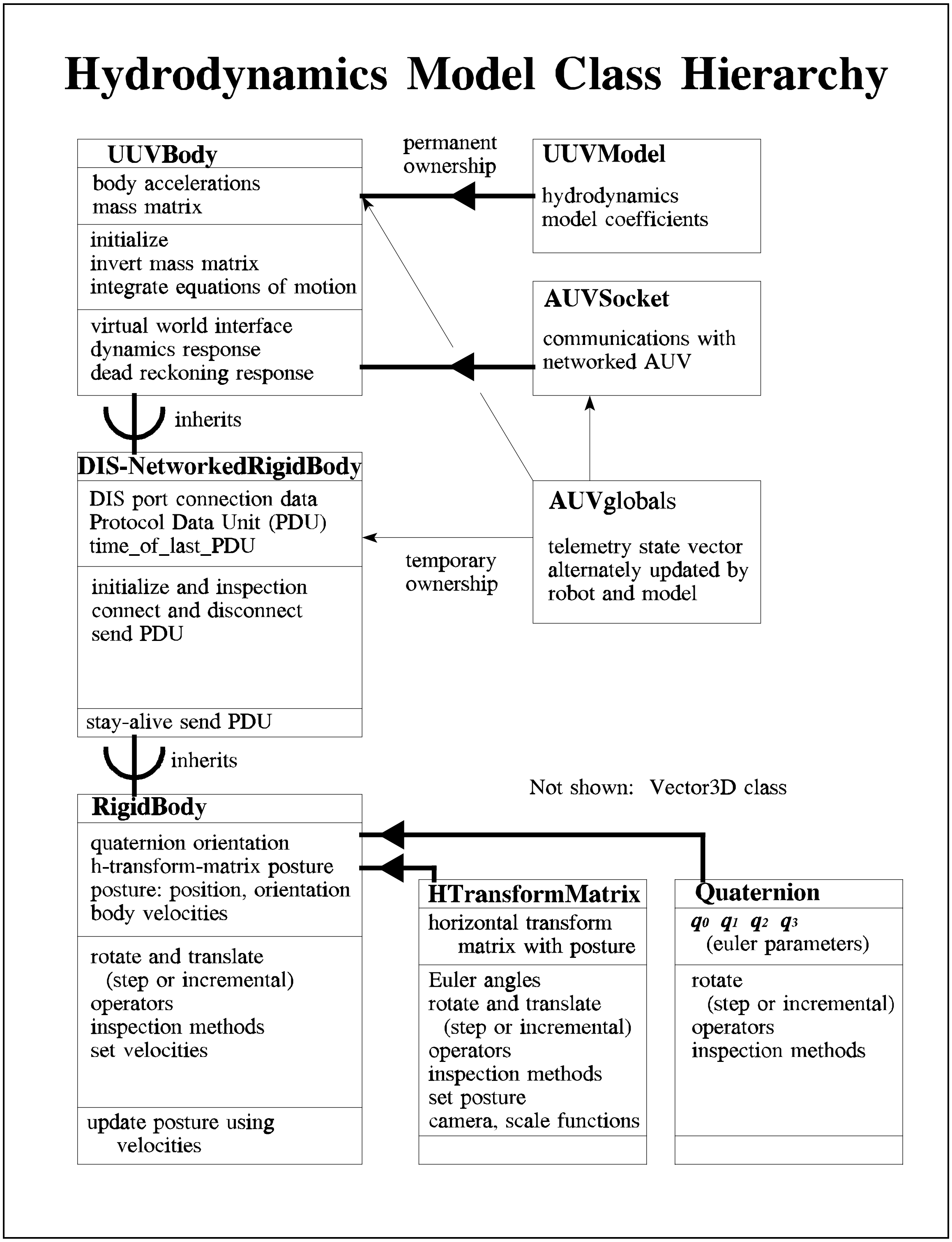

Figure 6.7. General real-time DIS-networked hydrodynamics model class hierarchy.

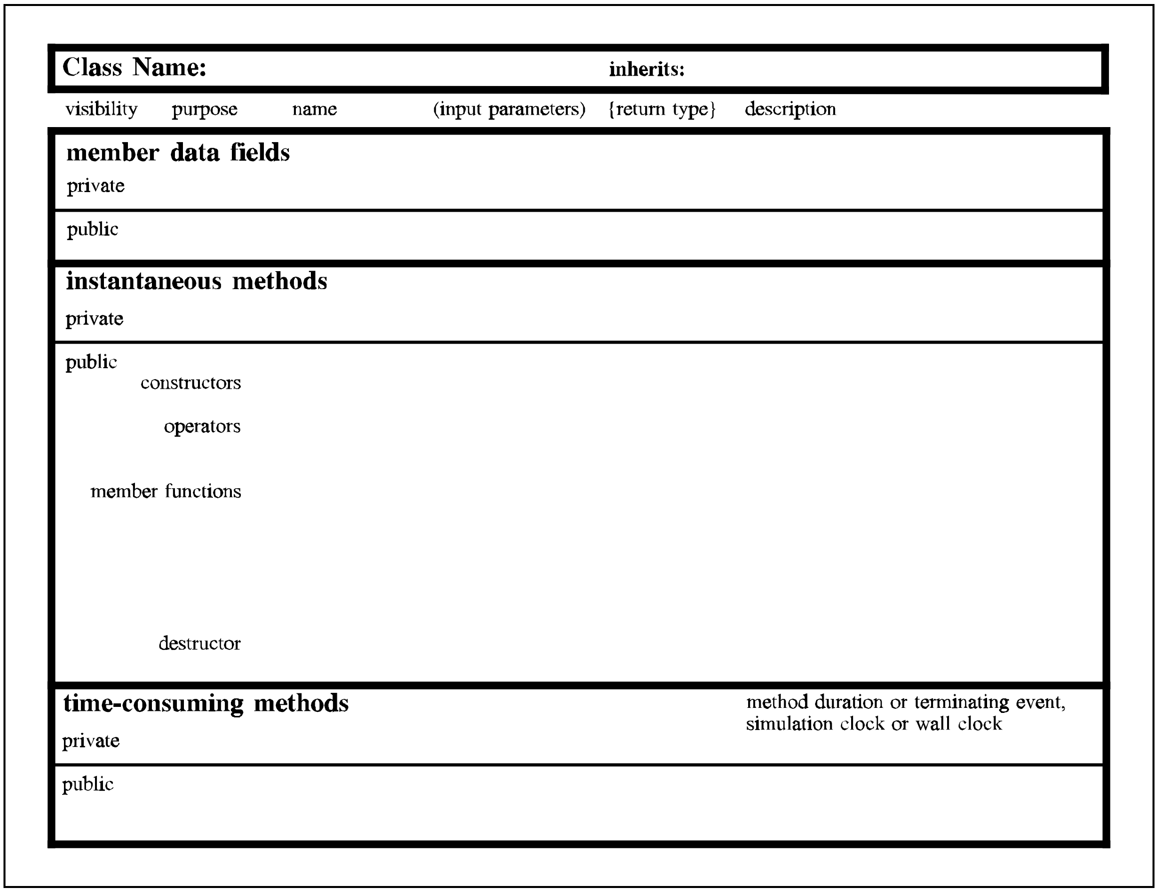

Figure 6.8. OOSPIC class diagram template for ***C++*** class definitions. Separation of class name, data fields, instantaneous methods and time-consuming methods clarifies class functionality and design.

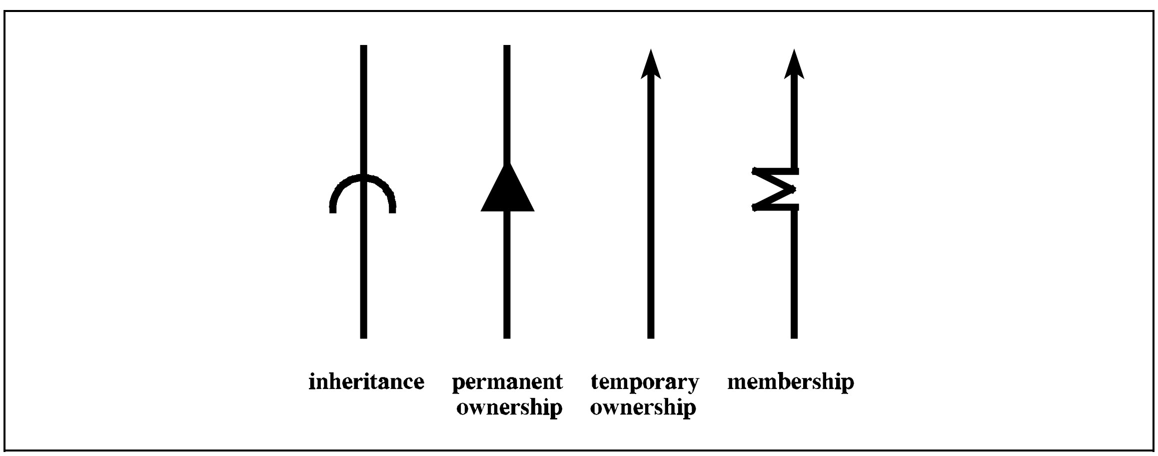

Figure 6.9. Object-Oriented Simulation Pictures (OOSPICs) arrow conventions.

The structure of the general real-time DIS-networked dynamics model presented here appears to be applicable to vehicles of arbitrary type. Documented source code matches the diagrams, equations and algorithms presented in this work (Brutzman 94e). Future work of interest in model design includes determining new parameter values using this model to emulate the characteristics of other underwater vehicles, adapting the model to accommodate dissimilar vehicle entities, porting the model into robot software as an on-board hydrodynamics response predictor, and investigating extensions to the model to support visualization, validation and verification of model relationships against archived or live data records of actual vehicle dynamics performance.

H. SIMULATING ON-BOARD INERTIAL SENSORS

Navigation and position keeping are fundamentally important capabilities for an AUV. Unfortunately the selection, purchase, installation, calibration, testing and interpretation of sensors is time consuming and expensive. A valuable benefit of a networked hydrodynamics model is that it can provide model values for "virtual sensors" which may or may not be physically installed.

There are three types of navigational sensors in common use: sonar, electromechanical and inertial. Navigational sonar sensors either detect the environment or use doppler difference ranging from beacons at known locations, and as such are not appropriately modeled using hydrodynamics parameters. Mechanical or electrical sensors for water flow, depth pressure, plane position, propulsor rpm and battery amp-hour consumption rate are directly represented by model variables for surge $\boldsymbol{u}$, depth $\boldsymbol{z}$ and vehicle state vector values respectively. Normally these sensors are reasonably accurate with zero bias and less than 5-10% error over their operating range. Inertial and gyroscopic detectors can also be modeled but additional considerations pertain.

Inertial navigation sensors are often called "strap-down" systems since they are aligned with vehicle body coordinates and physically fixed to the vehicle frame.

If possible they are kept near the center of gravity to minimize offset moment effects. Complete packages using solid state sensors and integrated circuit processing are now available at relatively low cost, providing angular rate and acceleration values about all three body axes. Many other small inertial units are also available which can provide similar functionality for one or two body axes. Velocity outputs are integrated internally from accelerations, and posture values are then found by integration of velocity.

Accuracy of inertial devices depends on pitch and roll angle estimation and sensitivity to acceleration. Inertial accelerometers are affected both by accelerations on the vehicle and accelerations due to gravity. Since the acceleration due to gravity is about ten times the acceleration of propulsors used by slow speed vehicles, an accurate estimate of vehicle pitch and roll is essential for isolating acceleration components unique to the vehicle. Because both position and rotation estimates are double integrations of accelerations, any noise or error in acceleration estimation is greatly amplified over the passage of time. Proper conversion from local inertial reference frames to geostationary or geocentric inertial reference frames is also necessary (Maloney 88). Additional errors and correction factors all can raise the complexity of the sensor and its model.

Electromechanical and inertial sensors can be precisely modeled by perfect "virtual sensors" using the hydrodynamics model. This is very useful for initial experimentation with navigation functions on the robot. For realistic modeling, however, accurate distributions for sensor bias, error and variance are needed. Such distributions can only be meaningfully applied using specifications and test results for actual hardware. Error models are feasible (Pappas 91) (Brancart 94) and can be modeled statistically (Law, Kelton 91). Simulating "virtual sensors" using the hydrodynamics model is of particular usefulness when evaluating robust vehicle control under variable operating conditions (especially simulated sensor failure). The key to success when producing such simulations will be incorporating statistically valid error models.

I. SPECIAL EFFECTS AND FUTURE WORK: ROBUST CONTROL, TETHER, OCEAN SURFACE, COLLISION DETECTION

The networked real-time availability of this model enables further work in several important research areas. The analysis and design of robust controllers focuses on producing stable performance when controlling multivariable systems with significant uncertainty (Dorato 87). Ordinarily this includes fixed control systems that meet performance measure criteria for specified uncertainty bounds. Example linear control algorithms used in the robot for posture control are included in the robot architecture chapter of this work. More sophisticated controller analysis appears in (Yoerger 85, 90) (Papoulias 89, 91) (Cristi 89) (Healey 89, 92b, 93, 94a) (Fossen 94) and numerous other references. Adaptive control methods and application of machine learning techniques to control are active areas of research (Goheen 87). This work is of particular interest given the paramount importance of vehicle stability despite any potentially chaotic (nonlinear instability) behavior which may emerge due to unforeseen interactions between multiple active controllers. The ability to repeatedly test controllers for yaw, depth, pitch, tracking and hovering while they are operating simultaneously on vehicle hardware in real time in the laboratory is a tremendous research tool provided by this model and the networked virtual world.

Although a tether is not ordinarily used on the NPS AUV, employment of a tether for power supply, task-level mission control or telemetry feedback can be very useful during vehicle testing. Tethers can also be a good way to prepare for using acoustic links, or to reliably test a vehicle in the open ocean prior to autonomous control. It is important to note that the operational characteristics of remotely operated vehicles are often dominated by tether dynamics. Incorporation of a tether injects significant forces and moments into the equations of motion, but tether forces can be realistically modeled (Abel 72) (Brancart 94) (Hover 94). Addition of a general tether model into this underwater vehicle hydrodynamics model is a valuable subject for future work.

Modeling ocean waves and surface interactions are also interesting areas for future work. Model complexity ranges from simple sinusoids to sophisticated numerical models obtained from supercomputer programs analyzing years of empirical oceanographic data (Covington 94) (Fossen 94) (Musker 88) (Blumberg 94). Usually the principle of superposition permits wave and current effects to be injected into the hydrodynamics model solution at the last algorithmic step, implying that highly complex ocean wave and circulation models can be solved off-line or in parallel. Incorporation of high-resolution ocean current models over computer networks is yet another worthy area for future research.

The hydrodynamics model presented here does not include collision effects. Abrupt changes in body acceleration and velocity may require extensions to the temporal integration algorithm. Detecting collisions and points of contact in a highly populated virtual world is a separate active research problem with an extremely high degree of computational complexity. Properly adapting the hydrodynamics algorithm to include realistic collision effects can be done meaningfully if performed in conjunction with the more general virtual world collision detection problem. This is another important area for future research.

J. SUMMARY

The requirements for a general networked underwater vehicle six degree-of-freedom hydrodynamics model are outlined for a robot connected to a virtual world. An overall comparison of vehicle dynamics in other environments shows that the underwater vehicle case is among the most difficult and crucial. No rigorous general model was previously available from a single source which is computationally suitable for real-time simulation of submerged vehicle hydrodynamics. The primary intended contributions of the hydrodynamic model developed here are clarity, correctness, generality, standardized nomenclature and suitability for real-time simulation.

Coordinate systems, variable definitions and coefficient nomenclature are explicitly defined. Kinematics equations of motion reveal constraints between representations in the body coordinate frame and world coordinate frame. Restrictions on Euler angles when pointing vertically are examined. Defining the underwater vehicle dynamics problem as a function of vehicle state vector and hydrodynamics state vector provides precise specifications of algorithm inputs and outputs. Dynamics equations of motion are derived in a form suitable for temporal integration in real time. Dimensionless coefficient values are presented for the NPS AUV and methods are discussed for determining coefficients of other vehicles. After extending previous work, a full set of component dynamics equations of motion are presented, including mass and inertia matrix determination. The dynamics equations of motion are in a form suitable for most existing underwater vehicles. Techniques are demonstrated for modifying these general equations to accommodate different vehicle physical configurations. Since the equations are written to run in real time, it may be computationally feasible to embed them in the robot execution logic as an onboard hydrodynamics response predictor for improved physical control.

Quaternion methods are examined as a possible alternative to Euler angle representations. The use of Distributed Interactive Simulation (DIS) network protocols for communication between virtual worlds imposes special considerations on the hydrodynamic model. An object-oriented networked rigid body dynamics class hierarchy illuminates the design and implementation of the hydrodynamics model. This class hierarchy may also be suitable for other types of networked vehicle models. Simulation of virtual sensors, robust control, tether considerations, ocean surface modeling and collision detection are all examined as possible components of the hydrodynamics model. Numerous considerations in these many areas are pointed out as useful candidates for future research, with the expectation that each can be implemented as compatible networked real-time extensions to the general underwater vehicle dynamics model.

VII. GLOBALLY NETWORKED 3D GRAPHICS AND VIRTUAL WORLDS

A. INTRODUCTION

Three-dimensional interactive graphics are ordinarily concerned with coordinating a handful of input devices while placing realistic renderings at fast frame rates on a single screen. Networking permits connecting virtual worlds with distributed models and completely diverse inputs/outputs on a truly global scale. Graphics and virtual world designers interested in large-scale interactions can now consider the world-wide Internet as a direct extension of their computer. A variety of networking techniques can be combined with traditional interactive 3D graphics to collectively provide almost unlimited connectivity. In particular, four component services are proposed as being necessary and sufficient for virtual world networking: reliable point-to-point socket communications, multicast communications protocols, interaction protocols such as the IEEE standard Distributed Interactive Simulation (DIS) protocol, and World-Wide Web connectivity.

The key specifications for virtual world networking are the application of appropriate network protocols and careful consideration of bandwidth. Distribution of virtual world components using point-to-point sockets enables upward scalability and real-time response. Multicast protocols permit moderately large bandwidths to be efficiently shared by an unconstrained number of hosts. Applications developed for the Multicast Backbone (MBone) permits open distribution of graphics, video, audio, DIS and other streams worldwide in real time. The DIS protocol enables efficient live interaction between multiple entities in multiple virtual worlds. The coordinated use of hypermedia servers and embedded World-Wide Web browsers allows virtual worlds global input/output access to pertinent archived images, papers, datasets, software, sound clips, text or any other computer-storable media. With these four network tools integrated in virtual worlds, 3D computer graphics can be simultaneously available anywhere.

B. NETWORKING BENEFITS

The benefits of networking a virtual world are many and worth enumerating. Any virtual world which attempts to model parts of the real world with nontrivial complexity will soon outstrip the computational capabilities and real-time capacity of any single computer. Heterogeneous processes need to be able to run on heterogenous processors. Massive archived datasets, sensor telemetry, component models, human users and autonomous entities can connect to the virtual world from wherever where they exist in the real world. This approach permits problem scalability, real-time response and interoperability. It also enables economies of scale since the structure of the virtual world can utilize an installed base of computers already connected to the Internet which numbers over twenty million. Since knowledge resource archiving and human access to the Internet is growing phenomenally at a sustained exponential rate of approximately 20% per month, virtual world design must address network connectivity and access efficiency in scalable ways.

C. BANDWIDTH SPECIFICATIONS FOR VIRTUAL WORLD NETWORKING

Three-dimensional computer graphics and network communications are both concerned with the delivery of information streams. In each case an all-encompassing criteria is bandwidth. In computer graphics, bandwidth concerns are manifested by frame rate, image size, level of detail, polygon culling and rendering complexity due to lighting models, texturing etc. The intended net result is delivery of effective visual information to a viewer. In networks, bandwidth is primarily measured by the information capacity of a channel in kilobits per second (Kbps) and is also affected by packet size, delivery latency, network loading, transport reliability and processor capacity. The net result is delivery of a information stream to one or multiple recipients.

It is useful to know the bandwidths of typical information streams since they can vary widely. Uncompressed video bandwidth transmitted on a network can consume as much as 60 Mbps. A 320x240 pixel 8 bit color video or graphics window reproduced by network video tool *nv* requires 128 Kbps for 1-3 frames per second, or 256 Kbps for 3-5 frames per second, where effective frame rate varies inversely with the number of pixels which vary from frame to frame. A telephone-quality audio channel (300-3300 Hz) requires 50-75 Kbps capacity depending upon the encoding algorithm employed. A musical instrument digital interface (MIDI) stream requires 32 Kbps. A representative entity DIS posture stream requires about 1 Kbps. One-time retrieval of data objects over the Internet has highly variable bandwidth which is principally dependent on the capacity of respective host connections and current intermediate network loading.

It is similarly important to know the capacity of various network connections. Most local area networks use Ethernet which has a maximum bandwidth of 10 Mbps. Fiber Distributed Data Interface (FDDI) is 100 Mbps. Microwave wireless bridges used to connect LANs typically have a bandwidth capacity of 1 Mbps. Modems on standard telephone lines can only support 2-20 Kbps. Typical fixed site connections to the Internet are T1 at 1.5 Mbps, or T3 at 45-155 Mbps (depending on whether electrical or optical signaling is used). Integrated services digital network (ISDN) lines are becoming available to business and home users, with line capacities measured in 64 or 128 Kbps increments up to a total of 1.5 Mbps. Frame Relay is a commercially available switching technique that supports best-effort delivery and variable-length data frames at bandwidths up to 2 Mbps. Broadband ISDN (BISDN) refers to Asynchronous Transfer Mode (ATM) (also known as Cell Relay) which uses fixed length data cells for switching bandwidths up to gigabits per second. Depending on contention-handling techniques used by the corresponding protocols, the effective bandwidth of each link type listed above may only be 80-90% of the theoretical maximum before collisions and collision recovery becomes prohibitive.

In every case, these various network connections are only of practical use to globally networked 3D graphics when they are compatible with the Internet Protocol (IP) suite. Given current implementations and eventual standardization of IP over

ATM (Armitage 94), IP compatibility exists for all of the listed connection types. Relatively high frame rate graphics can be generated over the Internet by low-end graphics workstations. Simultaneous duplication of graphics-related streams at both high and low bandwidths is feasible and desirable to accommodate these various bandwidth capacities. Duplicate imagery streams permits a variety of users to participate interactively via nearly any of the network connections listed above.

D. TERMINOLOGY AND NETWORK LAYERS

The integration of networks with computer graphics and virtual worlds occurs by invoking underlying network functions from within applications. Figure 7.1 shows

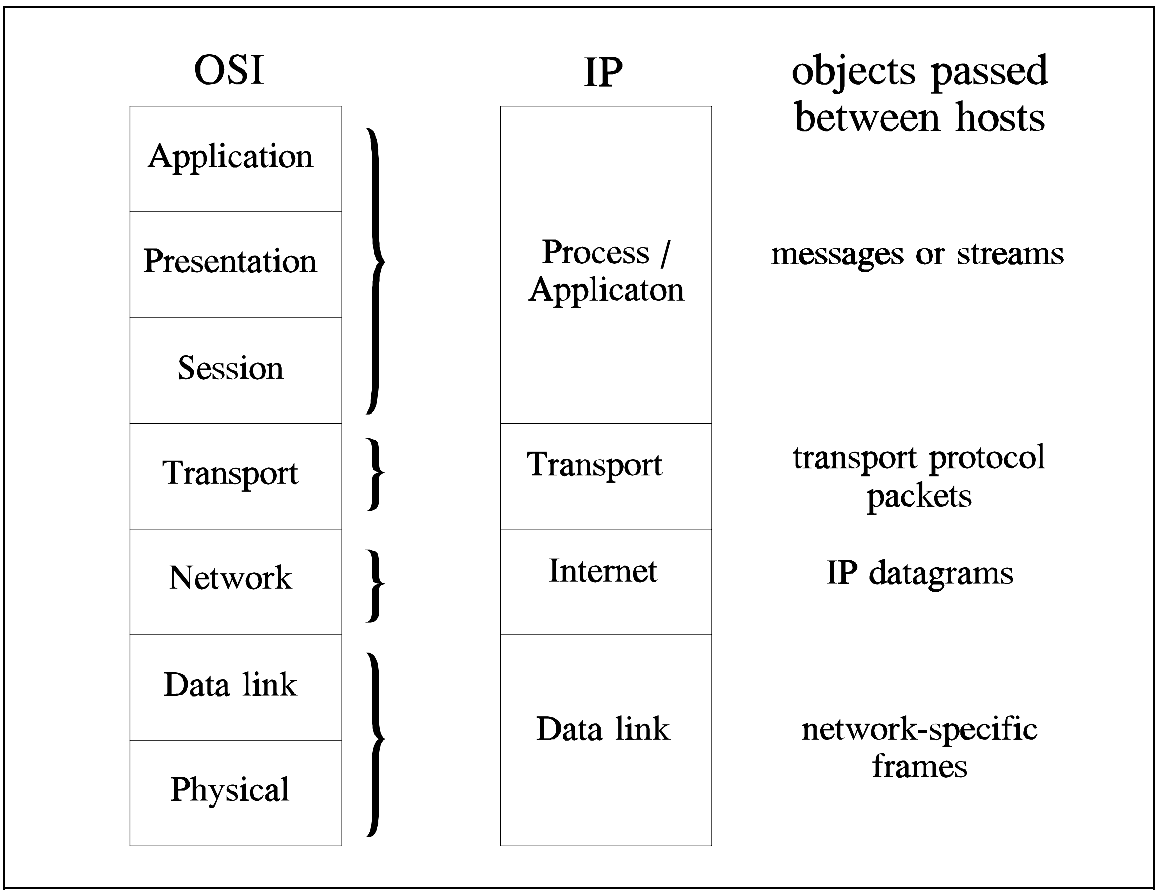

Figure 7.1 Correspondence between OSI and IP protocol layer models, and objects passed between corresponding layers on separate hosts.

how the seven layers of the well-known Open Systems Interconnection (OSI) standard network model generally correspond to the effective layers of the Internet Protocol (IP) standard. Functional characteristic definitions of the IP layers follow in Figure 7.2.

- **Process/Application Layer**. Applications invoke TCP/IP services, sending and receiving messages or streams with other hosts. Delivery can be intermittent or continuous.

- **Transport Layer**. Provide host-host packetized communication between applications, using either reliable delivery connection-oriented TCP or unreliable delivery connectionless UDP. Exchanges packets end-end with other hosts.

- **Internet/Network Layer**. Encapsulates packets with an IP datagram which contains routing information, receives or ignores incoming datagrams as appropriate from other hosts. Checks datagram validity, handles network error and control messages.

- **Data Link Layer**. Includes signaling and lowest level hardware functions, exchanges network-specific data frames with other devices. Includes capability to screen multicast packets by port number at the hardware level.

Figure 7.2. Summary of TCP/IP Internet layers functionality.

These diagrams and definitions are merely an overview but help illustrate the logical relationship and relative expense of different network interactions. In general, network operations consume proportionately more processor cycles at the higher layers. Minimizing this computational burden is important for minimizing latency and maintaining virtual world responsiveness.

Methods chosen for transfer of information must use either reliable connection-oriented Transport Control Protocol (TCP) or nonguaranteed delivery connectionless User Datagram Protocol (UDP). Each of these protocols is part of the Transport layer. One of the two protocols is used as appropriate for the criticality and timeliness of the particular stream being distributed. Understanding the precise characteristics of TCP, UDP and other protocols helps the virtual world designer understand the strengths and weaknesses of each network tool employed. A great deal more can be said about these and related topics. Since internetworking considerations impact all components in a large scale virtual world, additional study of network protocols and applications is highly recommended for virtual world designers. Suggested references include (Internet 94) (Stallings 94) (Comer 91) and (Stevens 90).

E. USE OF SOCKETS FOR VIRTUAL WORLD COMMUNICATION

The most common use of interprocess communications (IPC) among graphics and virtual world component processes is the socket. A socket is not a protocol but rather an application program interface (API) for communication between processes on different hosts (or a single host) via the network layer of the IP suite. Sockets provide a mechanism for passing data that is either reliable connection-oriented stream delivery, or nonguaranteed "best effort" connectionless datagram delivery. Interface details may vary between operating systems but socket syntax remains compatible and reasonably consistent on a variety of platforms.

Sockets originated with the Unix operating system as a way to make network communications syntactically similar to input/output, file and other stream operations. Implementing a connection-oriented socket usually requires three stages: open, read/write and close. Such socket use is not symmetric since sockets follow a client/server paradigm, where the server first opens a port and then waits for a client process to connect so that reliable two-way communication can begin. Normally sockets are used point to point between paired processes, such as tightly-coupled distributed virtual world components.

Connectionless sockets differ in that the ultimate destination address of the client need not be known by the server, with a corresponding lack of error detection and error recovery procedures to ensure reliable delivery. A connectionless approach is preferred when the data stream is continuous or in real time, since subsequent packets will automatically supersede and replace previous lost packets.

Broadcast protocols for socket communication are sometimes used for multiple-entity interaction. However such use is usually unacceptable due to indiscriminate consumption of bandwidth and unnecessary demand on processor cycles. The limitations of broadcast are the principal reasons for the current bottleneck in simultaneous communications among many entities. By way of analogy, consider the possibility that you were able to hear (and had to simultaneously listen to) every person speaking in the building where you work. It would be impossible to carry on any type of conversation since your ability to discriminate between speakers and words would be completely overwhelmed. A similar scenario occurs when large numbers of processes communicate indiscriminately via broadcast protocols: every process must receive and interpret every communication at the highest layers of the IP stack, and voluminous entity traffic produces a computational load that can eventually overwhelm processor capacity. Occasionally broadcast can be useful on a dedicated local area network among specific virtual world components, or among a limited number (dozens or perhaps a few hundreds) of entities. For large entity populations, it is necessary to avoid broadcast protocols and instead utilize multicast protocols, in order to logically partition the communication space and eliminate unnecessary interactions (Macedonia 94b).

F. MULTICAST PROTOCOLS AND THE MULTICAST BACKBONE (MBone)

IP multicasting is the transmission of IP datagrams to an unlimited number of multicast-capable hosts which are connected by multicast-capable routers. Multicast groups are specified by unique IP Class D addresses, which are identified by 1110_2 in the high-order bits and correspond to Internet addresses 224.0.0.0 through 239.255.255.255. Hosts choose to join or leave multicast groups and subsequently inform routers of their membership status. Of great significance is the fact that individual hosts control which multicast groups they monitor by reconfiguring their network interface hardware at the data link layer. Since datagrams from unsubscribed groups are ignored at the hardware interface, host computers can solely monitor and process packets from groups of interest, remaining unburdened by other network traffic (Comer 91) (Deering 89).

Multicasting has existed for several years on local area networks such as Ethernet and Fiber Distributed Data Interface (FDDI). However, with Internet Protocol multicast addressing at the network layer, group communication can be established across the Internet. Since multicast streams are typically connectionless UDP datagrams, there is no guaranteed delivery and lost packets stay lost. This best-effort unreliable delivery behavior is actually desirable when streams are high bandwidth and frequently recurring, in order to prevent network congestion and packet collisions. Example multicast streams include video, graphics, audio and DIS.

The ability of a single multicast packet to connect with every host on a local area network is good since it minimizes the overall bandwidth needed for large-scale communication. Note however that the same multicast packet is ordinarily prevented from crossing network boundaries such as routers. If a multicast stream that can touch every workstation were able to jump from network to network without restriction, topological loops might cause the entire Internet to become saturated by such streams. Routing controls are necessary to prevent such a disaster, and are provided by the recommended multicast standard (Deering 89) and other experimental standards. Collectively the resulting internetwork of communicating multicast networks is called the Multicast Backbone (MBone).

The MBone is a virtual network since it shares the same physical media as the Internet. A specially configured set of multicast-capable routers (mrouters) enables multicast packets to reach networks that have arranged for multicast connectivity. These mrouters can be upgraded commercial routers, or dedicated workstations running with modified kernels in parallel with standard routers. They are augmented by "tunneling," a scheme to encapsulate and forward multicast packets among the islands of MBone subnets through Internet IP routers that do not yet support IP multicast. The net effect of each routing scheme is identical for end users and applications: they can send and receive continuous multicast data streams throughout the MBone, and thus most of the Internet.

The MBone controls multicast packet distribution across the Internet in two ways: multicast packet hops through routers can be limited at the source using an attached time-to-live parameter, and sophisticated experimental mrouter pruning algorithms can adaptively restrict multicast transmission. Network administrators can also logically constrain the threshold capacity of multicast routes to avoid overloading physical link capacity. Multicast packet truncation is performed by decrementing the time-to-live (ttl) field each time the packet passes though an mrouter. A ttl value of 16 might logically limit a multicast stream to a campus, as opposed to values of 127 or 255 which might send a multicast stream to every subnet on the MBone (currently about 15 countries). A ttl field is sometimes decremented by large values under a global thresholding scheme provided to limit multicasts to sites and regions if desired.

Improved real-time delivery schemes are also being evaluated using the Real-time Transport Protocol (RTP) which is eventually expected to work independently of TCP and UDP (Schulzrinne 93). Other real-time protocols are also under development. The end result available today is that even with a time-critical application such as an audio tool, participants normally perceive conversations as if they are in ordinary real time. This behavior is possible because there is actually a small buffering delay to synchronize and resequence the arriving voice packets. Research efforts on real-time protocols and numerous related issues are ongoing, since every bottleneck conquered results in a new bottleneck revealed.

The MBone community must manage the MBone topology and the scheduling of multicast sessions to minimize congestion. Currently over 1500 subnets are connected worldwide. Topology changes for new nodes are added by consensus: a new site announces itself to the MBone mail list, and the nearest potential providers decide who can establish the most logical connection path to minimize regional Internet loading.

Scheduling MBone events is handled similarly. Special programs are announced in advance on an electronic mail list. Advance announcements usually prevent overloaded scheduling of Internet-wide events and alert potential participants. Cooperation is key. Newcomers are often surprised to learn that no single person or authority is "in charge" of either topology changes or event scheduling. Figure 7.3 shows a typical session directory (*sd*) list of programs available on the MBone.

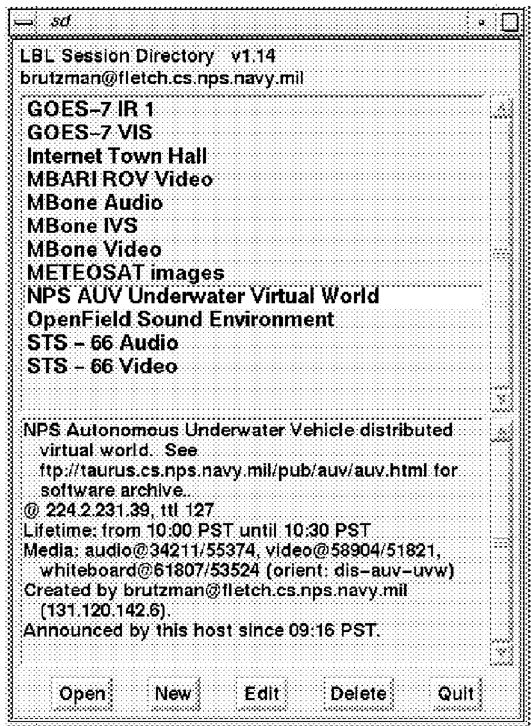

Figure 7.3 Session directory (*sd*) programs available on the MBone. Note DIS packets for NPS AUV Underwater Virtual World are sent over the whiteboard address (orientation: *dis-auv-uvw*).

Note session specifications in the advertisement window are used to automatically launch and connect video, audio, whiteboard, and DIS-compatible graphics viewer applications.

The MBone community is active and open. Work on tools, protocols, standards, applications, and events is very much a cooperative and international effort. Such cooperation is essential due to the limited bandwidth of many networks, particularly transocceanic links. So far, no hierarchical scheme has been necessary for resolving potentially contentious issues such as topology changes or event scheduling. Interestingly, distributed problem solving and decision making has worked on a human level just as successfully as on the network protocol level. Hopefully this decentralized approach will continue to be successful, even with the rapid addition of new users (Macedonia, Brutzman 94).

G. DISTRIBUTED INTERACTIVE SIMULATION (DIS) PROTOCOL USAGE

The Distributed Interactive Simulation (DIS) protocol is an IEEE standard for communication among entities in distributed simulations (IEEE 93, 94a, 94b). Although initial development was driven by the needs of military users, the protocol formally specifies the communication of physical interactions by any type of physical entity and is well-suited for general use. Information is exchanged using protocol data units (PDUs) which are defined for a large number of interaction types.

Multicast and broadcast DIS implementations are freely available and have been successfully utilized in real-time virtual battlefield exercises containing hundreds of active human and autonomous entities (Zeswitz 93) (Pratt 93, 94a) (Zyda 93b). Exploiting the features of multicast to logically partition DIS interactions in a manner similar to real world interactions is expected to permit scaling up virtual worlds to include 10,000 or more players (Macedonia 95a, 95b, 95c).

The principal PDU type is the Entity State PDU. This PDU encapsulates the position and posture of a given entity at a given time, along with linear and angular velocities and accelerations. Special components of an entity such as the orientation of moving parts can also be included in the PDU as articulated parameters. A full set of identifying characteristics can uniquely and completely specify the originating entity. A variety of dead reckoning algorithms permits computationally efficient projection of entity posture by listening hosts. Several dozen additional PDU types are also defined for simulation management, sensor or weapon interaction, signals, radio communications, collision detection and logistics support.

Of particular interest to virtual world designers is an optionally-addressable open format message PDU type. Message PDUs allow user-specified extensions to the DIS standard. Such flexibility coupled with the efficiency of Internet-wide multicast delivery permits extension of the object-oriented message-passing paradigm to a distributed system of essentially unlimited scale. Of related interest is ongoing research by the Linda project into the use of "tuples" as the communications unit for logical entity interaction (Gelernter 92a, 92b) (Carriero 90). It is reasonable to expect that free-format DIS message PDUs might also provide remote distributed connectivity resembling that of tuples to any information site on the Internet, further extended by using mechanisms which already exist for the World-Wide Web. This is a promising area for future work.

H. INTERNET-WIDE DISTRIBUTED HYPERMEDIA VIA THE WORLD-WIDE WEB (WWW)

The World-Wide Web (WWW) project has been defined as a "wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents" (Hughes 94). Fundamentally the WWW combines a name space consisting of any information store available on the Internet with a broad set of retrieval clients and servers, all of which can be connected by easily-defined hypertext markup language (*.html*) multimedia links. This globally-accessible combination of media, client programs, servers and hyperlinks can be conveniently utilized by humans or autonomous entities. The Web has fundamentally shifted the nature of information storage, access and retrieval (Berners-Lee 94a, 94b) (Hughes 94) (Vetter 94).

Universal Resource Locators (URLs) are a key WWW innovation (Figure 7.4). A block of information might contain text, document, image, sound clip, video clip, executable program, archived dataset or arbitrary stream. If that block of information exists on the Internet, it can be uniquely identified by host machine IP address,

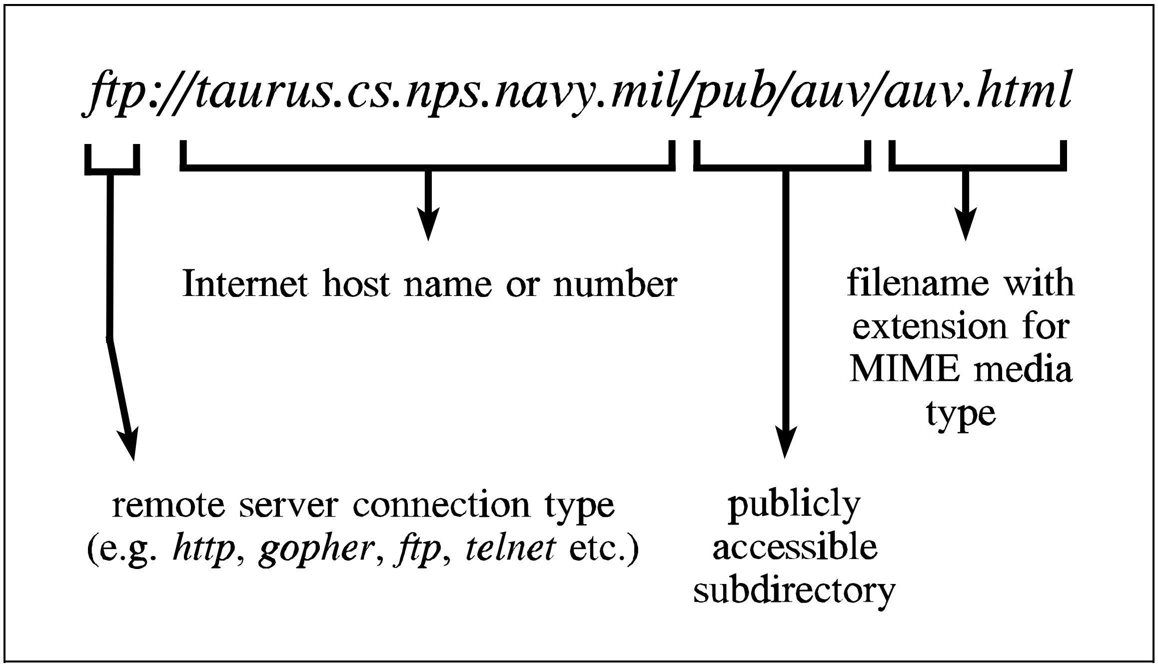

Figure 7.4. Example Universal Resource Locator (URL) components.

publicly visible local directory, local file name, and type of client needed for retrieval (such as anonymous ftp, hypertext browser or gopher). Ordinarily the local file name also includes an extension which identifies the media type (such as *.ps* for PostScript file or *.rgb* for an image). File type extensions are ordinarily specified by Multipurpose Internet Mail Extensions (MIME) (Borenstein 93). Thus the URL completely specifies everything needed to retrieve any type of electronic information resource. Example URLs appear in the list of references, e.g. (Hughes 94).

If one considers the evolving nature of the global information infrastructure, it is clear that there is no shortage of basic information. Quite the opposite is true. Merely by reading the *New York Times* daily, any individual can have more information about the world than was available to any world leader throughout most of human history! Multiply that single information stream by the millions of other information sources becoming openly available on the Internet, and it is clear that we do not lack content. Mountains of content have become accessible. What is needed now is context, some way to locate and retrieve related pieces of information or knowledge that a user needs in a timely fashion.

The World-Wide Web provides an open and easy way for any individual to provide context for the mass of content available on the Internet. For virtual world designers this is a particularly inviting capability. Virtual worlds are intended to model or extend the real world (Zyda 93a). Access to any media available world-wide can now be embedded in virtual worlds, enabling much greater realism and timeliness for virtual world inputs.

What about scaling up? Fortunately there already exists a model for this growing mountain of information content: the real world. Virtual worlds can address the context issue by providing information links similar to those that exist in our understanding of the real world. Furthermore, the structure and scope of a virtual world relationships can be dynamicly extended by passing WWW references over multicast network channels (e.g. as a DIS message PDU). This efficient distribution of information lets any remote user or component in a virtual world participate and interact in increasingly meaningful ways.

Extensions to the World-Wide Web to support globally distributed virtual reality and virtual world functionality are the subject of active investigation (Pesce 94). A Virtual Reality Modeling Language (VRML) specification and implementation is being developed by a large and informal working group (Bell 94). This group hopes to produce public browsers for the exploration of easily and consistently defined virtual worlds. The key components of VRML are likely to be a scene description language (e.g. modified *Open Inventor* file format), existing World-Wide Web functionality (e.g. *.html*), and entity behavior descriptions (e.g. *Open Inventor* engines), augmented by multicast communications (e.g. MBone) and active entity interaction protocols (e.g. DIS).

I. NETWORK APPLICATION IMPLEMENTATION EXAMPLES

Examples of the networked communication methods discussed here have been implemented in a distributed underwater virtual world, designed to support a single networked autonomous underwater robot while permitting any number of human observers. Remote participants use 3D real-time interactive computer graphics as a window into the underwater virtual world. Robot to virtual world communications are performed using a reliable stream socket. The virtual world provides real-time physically-based modeling of six degree-of-freedom vehicle hydrodynamics and sonar. Vehicle position and posture are output using multicast DIS 2.0.3 entity state PDUs. Remote graphics viewers can receive PDUs from any location on the MBone to render robot motion and virtual world interaction, again in real time, seen from whatever viewpoint each individual user might choose. Graphics windows and audio can also be multicast using standard MBone video and voice applications. A diagram of virtual world communication flows appears in Figure 7.5.

On the fly text-to-speech data sonification is provided using a WWW client which relays mission script commands to a sound server in the Netherlands (Belinfante 94). That remote sound server parses arbitrary text strings into phonemes and then generates a corresponding audio file, which is returned to the virtual world for local play. Text-to-speech sound queries are played and saved locally using a filename matching the original text, ensuring that network bandwidth consumption is minimized during repetitive queries.

A WWW home page provides free access to source code, binary executable programs, installation and help guides, reference papers and pertinent images to anyone with Internet access (Brutzman 94a, 94b, 94e). Modifications to the standard MBone session directory configuration file are also provided which enable remote MBone users to participate using the graphics viewer, DIS communications, default video stream, virtual world audio output and *Mosaic* display of the virtual world home page. All of these applications can be launched in concert with the click of a single button on the MBone session directory. As participation in remote virtual worlds

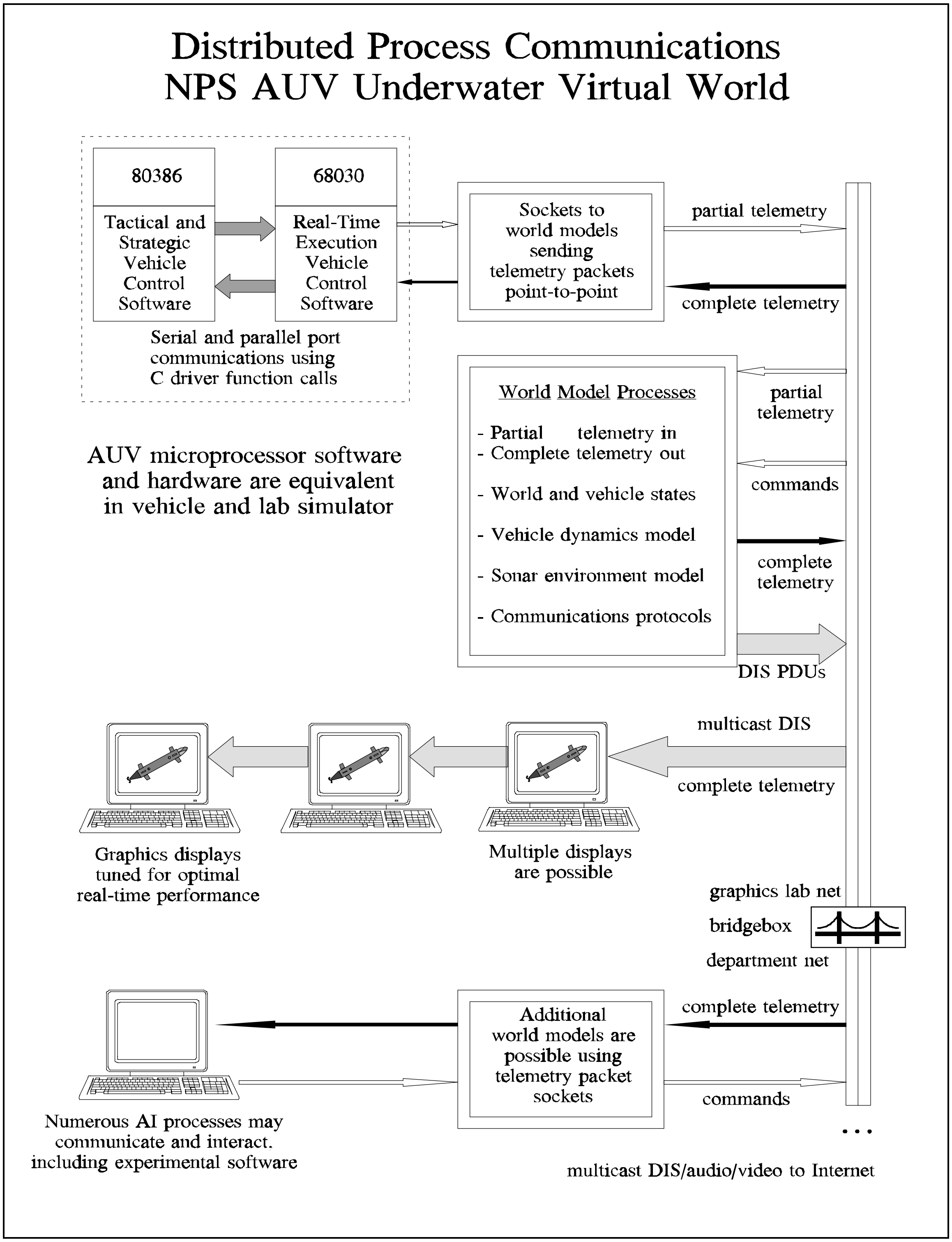

Figure 7.5. Distributed communications in NPS AUV Underwater Virtual World.

approaches the ease of use of a telephone, collaboration and participation in computer graphics-enhanced virtual worlds are expected to grow dramatically (Brutzman 94c, 94d) (Rhyne 94).

It is perhaps startling to hear someone say, "Here is an interactive multimedia television station that you can use to send out computer graphics and virtual world interactions between your desktop and the world." These are powerful concepts and powerful tools that extend our ability to communicate and collaborate tremendously.

J. SUMMARY AND FUTURE WORK

Four network components are proposed as being sufficient for global-distributed virtual world networking: sockets, multicast communications protocols, the Distributed Interactive Simulation (DIS) protocol and World-Wide Web connectivity. Sockets are best used for direct communication among tightly-coupled virtual world components and not for participants. Multicast protocols and the MBone provide efficient Internet-wide distribution of graphics, video, audio and DIS entity state information in a way that permits scaling up to very large numbers of active participants. DIS provides well-defined and standardized ways for physical interaction communications among multiple distributed entities in real time. The World-Wide Web enables virtual worlds to utilize as much of the real world as can be connected to the Internet, both as inputs and outputs.

A myriad of opportunities previously considered impossible are now becoming accessible. MBone, DIS and the World-Wide Web are changing the fundamental nature of the Internet. A distributed approach works both on a human level and a technical level. Scientific collaboration, shared experiences, simulation, training, education, virtual environments, high-bandwidth networked graphics, remote presence and telerobotics are all affected by these capabilities. Implementation of these concepts in an underwater virtual world has demonstrated their feasibility and value.

Open access to any type of live or archived information resource is available for use by individuals, programs, collaborative groups and even robots. Virtual worlds are a natural way to provide order and context to these massive amounts of information. World-wide collaboration works, both for people and machines. Finally, the network is more than a computer, and even more than your computer. The network becomes our computer as we learn how to share resources, collaborate and interact on a global scale.

VIII. SONAR MODELING AND VISUALIZATION

A. INTRODUCTION

This chapter describes the role of sonar modeling and sonar visualization in an underwater virtual world. The potentially significant effects of sound speed profile (SSP) on sound ray paths in the ocean are briefly examined, and example SSP plots are presented showing component measurements and possible ray path variations. Differences in sensor modalities and difficulties in forming mental models provide motivation for utilizing scientific visualization techniques to graphically render sonar. The necessity for a real-time sonar model makes the RRA algorithm (Ziomek 93, 94) appear to be a desirable choice based on offline results. Since short-range models are the most time-critical sonar application, an example geometric sonar model is presented for the NPS AUV test tank. A discussion of sonar parameter and graphics rendering considerations for sonar visualization is presented along with preliminary rendering examples. A great deal of important future work is possible in this area.

B. SOUND SPEED PROFILE (SSP)

The behavior of sound waves in the ocean is highly variable. Sound waves "bend" as they travel, away from the direction of higher sound speed and toward the direction of lower sound speed. This is an example of Snell's Law within a continuously varying medium. Since this bending may cause significant sound wave path changes, and since it does not occur uniformly over a wave front, the travel of sound through the ocean is highly nonlinear.

The primary factor influencing sound path is the sound speed profile (SSP). Water depth and bottom type can also have significant effects. Descriptions of SSP, water depth and bottom type effects on sound propagation are described in detail in (Etter 91) (Urick 83). Sound may be bent towards the bottom or surface, reflect off bottom or surface, be masked at certain depths by "shadow zones," travel for long ranges via convergence zones, or remain trapped in a deep sound channel.

The many ways that sound can travel in the ocean is highly variable. Assuming knowledge of local bathymetry, the primary information needed for sonar prediction is the SSP. Three factors control local sound speed: salinity, temperature and pressure. These parameters can be determined by measuring conductivity, temperature and density (each versus depth) directly in the water column. Empirical formulas have been determined which utilize conductivity, temperature and density to calculate sound speed. Typical SSP datasets are noisy and highly redundant, and large SSPs may be subsampled, smoothed or represented by polynomial approximations for computational tractability. Figure 8.1 shows a typical SSP plot taken from deep water in Monterey Bay in September 1990 along with component conductivity, temperature and density contributions (Rosenfeld 94). Figure Figure 8.2 shows the large possible variations in effects of an example SSP on ray paths, calculated by the RRA algorithm for a set of rays initially separated by only 0.4°.

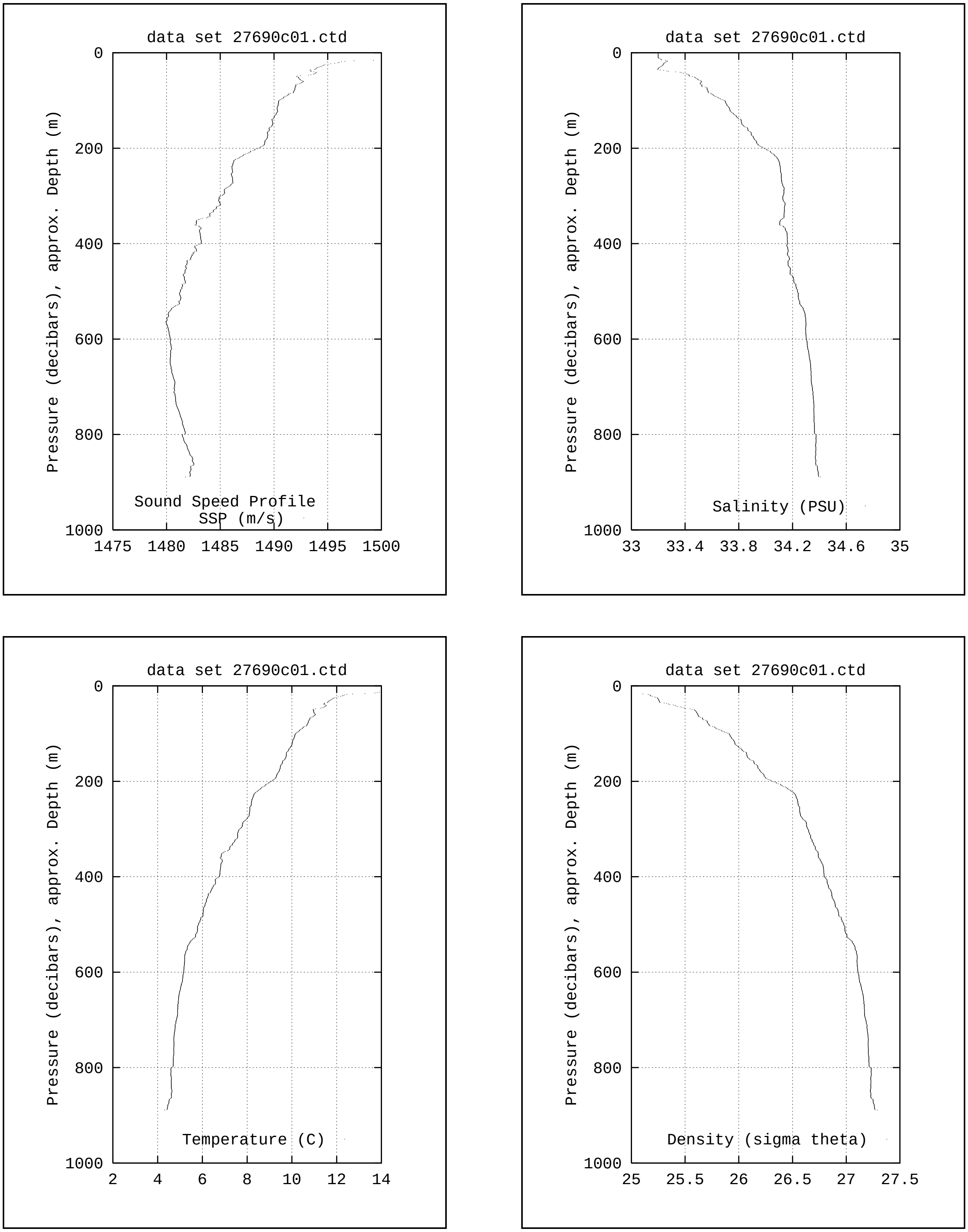

Figure 8.1 Representative sound speed profile (SSP) plot. Includes component conductivity (salinity), temperature and density (CTD) data plots (Rosenfeld 93).

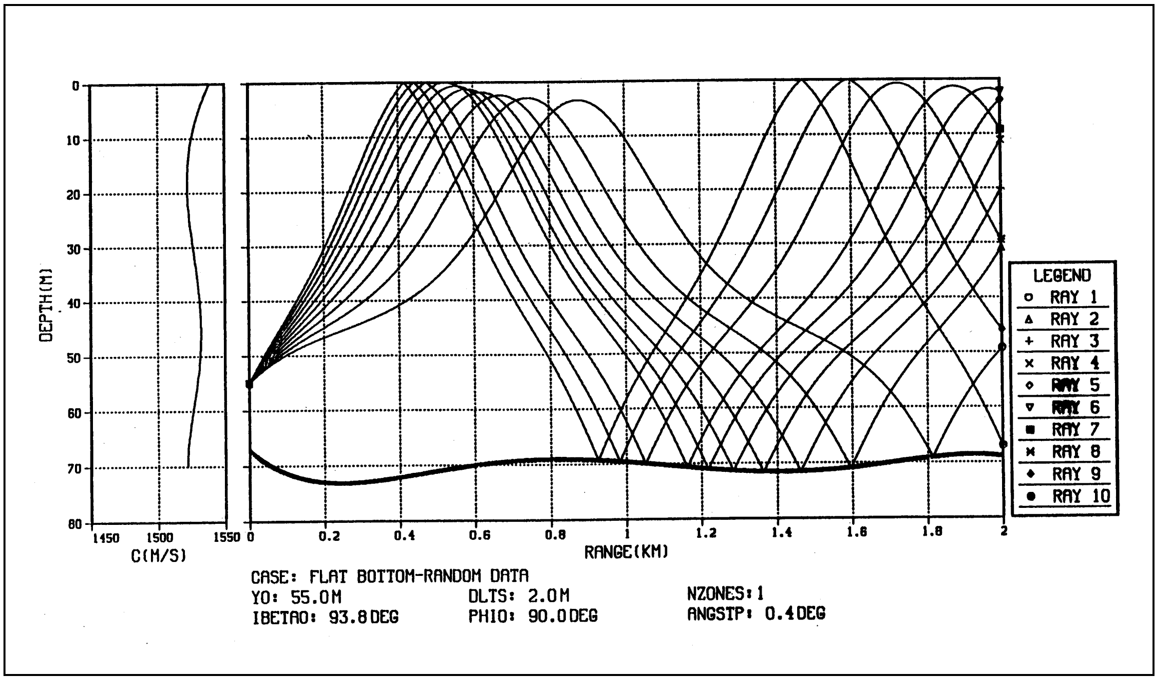

Figure 8.2. Example Recursive Ray Acoustics (RRA) algorithm plot showing sound ray bending due to sound speed profile (SSP) and bathymetry. Initial vertical orientation difference between rays is only 0.4° (Ziomek 93).

C. MENTAL MODELS AND SCIENTIFIC VISUALIZATION CONSIDERATIONS

The modalities of sonar sensing are much different from that of vision. For active sonar, ranges are measured by the time difference between pulse transmission and return detection. Multiplication of this time difference by the speed of sound in water provides a very accurate range estimate. For passive sonar, ranges to an object producing sound are not directly calculable but can sometimes be deduced by maneuvering and geometric analysis. For both active and passive sonars, bearings are typically accurate only within a few degrees. In contrast, vision techniques typically provide very accurate bearings with approximate ranges. As a result, perception algorithms based on range data and approximate bearing data are counterintuitive. Combined with the complexity of sound travel, it is difficult for individuals to visualize and conceptualize underwater sonar effectively. Sonar operators on submarines typically need a year of schooling and experience to qualify before their mental models become sufficiently familiar to permit unsupervised watchstanding (Brutzman 93b).

It is reasonably conjectured that improved sonar visualization can dramatically improve an individual's ability to understand the intricacies of sonar behavior. It is within current computational capabilities to calculate the physical path taken by sound through a highly variable sonar environment. Rendering the results using 3D computer graphics can provide useful feedback to human observers regarding sonar performance. Such feedback can enable the production of effective analysis and classification algorithms suitable for real-time autonomous use by AUVs (Brutzman 92a, 92e) (Compton 92).

D. REAL-TIME SONAR MODEL RESPONSE AND THE RECURSIVE RAY ACOUSTICS (RRA) ALGORITHM

As previously described in (Etter 91, 92) a great variety of sonar models exist, but unfortunately most are restricted to highly specific environmental domains. Additionally most sonar models are computationally expensive and are thus unsuitable for real-time performance. Implementation of an AUV sonar model within an underwater virtual world requires real-time response. Multiple model simultaneous real-time response in the virtual world can be accomplished through distribution on multiple processors if necessary. In practice at a 10 Hz rate, multiple processor distribution has not been necessary for the core models interacting directly with the AUV.

Interestingly, the speed of sound in water is relatively slow (typically 1650 yards/sec) compared to the speed of light in air. For active sonars, time of ping travel corresponds to twice the range to target plus any changes due to relative vehicle motion. This implies that approximately one second of processing time can be available for calculating each 800 yards of active sonar travel. Given that effective sonar ranges can be 10 miles or greater in distance, a great deal of computer time may be available for sonar calculations in tactical situations. In offline experiments, implementations of the RRA algorithm have demonstrated adequate computational performance. It is expected that implementation and integration of the RRA algorithm as an online model for active or passive sonar will meet all underwater virtual world timing requirements.

E. AN EXAMPLE GEOMETRIC SONAR MODEL

At short ranges, timing requirements can be critical. Fortunately at shorter ranges the effects of SSP on sound wave bending are negligible. Rapid calculation of sonar response at short ranges is possible through application of computational geometry techniques. An example geometric sonar model for the 20 ft by 20 ft NPS AUV test tank has been constructed which demonstrates adequate real-time response in this worst case scenario. The geometric model is capable of 10 Hz response without parallelization. A diagram of tank geometry appears in Figure 8.3. A graphics rendering of the NPS AUV ST-1000 sonar in the test tank as calculated by this model follows in Figure 8.4.

The following formulae are used to calculate the coordinates of the sonar echo return (R_x, R_y) based on sonar location (S_x, S_y) and sonar orientation ψ_{sonar}. The *precede* Boolean operator ($\prec$) returns TRUE if the first angle precedes the second angle by less than 180°, expressed algebraically as follows:

$$\{\alpha \prec \beta\} \equiv \{\text{normalize2}(\beta - \alpha) > 0\} \tag{8.1}$$

As previously defined in Chapter IV, *normalize2 (angle)* normalizes an angle to the range (-π/2..π/2].

For sonar-relative quadrant I (SA $\prec \psi_{sonar} \prec$ SB):

$$\begin{aligned} R_x &= 10 \\ R_y &= S_y + \sin(\psi_{sonar})\,(10 - S_x) \end{aligned} \tag{8.2}$$

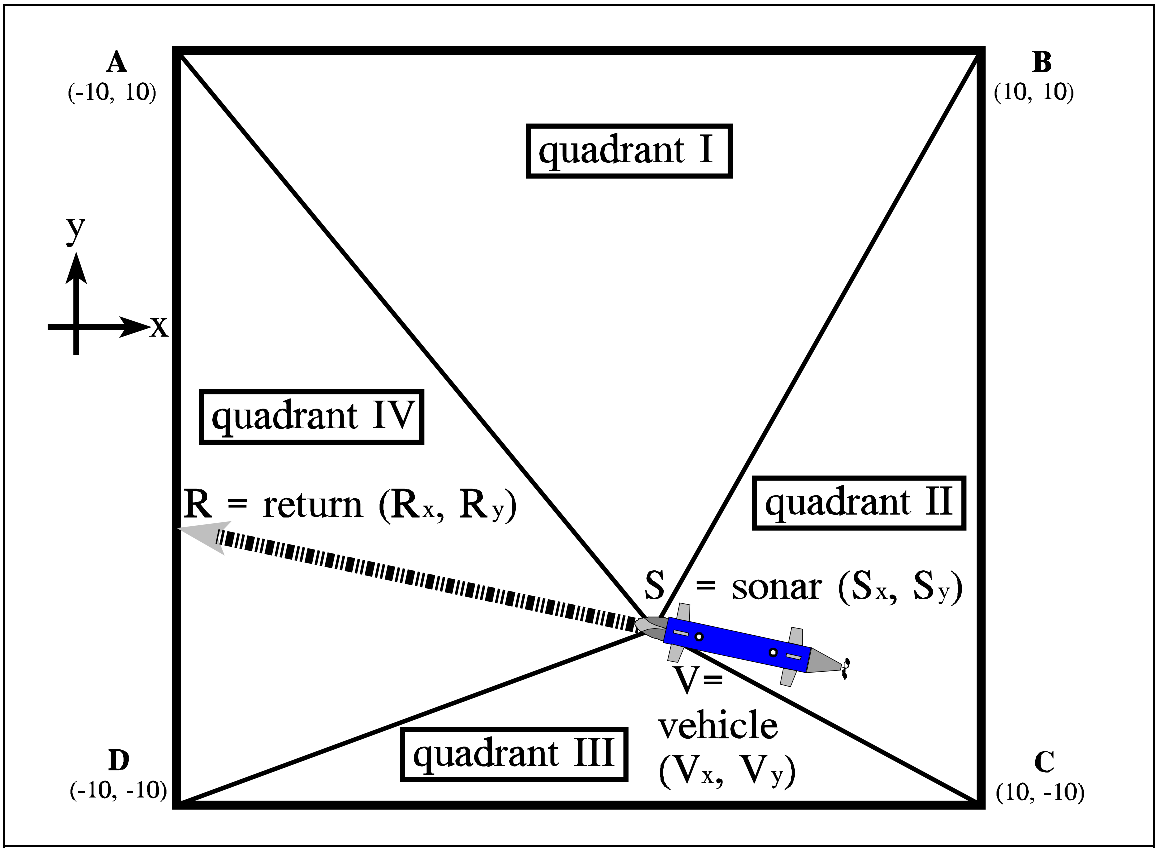

Figure 8.3. NPS AUV test tank geometry.

For sonar-relative quadrant II (SB $\prec \psi_{sonar} \prec$ SC):

$$
\begin{aligned}
R_x &= S_x - \sin(\psi_{sonar} - 90°)\,(10 - S_y) \\
R_y &= 10
\end{aligned}
\qquad \textbf{(8.3)}
$$

For sonar-relative quadrant III (SC $\prec \psi_{sonar} \prec$ SD):

$$
\begin{aligned}
R_x &= -10 \\
R_y &= S_y - \sin(\psi_{sonar} - 180°)\,(10 + S_x)
\end{aligned}
\qquad \textbf{(8.4)}
$$

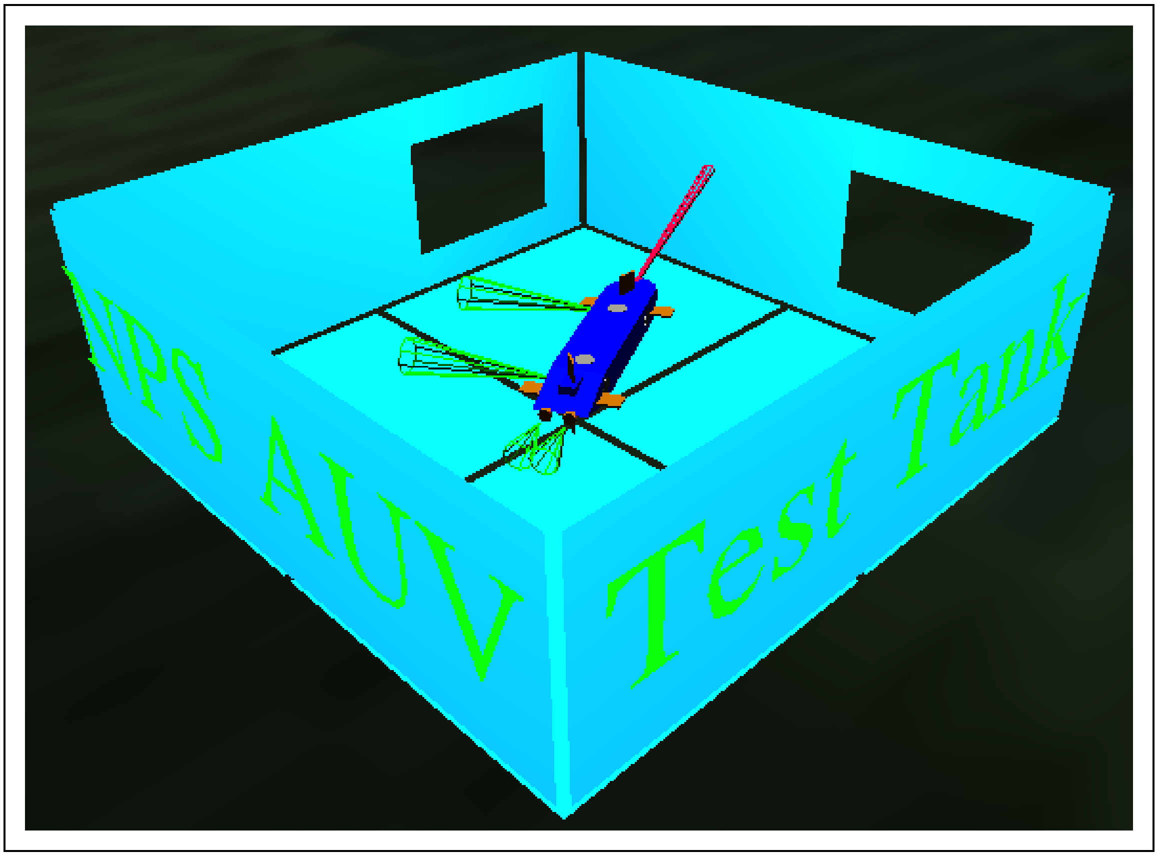

Figure 8.4. Sonar pointing towards test tank wall, as seen from behind AUV.

For sonar-relative quadrant IV $(SD \prec \psi_{sonar} \prec SA)$:

$$\begin{aligned} R_x &= S_x + \sin(\psi_{sonar} + 90°)\,(10 + S_y) \\ R_y &= -10 \end{aligned} \qquad \textbf{(8.5)}$$

Sonar offset coordinates (S_x, S_y) can be calculated from vehicle coordinates (V_x, V_y) using vehicle orientation ψ as follows:

$$\begin{aligned} S_x &= V_x + \cos(\psi)\,(x_{longitudinal\ sonar\ offset}) \\ S_y &= V_y + \sin(\psi)\,(x_{longitudinal\ sonar\ offset}) \end{aligned} \qquad \textbf{(8.6)}$$

Sonar range is determined using the Pythagorean theorem:

$$sonar\ range = \sqrt{(R_x - S_x)^2 + (R_y - S_y)^2} \qquad \textbf{(8.7)}$$

Development of individual geometric models for the large variety of objects populating a virtual world can be prohibitively laborious. For short and intermediate ranges, this problem is a variation of the virtual world collision detection problem which is solvable in real time for terrain and hundreds of objects (Pratt 93). Computationally efficient collision detection is the subject of active research for larger worlds (such as architectural ship design models) consisting of hundreds of thousands of objects (Zyda 93a). In an underwater environment the density of active entities is typically sparse, and sonar interactions are primarily concerned with terrain and a relatively small number of mobile entities. Thus geometric model switching corresponding to areas of interest in the underwater virtual world is a feasible approach.

Interestingly, graphics toolkits such as *Open Inventor* provide mechanisms for querying the scene database to determine ray intersection points (Wernecke 94a). Conceivably, the same scene database which is used to render the population of objects in the virtual world can also be used for sonar "collision" detection, perhaps independently of graphics rendering. This is a promising approach for automatic determination of sonar detections which is independent of the geometry of individual objects in the virtual world. Such an approach is also highly scalable through reasonably efficient construction or optimization of scene databases.

F. SONAR RENDERING FOR VISUALIZATION

Sonar data has high dimensionality and ordinarily is difficult to visualize. Scientific visualization methods specialize in the selective application of various graphical rendering techniques to extract the maximum possible information out of large and abstract datasets (Keller 93). Scientific visualization is therefore a direct example of a guiding precept in computer science:

> The purpose of computing is insight, not numbers. (Hamming 86)

A great many possibilities for sonar visualization present themselves. A preliminary consideration of sonar parameters and computer graphics attributes reveals a large number of relatively orthogonal characteristic parameters and primitive rendering operations. They are listed in Figure 8.5. Key criteria when rendering sonar

Sonar Parameters	Rendering Techniques
sound pressure level (SPL)	color variations
depth	intensity
absolute range of travel	transparency
downstream horizontal range	illumination
slant range	directional lights
signal excess for detection	individual rays
phase	wave fronts versus ray groups
pitch angle	density of ray bundles
target intersection	fog
history of previous returns	animation
SSP variations in temperature, salinity, pressure	volume visualization techniques
	blur
attenuation by absorption, scattering, spreading	data smoothing
	data enhancement/interpolation
pulse width	data sonification
frequency and doppler shift	loading and modifying images previously rendered offline
background noise, biologics, interference	

Figure 8.5. Preliminary listing of orthogonal sonar parameters and orthogonal computer graphics rendering techniques for scientific visualization.

data must include the ability to focus on individual parameters of tactical interest, matching orthogonal parameters to rendering techniques which are not mutually interfering, real-time response corresponding to short or long sonar ranges, animation of spatial or temporal changes, and selectable user control of either visualization primitives or sonar parameters of interest.

As a rudimentary example of sonar-related visualization, a rendering of SSP data appears in Figure 8.6. Formal application of scientific visualization techniques to sonar rendering is a promising topic for future work. It is likely that best results will be obtained by using sonar data structures which are equally suitable for online representation in the virtual world and offline rendering using visualization toolkits.

G. SUMMARY AND FUTURE WORK

Sonar modeling and sonar visualization are crucial components in an underwater virtual world for an autonomous underwater vehicle. Accurate real-time sonar modeling is necessary to produce realistic sensor interactions with the vehicle. Visualization is necessary for robot designers to create mental models of the often counterintuitive performance of sonar in highly variable ocean environments. Such mental models are of proven benefit when designing and evaluating robot sensing algorithms. SSP effects and an example geometric sonar model are also examined.

Promising areas for future work are dependent on successful incorporation of a general sonar model (or models) into the underwater virtual world. The RRA algorithm shows strong potential for rapid and accurate generation of sonar rays in real time. Additional work includes the formal use of scientific visualization techniques to match up typically orthogonal properties of sonar response to typically orthogonal rendering methods. It is expected that user control of parameters and combined offline/online algorithm analysis will be necessary for best results.

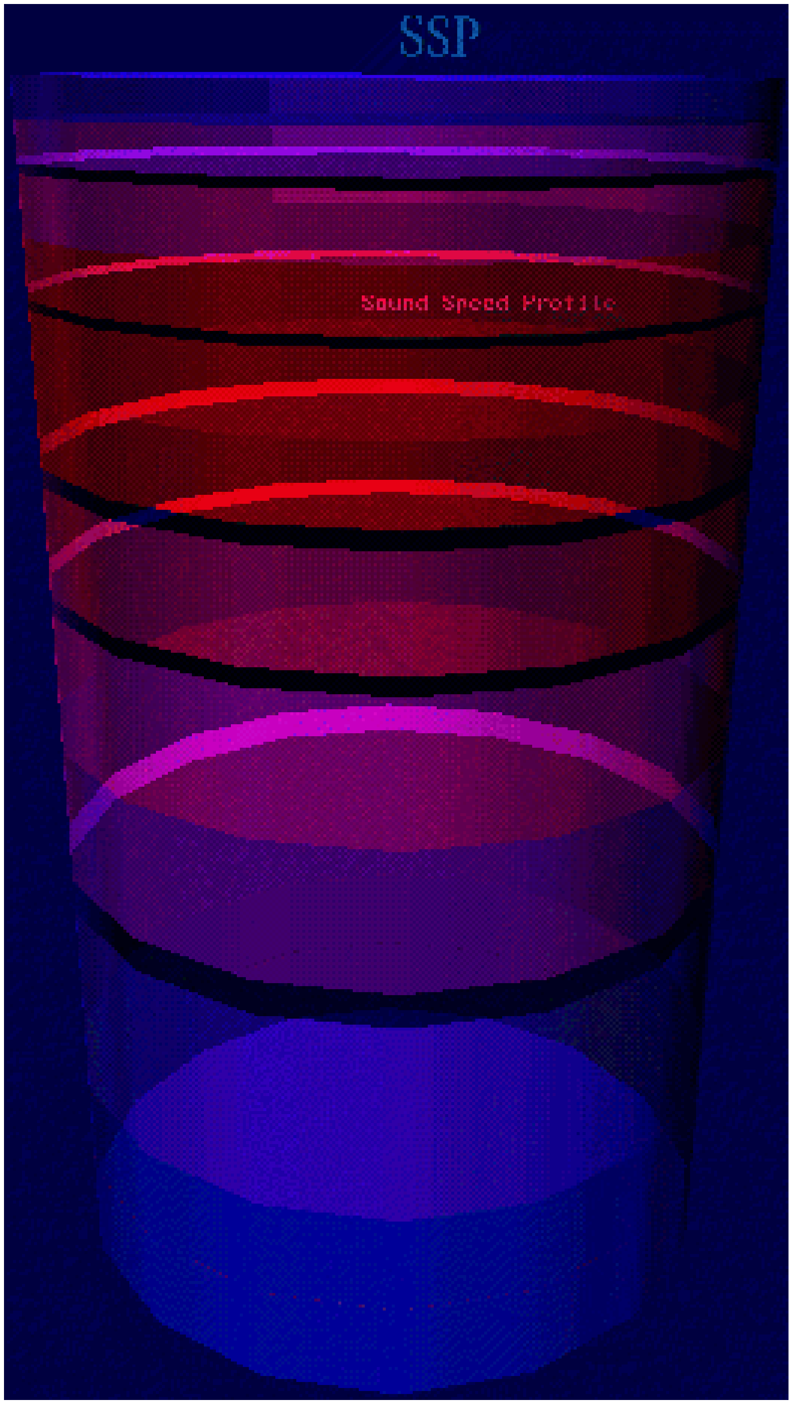

Figure 8.6. Example graphics visualization of subsampled Sound Speed Profile (SSP). Sound speed is mapped to cylinder color at intervals proportional to local depth, producing a 3D information icon.

IX. EXPERIMENTAL RESULTS

A. INTRODUCTION

Experimental testing is essential to producing realistic response in a virtual world. Real world behavior can be predicted and analyzed in the laboratory by running tests which exercise all vehicle systems, and by reproducing mission scenarios which are used in the real world. Since no such thing as a completely benign AUV test environment exists in the real world, laboratory virtual world testing is essential and can overcome impediments associated with the use of tethers and acoustic telemetry. Repeatability of results and statistical control of sensor errors enable tests and machine learning algorithms which are not feasible in the real world. Duplication of at-sea test results in the laboratory can serve as validation of virtual world functionality, at least as is seen from a robot perspective.

An extended laboratory test mission is examined to illustrate how hydrodynamics response is highly complex and requires detailed analysis. Network response is also evaluated for several experiments that utilized the Internet-wide Multicast Backbone (MBone). Network parameters of greatest interest are temporal latency, bandwidth requirements, and suitability for large-scale distributed simulation.

B. PREDICTING AND ANALYZING REAL-WORLD BEHAVIOR IN THE LABORATORY

The key to producing reliable robot software is repeated testing. There are many reasons why risk-free end-to-end testing of all systems aboard an AUV is rarely possible. Leaks can occur in shallow or deep water. Vehicle hydrodynamics response is complex and is also crucial to understanding physical behavior. The many effects involved in underwater motion make accurate posture response an essential prerequisite for meaningful testing of vehicle control algorithms and intelligent control architectures. Underwater vehicles contain too many fragile components to "navigate by collision" as some indoor robots do. Underwater sonar range sensors do not operate in air. Underwater vehicles are usually very heavy, and test stand mountings with multiple degrees of freedom are impractical.

Repeated testing using a tether for remote monitoring and emergency intervention is an effective test technique when preparing for open-ocean autonomous missions (Brancart 94) (Pappas 91). Unfortunately tethers induce significant drag effects, and tether management either requires very expensive tether control systems or continuous human supervision. Acoustic telemetry can free the vehicle from these impediments, but acoustic communications are always prey to intermittent loss due to factors such as multipath arrival, masking, attenuation, and sound wave propagation away from source or receiver. Deployment and recovery of vehicles in the water is always costly and time-consuming, limiting the scope of test programs. In-water results are usually nonrepeatable due to changing conditions or lack of time. This inability to reliably repeat tests on demand greatly complicates software engineering tasks such as debugging, algorithm tuning, and logic verification.

Laboratory testing using a virtual world can produce repeatable results that are based on realistic hydrodynamics response and realistic sonar predictions. Laboratory tests can attempt to replicate in-water results as a means of tuning models to more accurately represent the real world. Since a virtual world includes everything normally detectable by the robot in the real world, a virtual world can be validated by identical robot operation in identical scenarios in either world. In a sense this serves as a kind of Turing test for the virtual world: if robot operation is identical in the real world and the virtual world, then the virtual world is functionally equivalent to the real world. In practice, small differences are usually expected which always need to be fed back into tuning virtual world component models more exactly. Note that the sophistication of this approach will likely lead to more rigorous consideration of interactions among multiple models which is impossible using standalone simulations.

As virtual world component models become more reliable and robust, vehicle deviations from predicted behavior in the real world will be less frequent. A robot can be programmed to recognize and measure such deviations, eventually automating many details regarding model error detection and correction. Embedding virtual world models as predictors in robot control logic will lead to robust failure diagnosis and correction schemes, perhaps coupled with machine learning techniques for greater generality.

A significant advantage of laboratory testing over real world testing is the ability to eliminate or statistically control error deviations in sensor measurements. Usually robot designers need first to test their programs under perfect conditions to demonstrate correctness, and then test again under error-prone conditions to demonstrate robustness. Setting error distributions of sonars or inertial measurement devices permits statistical analysis of arbitrary measures of effectiveness (MOEs) for large numbers of replications. Such testing is useful for determining overall system effectiveness over a range of operating conditions, and also enables machine learning techniques based on massive repetitive training.

Validation and verification of underwater virtual world models for dynamics and sonar needs to be an ongoing part of any AUV research and development program. The complexity and subtlety of these large models means that multiple effects may contribute to a given response, and any change to a hydrodynamic coefficient may ripple through the model with unexpected side effects. A set of standardized vehicle missions and documented responses needs to be duplicated and compared in the virtual world whenever such model changes occur. This process also is a likely candidate for automation as model reliability improves.

Verification Validation and Accreditation (VV & A) is a set of methodologies concerned with showing that simulation models are correct representations of reality. Some key terms follow:

- *Verification*: Substantiation that the computer program implementation of a conceptual model is correct and performs as intended. (Kneppell 93)

- *Validation*: Substantiation that a computer model, within its domain of applicability, possesses a satisfactory range of accuracy consistent with the intended application of the model. (Kneppell 93)
- *Accreditation*: official determination that the model and program are valid enough.

Confidence assessments can be performed to assess the credibility of a simulation. Detailed methodologies have been developed for conducting such assessments (Kneppel 93) (Law, Kelton 91). All aspects of a simulation are evaluated including level of detail, scope of intended use, fidelity, granularity, data verification, constraining assumptions and model validity. Concerns specific to hardware-in-the-loop simulations include timing constraints, information exchange and system integration. While operational tests are considered to be of greatest importance in validation and verification, overall confidence assessments remain a value judgement determined by extensive evaluation of all aspects of a simulation. As with many software engineering practices, formal approaches to verification and validation are of greatest value in ensuring correct design and implementation. Accreditation is expected to be a future issue for models and virtual worlds used by the U.S. Department of Defense (DoD), but current accreditation policies are immature and not applicable to this project. Recommended future work for this and other virtual worlds is performance of a formal independent validation and verification confidence assessment in accordance with (Kneppel 93). Such an assessment might uncover inadvertently-missing virtual world components, and can also help establish a rigorous theoretical definition of the formal requirements needed for globally networked large-scale virtual worlds.

C. SIMULATION RUN ANALYSIS: *mission.script.siggraph*

A great number of execution level mission scripts have been developed to test the many facets of the hydrodynamics model. There is a rich set of execution level script commands available, any of which can be provided via command file *mission.script* or by the user via keyboard. The execution level script command language is also designed to serve as application communications protocol between the real-time execution level and supervising tactical level. Syntax of the script command language appears in (Brutzman 94e). In general these simple commands are similar to those which might be given by a diving officer on a submarine, with the addition of waypoint following and hovering behaviors.

Comprehensive analysis of numerous detailed hydrodynamics variable plots is difficult to perform but remains essential when verifying quantitative robot and model performance. However, intuitive observation and qualitative evaluation of missions is not possible without a 3D real-time graphics viewer. The value of such a viewer cannot be overemphasized. Subtle (and occasionally gross) vehicle events are often not noticeable on the telemetry plots until the user recognition has been cued by the graphics viewer. In most work on hydrodynamics, plots are the only way to formally evaluate performance. Plots still serve an essential function in qualitative analysis, but integration of a live 3D real-time viewer means that users are no longer required to mentally integrate dozens of temporal response curves while attempting to visualize true vehicle behavior.

The most comprehensive robot hydrodynamics test provided in this work is the "SIGGRAPH" mission file (*mission.script.siggraph*), which was used repeatedly during the presentation of the underwater virtual world at the SIGGRAPH 94 conference (Brutzman 94b). The mission script appears in Figure 9.1. A time log of mission output orders appears in Figure 9.2. Twenty plots examining vehicle-environment hydrodynamic interaction follow (Figure 9.3 through Figure 9.22). These plots are automatically produced from robot mission telemetry and can be generated for any robot mission (Brutzman 94e). Essentially these plots show the temporal relationships among three dozen key hydrodynamic variables throughout a mission. A large number of additional test missions focused on specific robot-environment interactions are provided with the underwater virtual world distribution (Brutzman 94e). The SIGGRAPH mission vehicle behavior plots which follow have been manually verified using hydrodynamics coefficients and similar test results produced by earlier NPS AUV hydrodynamics theses (Warner 91) (Bahrke 92) (Torsiello 94).

```
# ,,,,
# Hello SIGGRAPH!
# ,,,,

# This is a mission
script
# for the
# N P S Autonomous
Underwater Vehicle
# ,,,,

# your mission is
# siggraph test

#
mission.script.siggraph

# ,,,,
# we are having some
fun!

# ,,,,
# Graphics rendering
uses Open Inventor
# from Silicon Graphics
,,, S G I
# ,,,,

# D I S multicast
version 2 0 3
# from the
# Naval Postgraduate
School
# ,,,,

# initialize vehiCle

position    0 0 0

orientation 0 0 0

# reset dynamics clock
time        0

thrusters-on

# propellers off
rpm       0

course  000
depth     0

wait      1

# going deep
# with thrusters
depth    45

# here we go!
wait     20

# still going deep
wait     20

# almost there, hang on
wait     25

# at depth
# change course

course  090
wait     10

# stabilizing depth
# during course change
wait     10

# lateral thruster
control

lateral   2
# feet per second
wait      3  seconds

lateral  -2
# feet per second
wait      3

lateral   0
# no more lateral
movement

RUDDER    0

heading  20
wait 5

# propellers on
rpm      400
wait       4

# aim for the window
course   005
rpm      700
wait       9

# building up speed
course   000
wait      11.3

depth     48.2
course   270
wait      26

course   180
wait      10

thrusters-off
# going shallow
# with propellers

depth      4
wait       5

# vehicle hydrodynamics
are
# coupled in six degrees
of freedom
wait       5

# please note that
# vehicle instability
# is very possible
# ,,,,

wait      10

# rudder spiral
rudder   -12
wait      10

# wow this is hard!
# ,,,,
wait      10

# we are spiraling up to
the surface
wait      23

# at the surface
thrusters-on
course   090
wait      10

# moving back above the
pool
wait       8

# we are now stabilizing
rpm     -700
wait      10

# propellers off
rpm        0
wait      10

course     0
hover  0 0 0
wait      81.5

wait       0.1

# A U V is stable at
origin

# time to restart the
mission
# don't forget to add
your e-mail address

# mission complete
quit
```

Figure 9.1. Canonical execution level mission script: *mission.script.siggraph*

```
# NPS AUV file mission.output.orders: commanded propulsion orders versus time
#
#        timestep:  0.10 seconds
#
# time  heading North East Depth    rpm   rpm    stern  stern   vertical     lateral
#                 x     y     z      port  stbd   plane  rudder  thrusters   thrusters
#                                                                bow/stern   bow/stern

   0.0     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
   0.0     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
   1.0     0.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  21.0     0.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  41.0     0.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  66.0    90.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  76.0    90.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  86.0    90.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  89.0    90.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  92.0    20.0   0.0   0.0  45.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
  97.0    20.0   0.0   0.0  45.0  400.0  400.0    0.0    0.0    0.0   0.0   0.0   0.0
 101.0     5.0   0.0   0.0  45.0  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 110.0     0.0   0.0   0.0  45.0  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 121.3   270.0   0.0   0.0  48.2  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 147.4   180.0   0.0   0.0  48.2  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 157.4   180.0   0.0   0.0   4.0  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 162.4   180.0   0.0   0.0   4.0  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 167.4   180.0   0.0   0.0   4.0  700.0  700.0    0.0    0.0    0.0   0.0   0.0   0.0
 177.4   180.0   0.0   0.0   4.0  700.0  700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 187.4   180.0   0.0   0.0   4.0  700.0  700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 197.4   180.0   0.0   0.0   4.0  700.0  700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 220.4    90.0   0.0   0.0   4.0  700.0  700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 230.4    90.0   0.0   0.0   4.0  700.0  700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 238.4    90.0   0.0   0.0   4.0 -700.0 -700.0  -12.0    0.0    0.0   0.0   0.0   0.0
 248.4    90.0   0.0   0.0   4.0    0.0    0.0  -12.0    0.0    0.0   0.0   0.0   0.0
 258.4     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
 258.4     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
 339.9     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
 340.1     0.0   0.0   0.0   0.0    0.0    0.0    0.0    0.0    0.0   0.0   0.0   0.0
```

Figure 9.2. Resulting time log of robot mission output orders: *mission.output.orders*

An event-by-event analysis of the SIGGRAPH mission follows in Table 9.1 to identify key relationships and results. This analytical timeline was produced by examining the original mission script Figure 9.1, the condensed mission orders Figure 9.2 and individual hydrodynamics plots (Figure 9.3 through Figure 9.22). Finally graphics images showing vehicle thrusters, propellers and plane surfaces operating simultaneously appear in Figure 9.23 and Figure 9.24. In these figures, green wireframe cones are proportional to the thrust of sea water from the cross-body thrusters and the propellers.

Table 9.1. Timeline Analysis of SIGGRAPH Mission.

time mm:ss	time sec	Analytic results, ordered robot changes and pertinent plots.
0:00	0.0	Initial posture at origin (0, 0, 0) with orientation (0, 0, 0).
0:01	1.0	Behavior stable and unchanging prior to first order. Change depth to 45 ft using vertical thrusters only. Plots 4, 20, 10, 13, 20.
1:16	66.0	At ordered depth, heave rate stabilizing. Change ordered course to right from 000° to 090° using lateral thrusters. Plots 8, 9, 16, 20. Note slight downward pitch angle θ due to coupling with vertical heave velocity rate $\dot{w}$ via coefficient $M_{\dot{w}}$. Plots 6, 7, 15.
1:26	86.0	Reached ordered course 090°. Begin lateral motion to right using lateral thrusters. Plots 2, 12, 16, 20.
1:29	89.0	Reverse lateral thrust to cancel sway velocity $\boldsymbol{v}$. Plots 2, 12, 16, 20.
1:32	92.0	Lateral sway velocity $\boldsymbol{v}$ reduced. Change course to left to 020° using thrusters. Plots 8, 9, 16, 20.
1:37	97.0	Still changing course. Turn on propellers to 400 rpm, enabling rudders and planes. Plots 1, 8, 9, 11, 16, 17, 18, 19, 20.
1:41	101.0	Continue changing course left to 005° for test tank window exit. Forward surge velocity $\boldsymbol{u}$ starting to increase. Plots 1, 8, 9, 11, 16, 17, 18, 19, 20.
1:50	110.0	Continue changing course left to 000° for test tank window exit.
2.01	121.3	Building up speed, ready to maneuver to enter torpedo tube. Come left to course 270°, go down to depth 48.2 ft using propellers, planes and thrusters simultaneously combined. Corresponding graphics images appear in Figure 9.23 and Figure 9.24. Plots 1 through 20.

time mm:ss	time sec	Analytic results, ordered robot changes and pertinent plots.
2:27	147.4	Torpedo tube transit completed via dead reckoning. Change course to left to 180°. Plots 1, 2, 3, 5, 8, 9, 11, 12, 14, 15, 16, 17, 20.
2:37	157.4	Steady on course, forward surge velocity ***u*** starting to increase. Secure thrusters. Begin going shallow to ordered depth 4 ft using forward momentum, plane surfaces and propellers. Plots 1, 4, 6, 7, 10, 13, 15, 18.
2:57	177.4	While continuing to shallow, begin a spiral to right. Ordered rudder held at -12°. AUV geographic position stays within ordered turning radius during extended depth transient. Changes and interactions are again visible among all hydrodynamics variables. Plots 1 through 20.
3:40	220.4	Near ordered depth just below the surface. Turn on thrusters, resume closed-loop rudder control by ordering course 090°. Head back to origin above the pool. Vehicle roll ***p*** from spiral turn restabilizes with forward motion. Plots 1, 2, 3, 5, 6, 8, 9, 10, 11, 12, 14, 16, 17, 20.
3:58	238.4	Nearing origin, slow and stabilize. Reverse propellers to -400 rpm, reducing forward velocity ***u***. Plots 1, 2, 3, 11, 19.
4:08	248.4	Forward velocity ***u*** almost zero, near origin. Zero propellers, coast, slow due to drag and stabilize. Plots 1, 2, 3, 11, 19.
4:18	258.4	Approximately at origin with small velocities remaining. Change ordered course to 000° and shift to hover mode. New ordered position for hovering is origin, depth 0 ft. Propellers now follow forward/aft position error, all plane surfaces zeroed, vertical thrusters control depth, and lateral thrusters track port/starboard position error. Plots 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15, 16, 19, 20.
5:50	350.0	Hovering has fully stabilized AUV at origin with zero posture. Mission complete. Plots 1 through 20 stable.

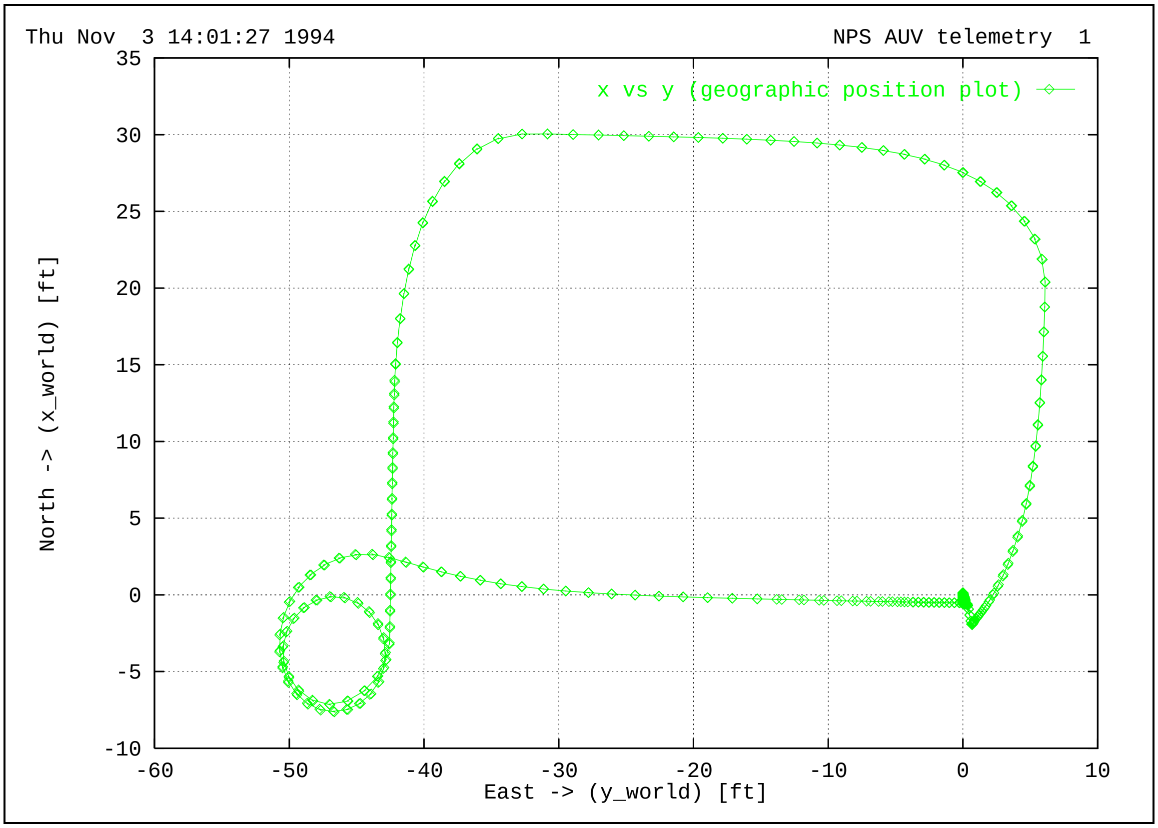

Figure 9.3. Geographic plot (world ***x*** and ***y*** coordinates) of AUV position track.

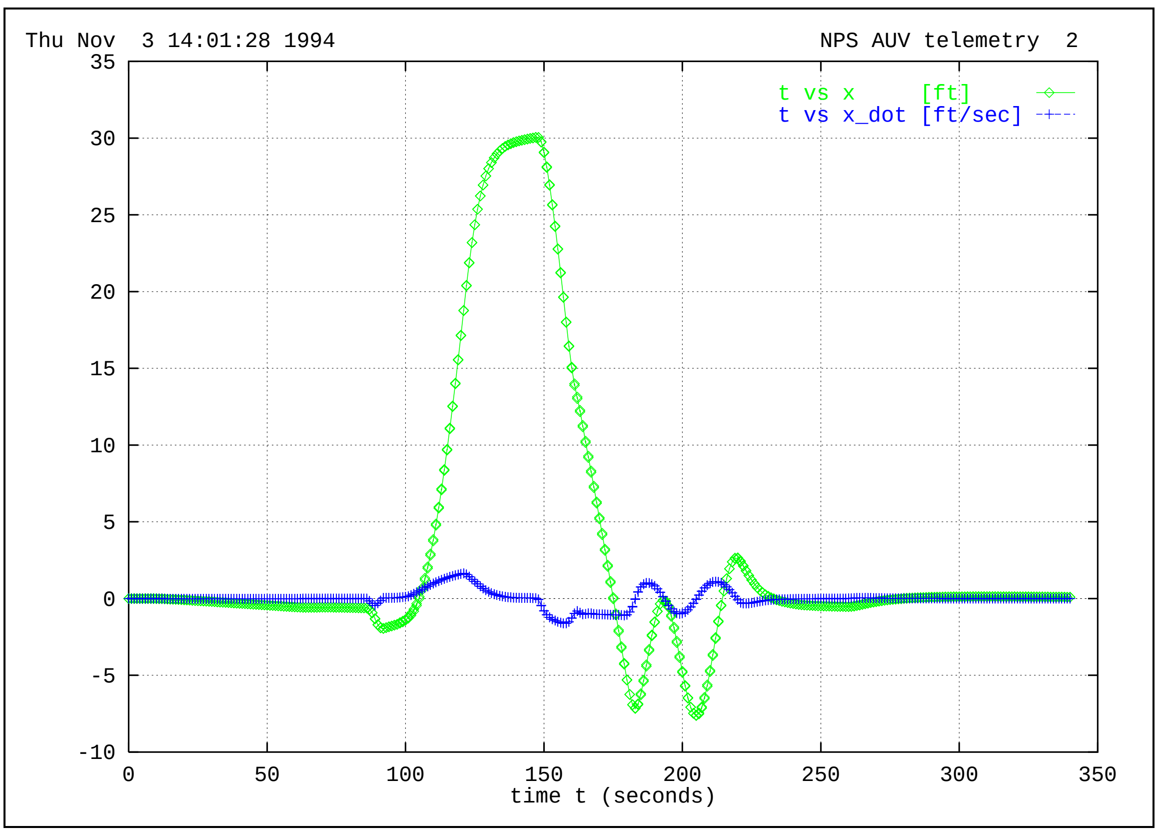

Figure 9.4. World position coordinate ***x*** and derivative versus time ***t***.

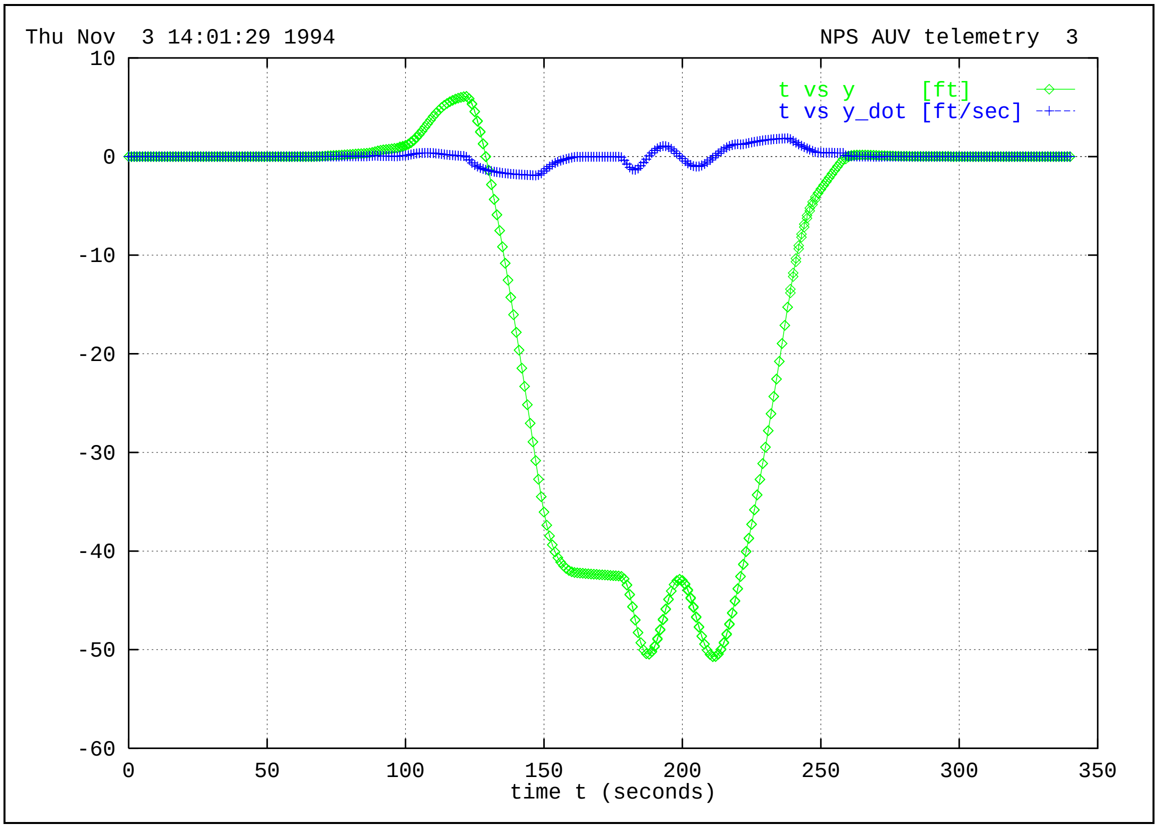

Figure 9.5. World position coordinate ***y*** and derivative versus time ***t***.

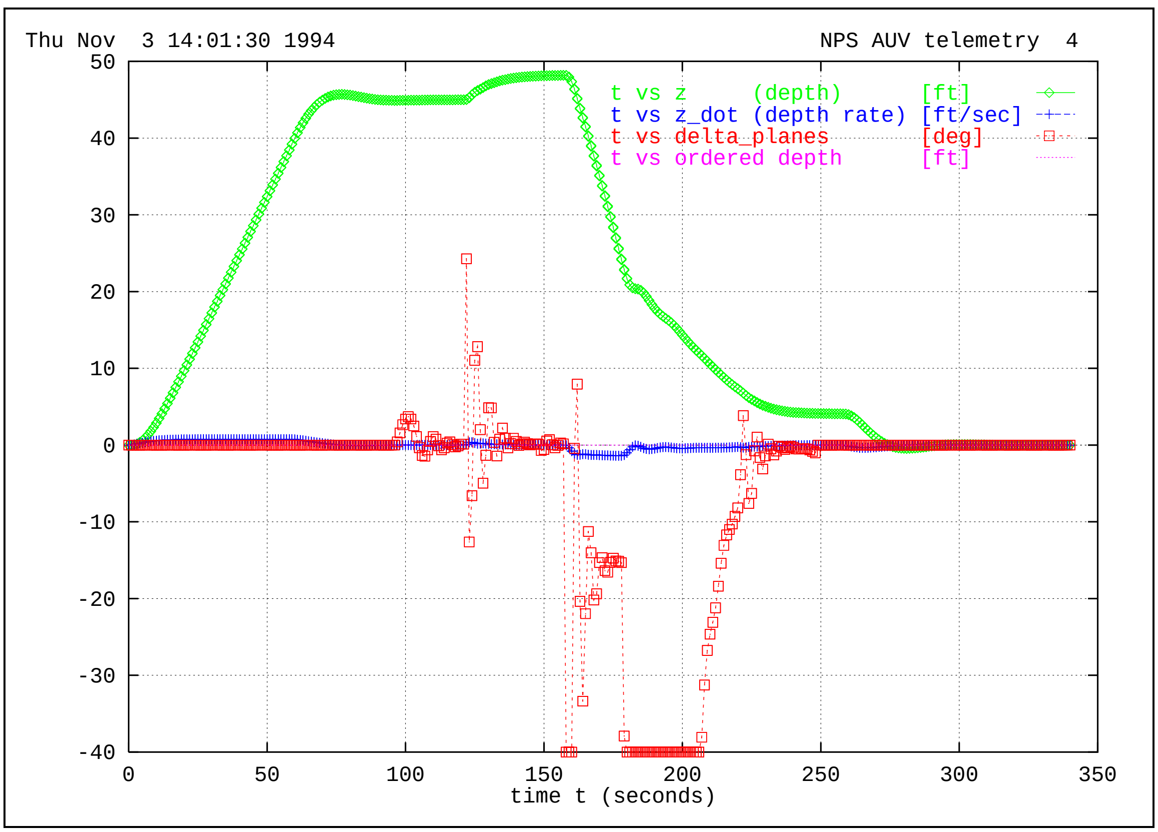

Figure 9.6. World depth coordinate ***z*** and derivative versus time ***t***.

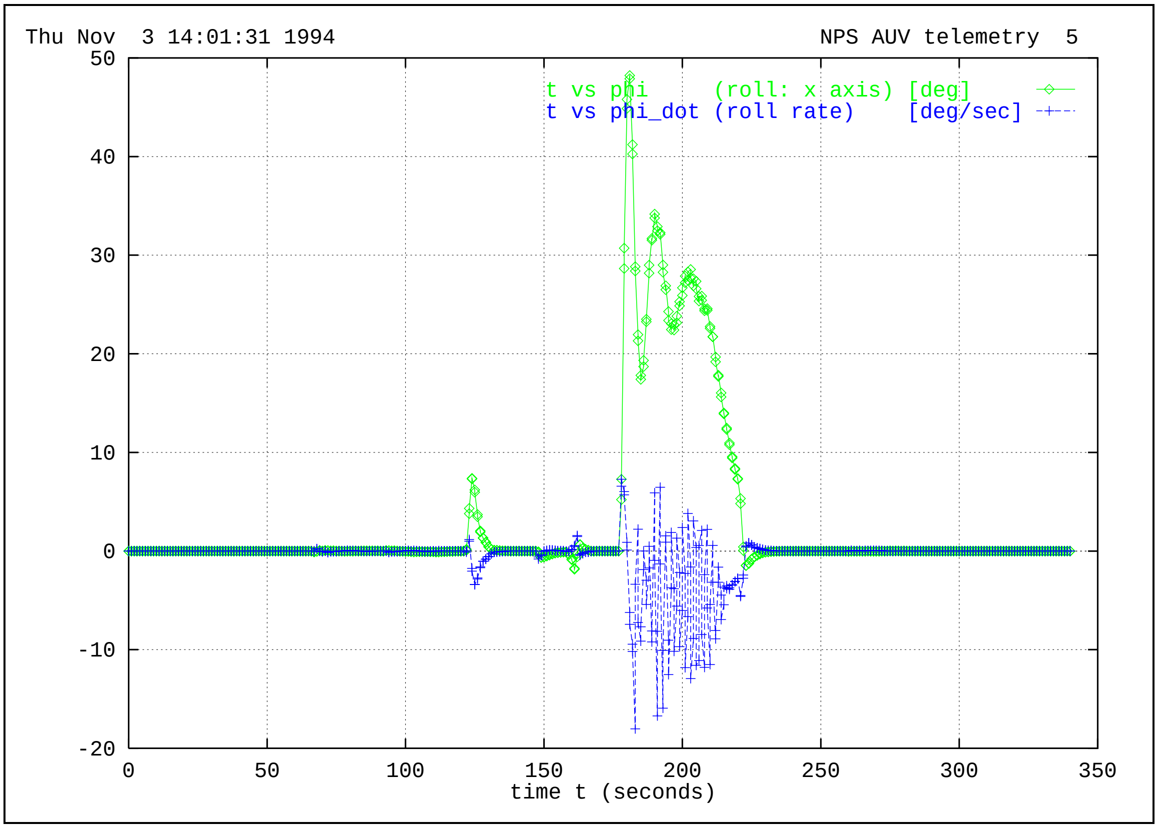

Figure 9.7. World roll Euler angle ϕ and derivative versus time ***t***.

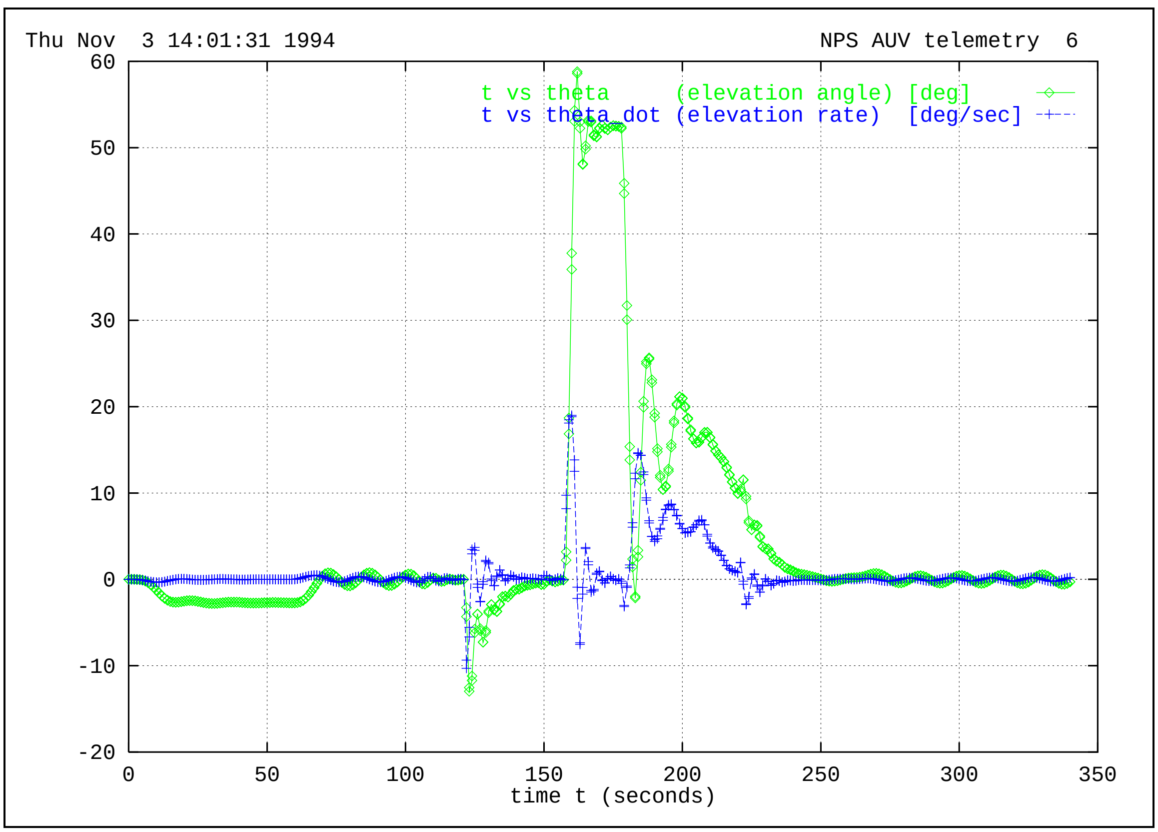

Figure 9.8. World pitch Euler angle θ and derivative versus time ***t***.

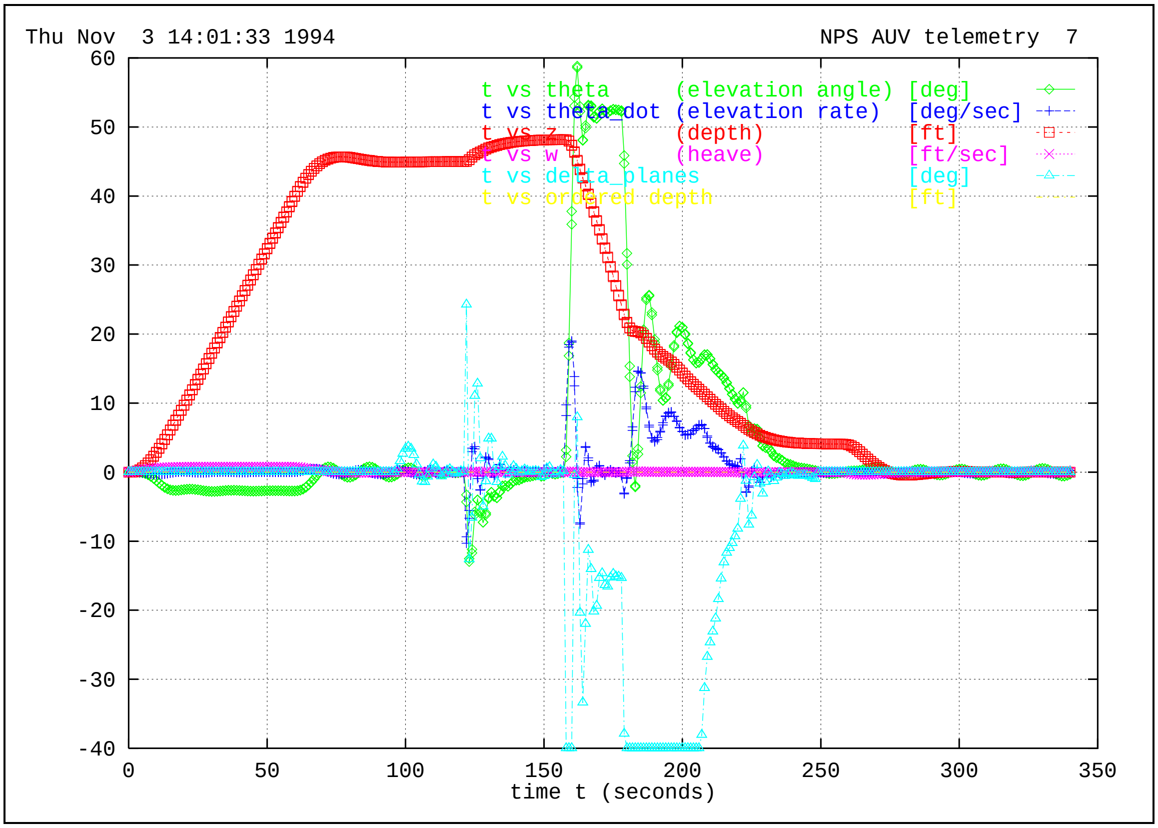

Figure 9.9. World theta Euler angle θ and related variables versus time ***t***.

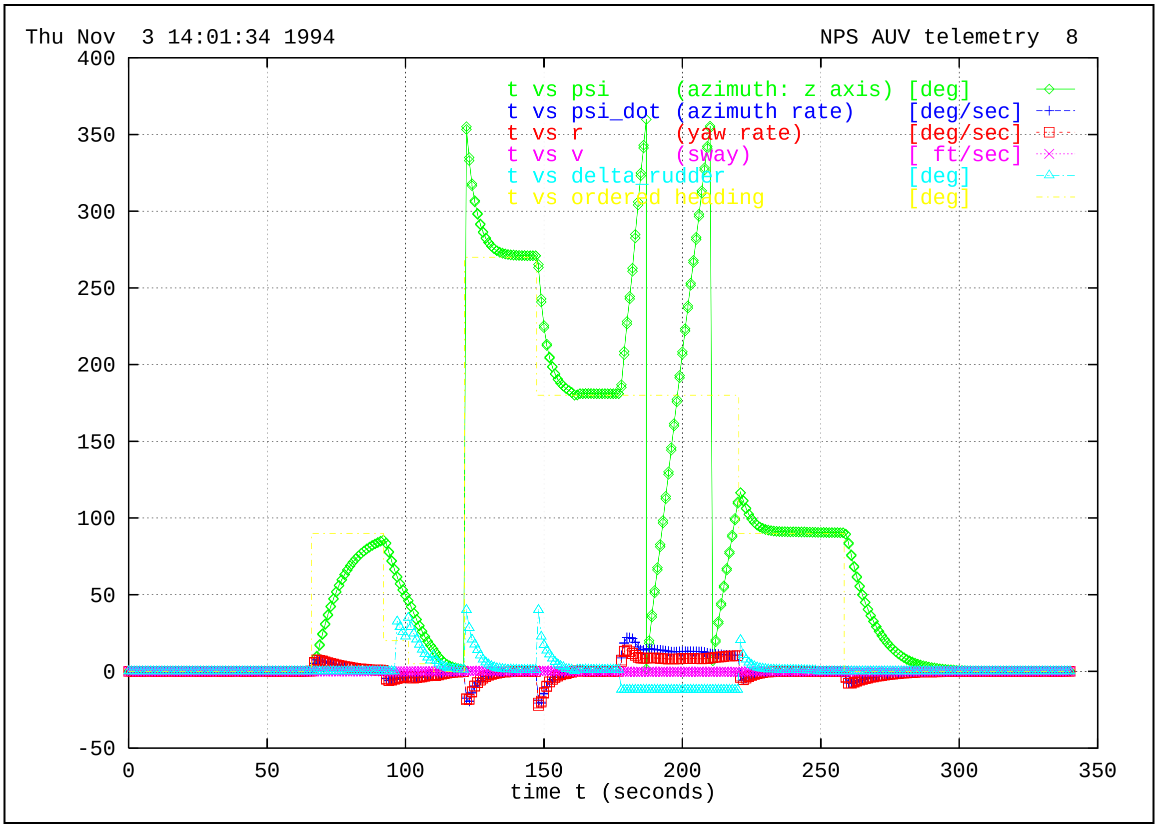

Figure 9.10. World yaw Euler angle ψ and derivative versus time ***t***.

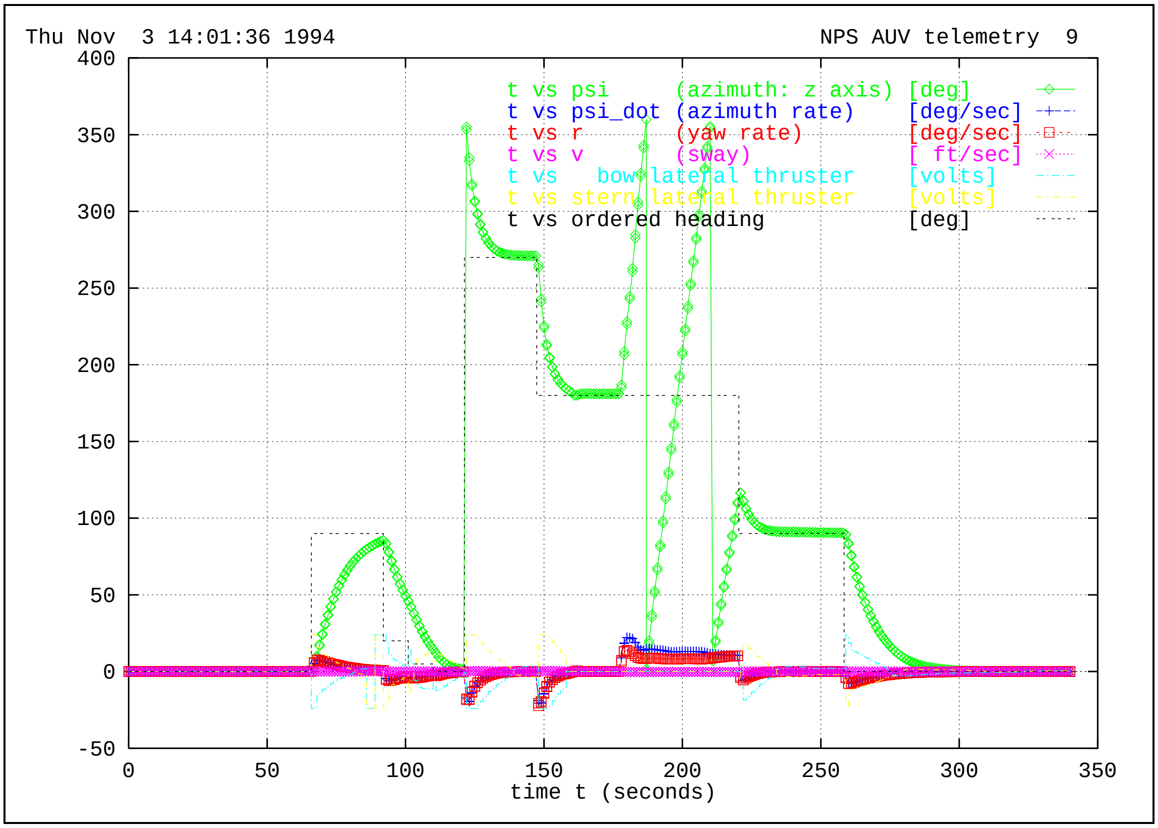

Figure 9.11. World yaw Euler angle ψ and lateral thrusters versus time ***t***.

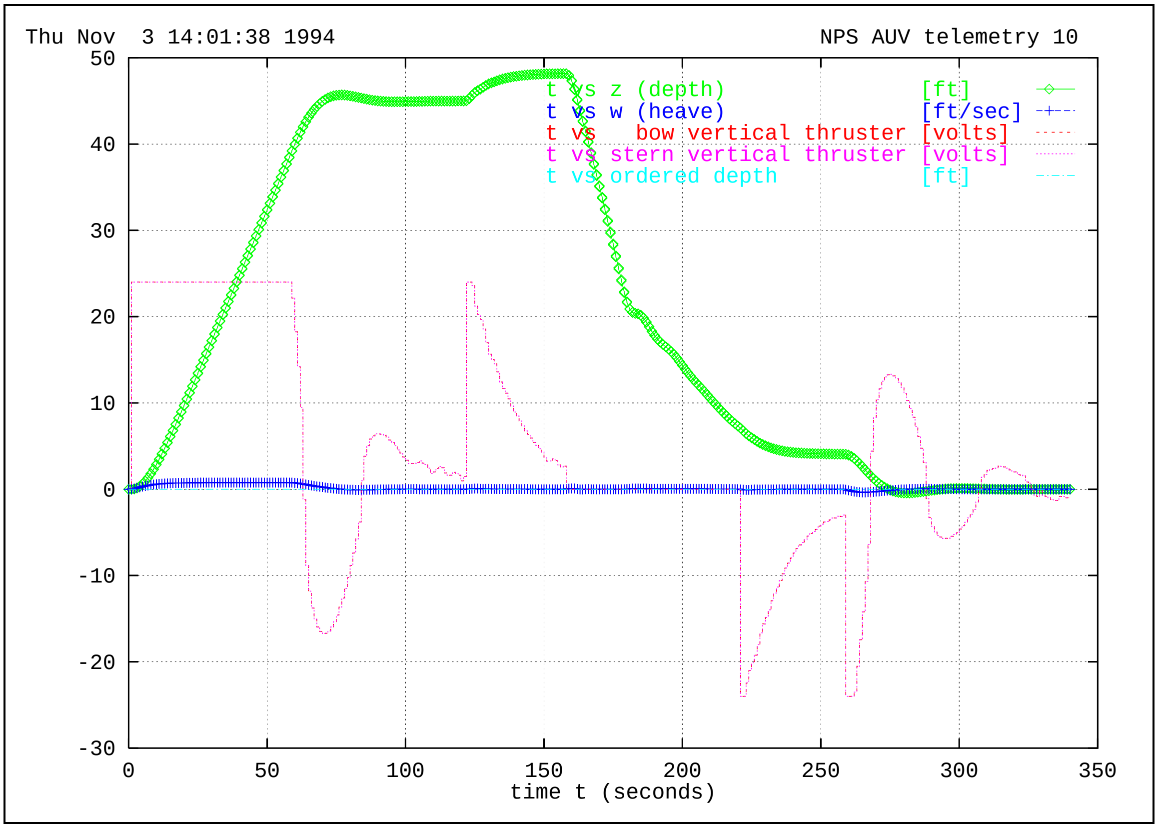

Figure 9.12. World depth coordinate ***z*** and related variables versus time ***t***.

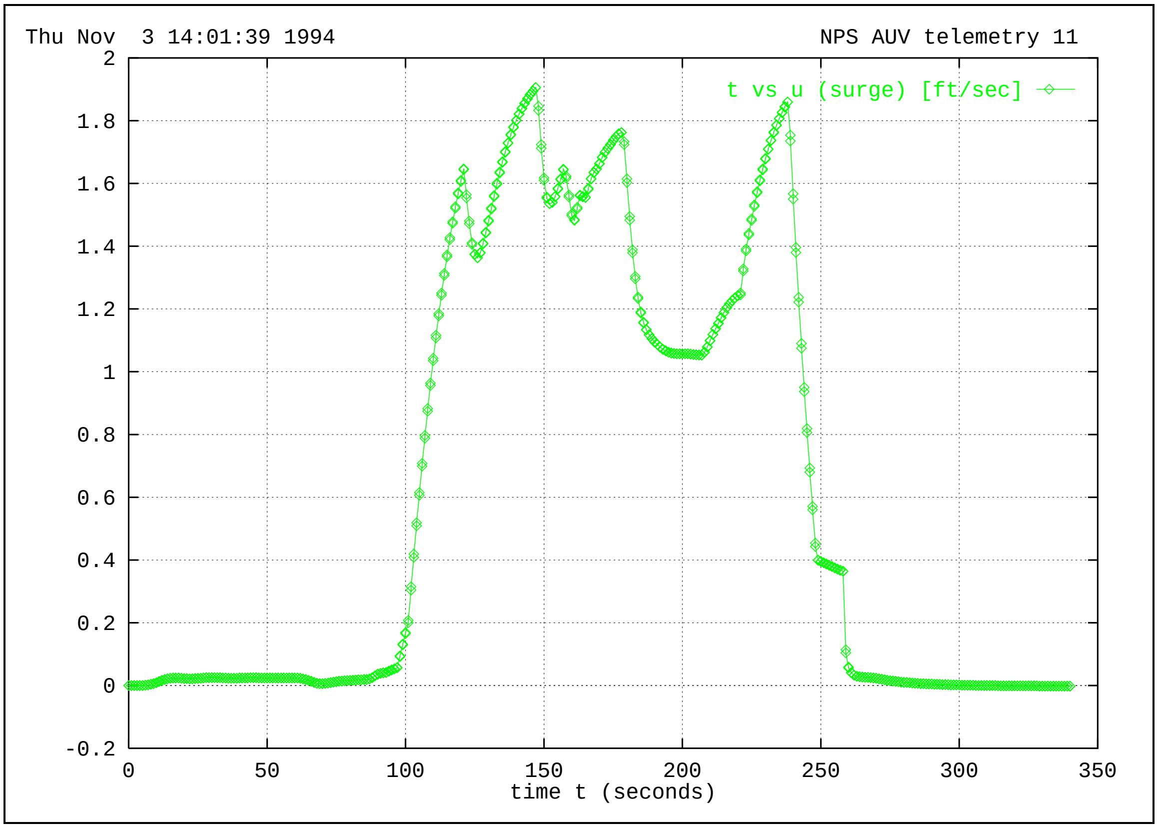

Figure 9.13. Body longitudinal surge velocity ***u*** versus time ***t***.

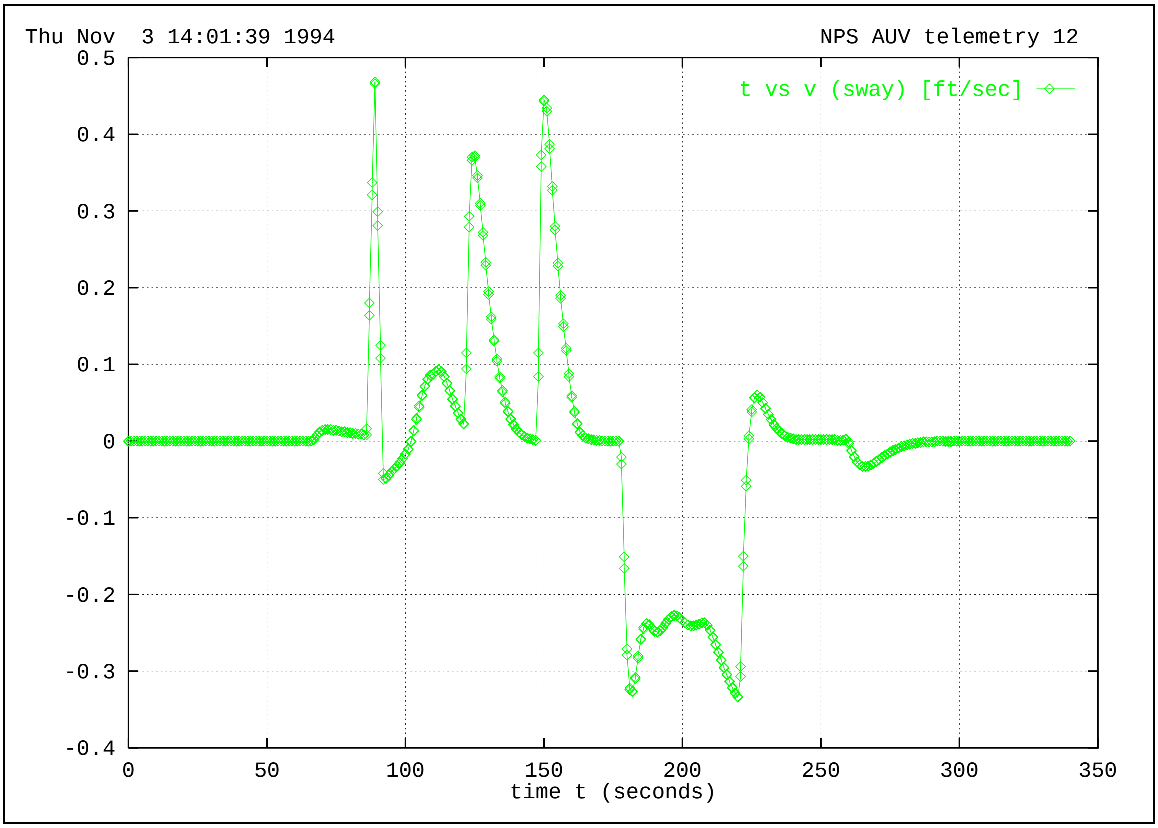

Figure 9.14. Body lateral sway velocity ***v*** versus time ***t***.

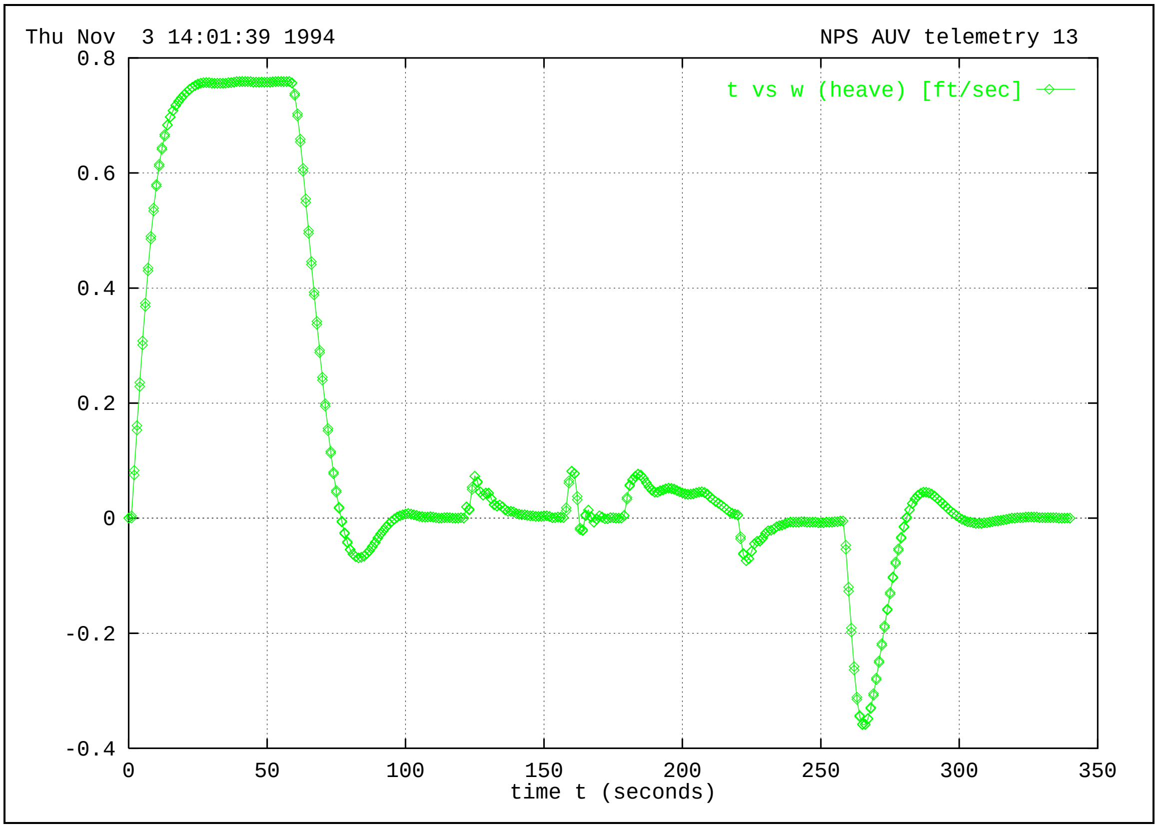

Figure 9.15. Body vertical heave velocity ***w*** versus time ***t***.

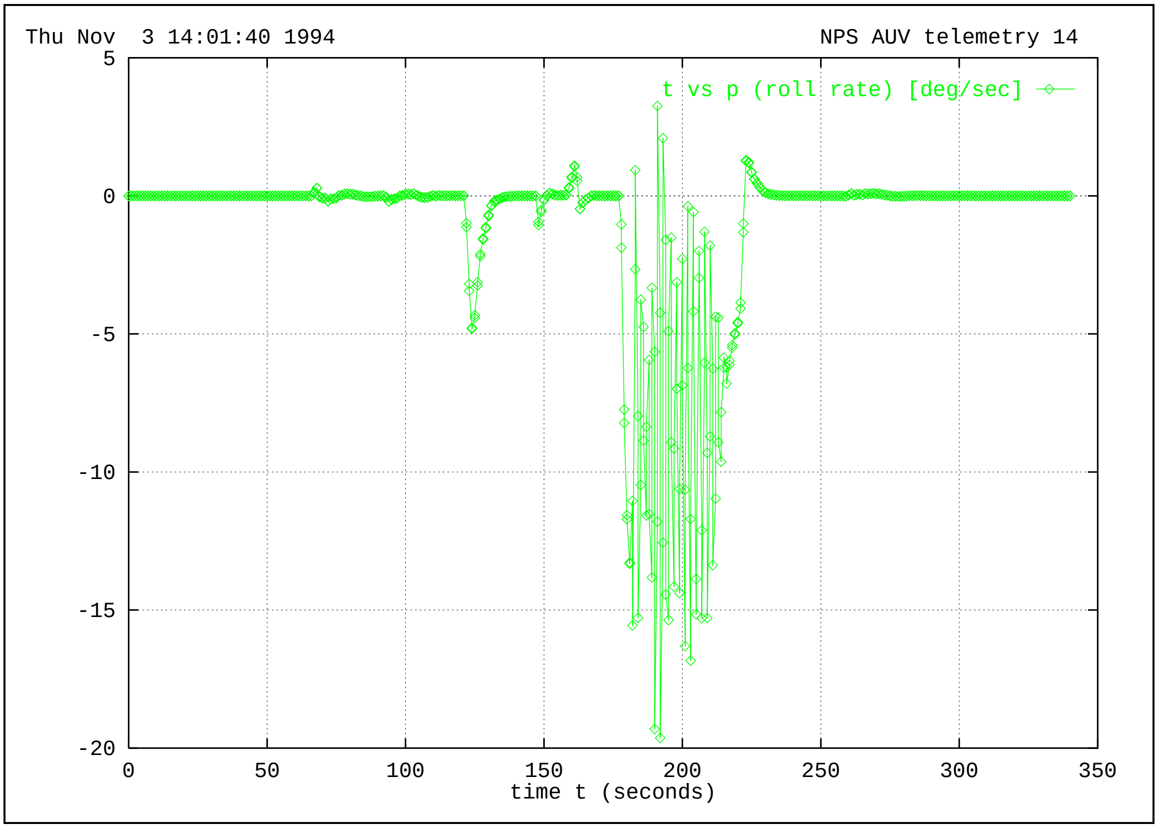

Figure 9.16. Body longitudinal rotation roll rate ***p*** versus time ***t***.

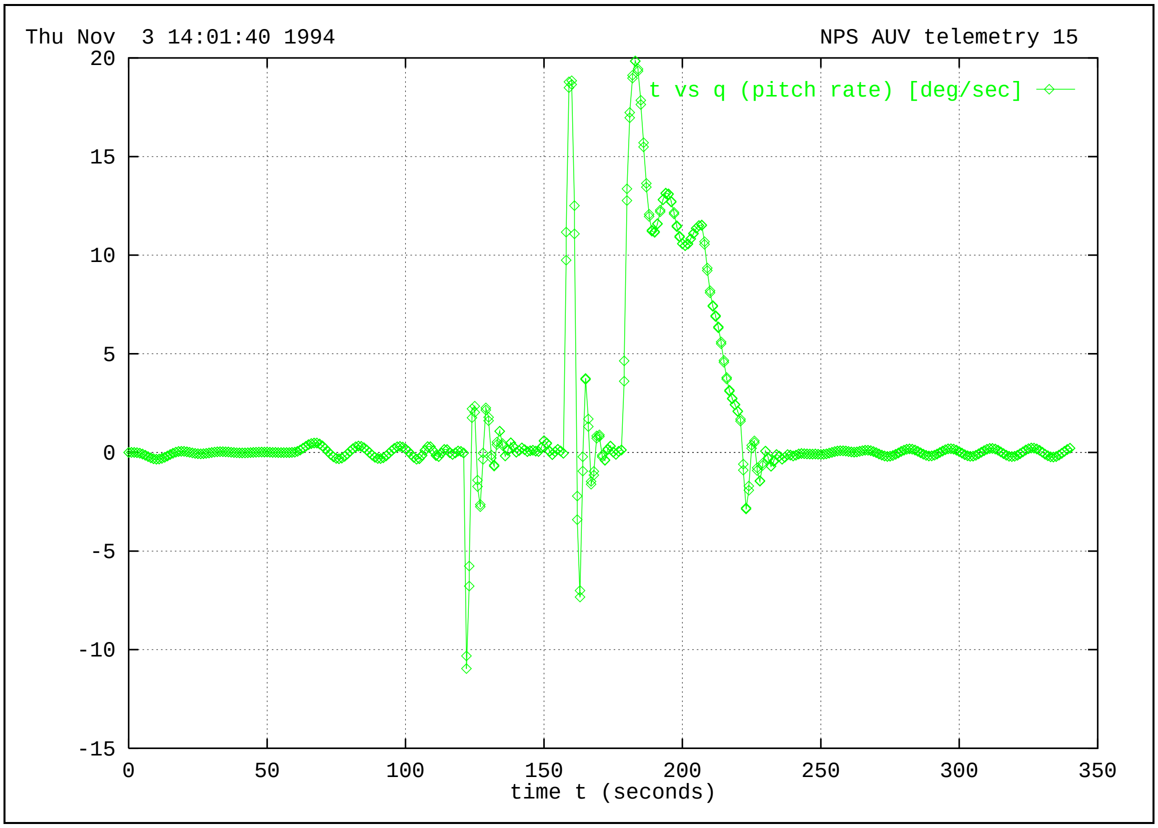

Figure 9.17. Body rotational pitch rate ***q*** versus time ***t***.

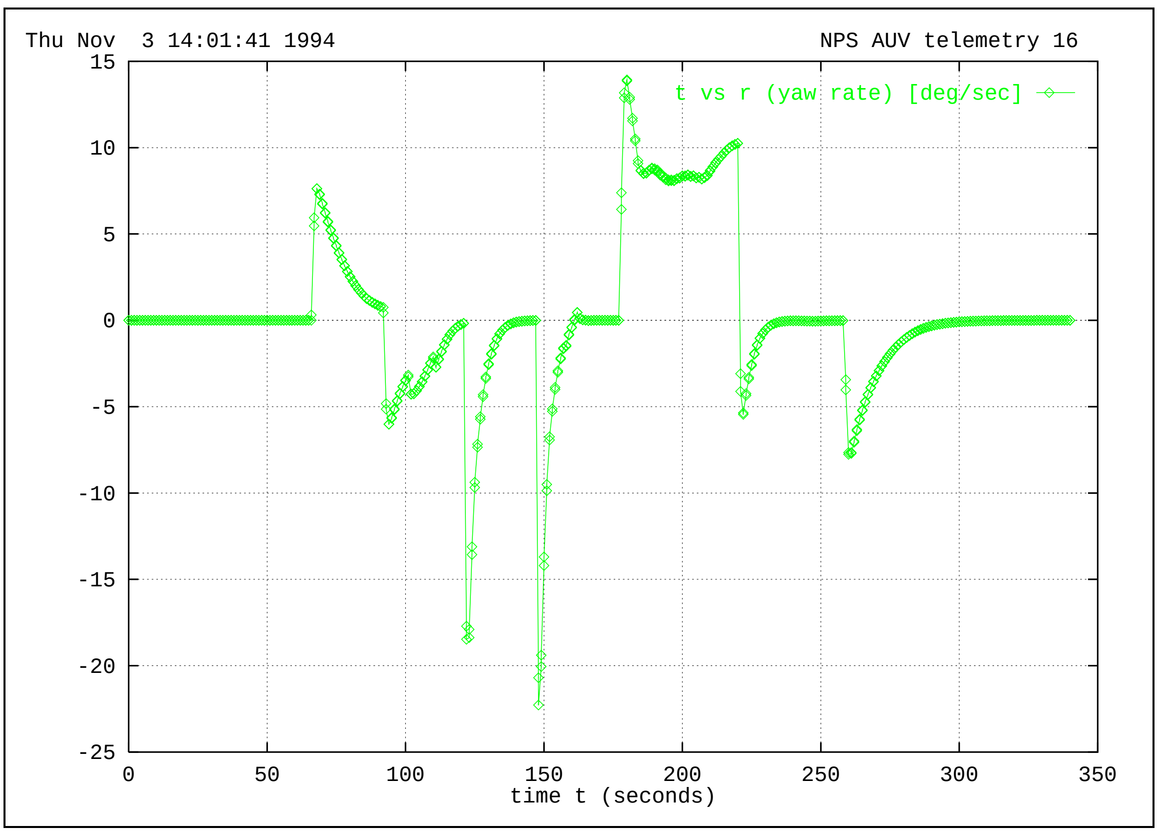

Figure 9.18. Body vertical rotation yaw rate ***r*** versus time ***t***.

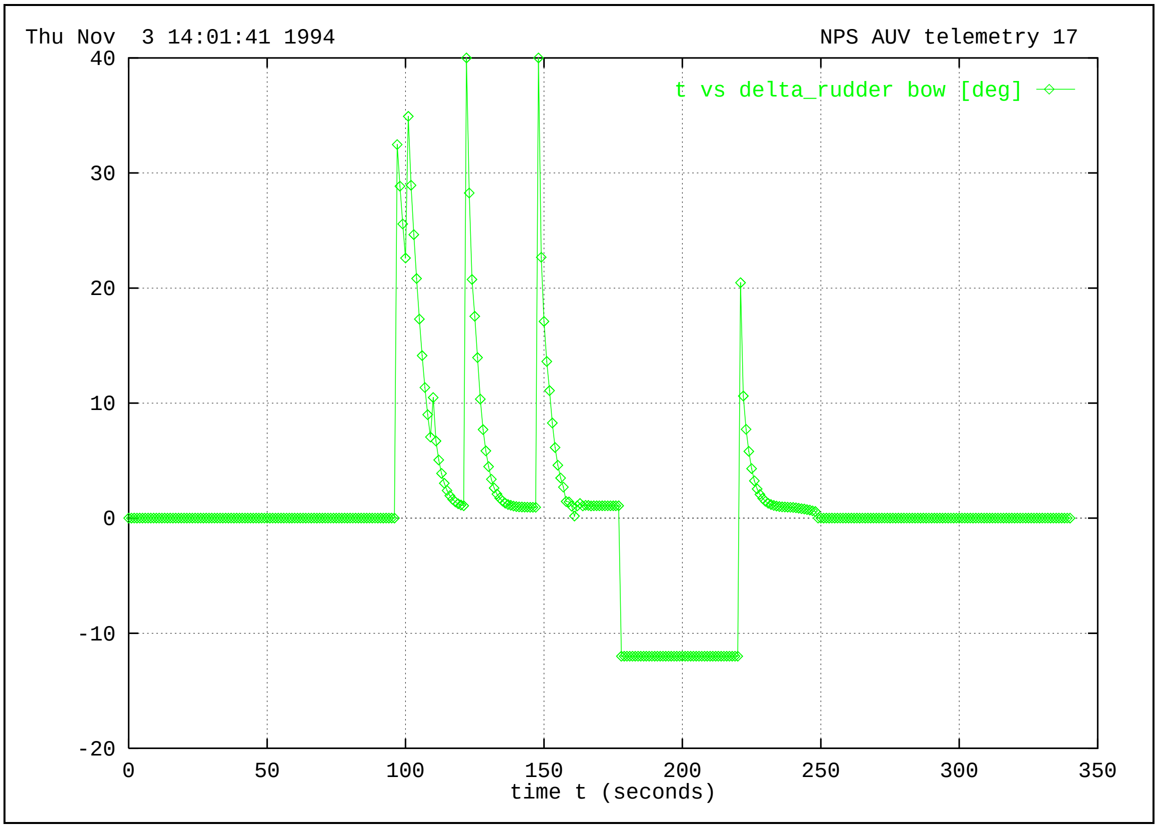

Figure 9.19. AUV bow rudders rotation (stern rudders opposed) versus time ***t***.

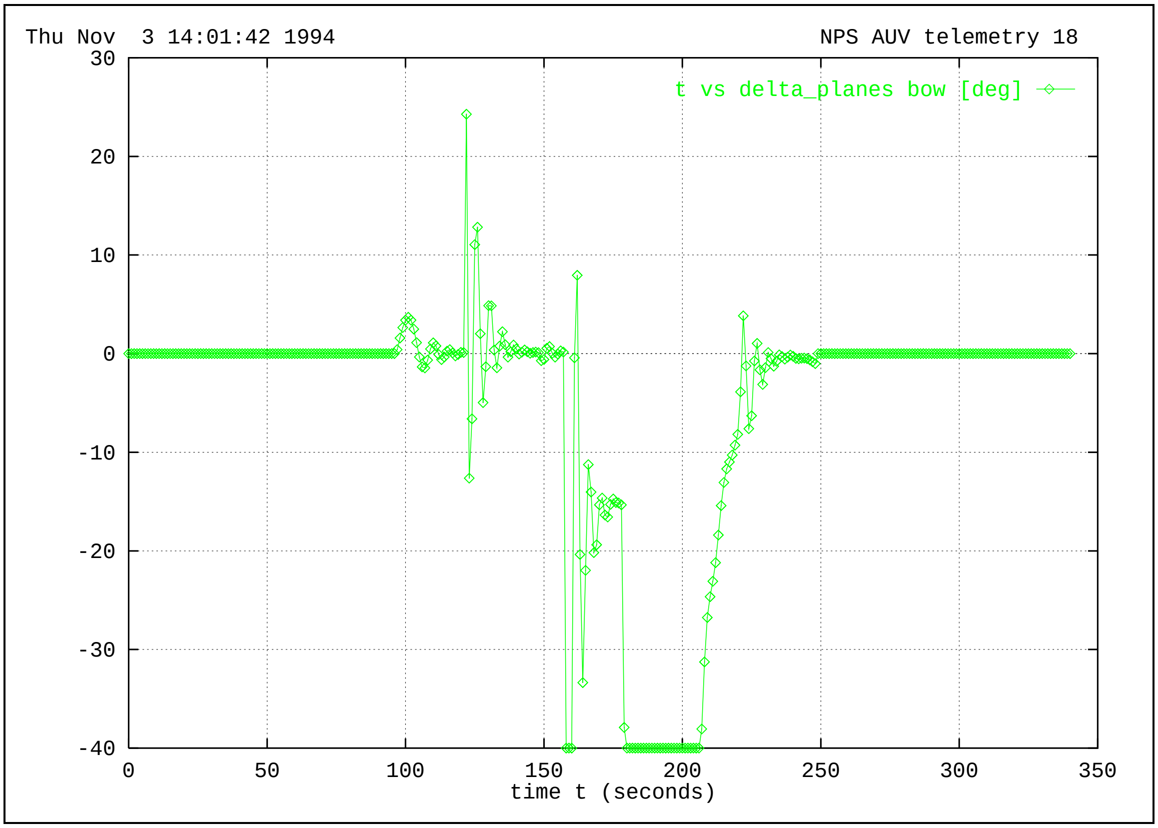

Figure 9.20. AUV bow planes rotation (stern planes opposed) versus time ***t***.

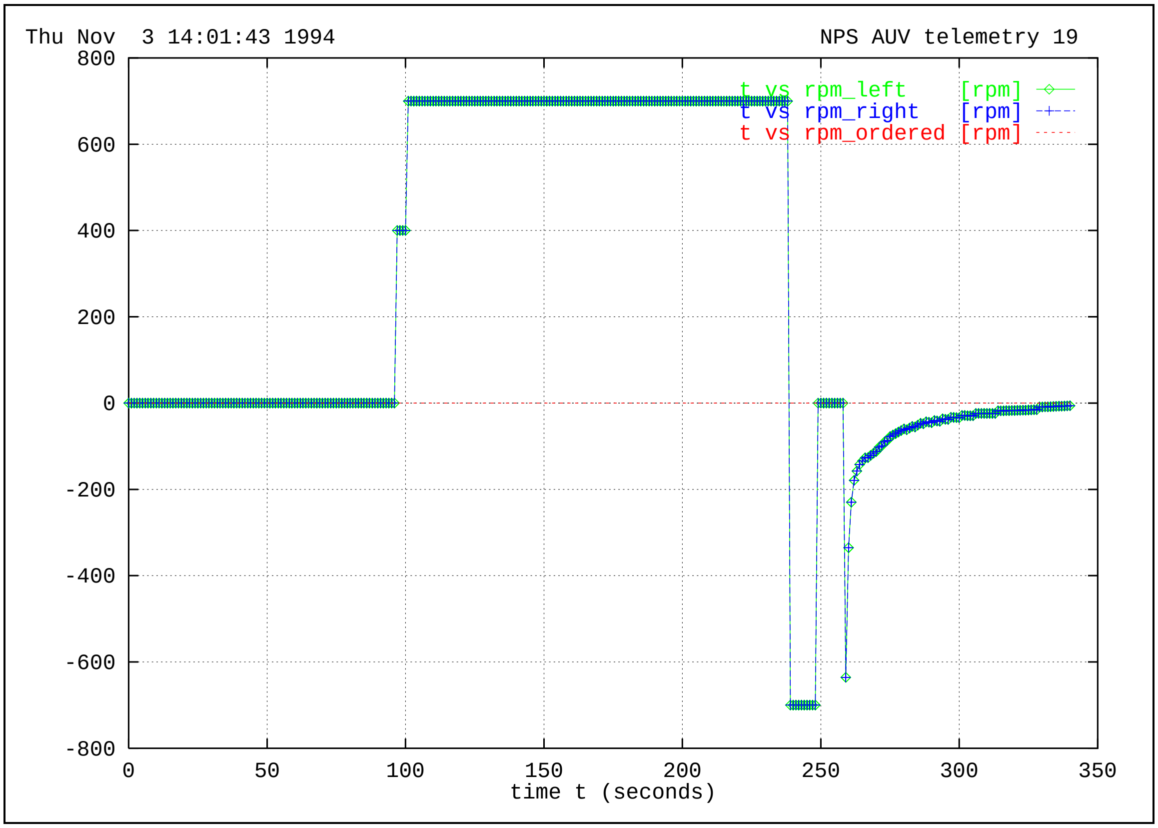

Figure 9.21. AUV port and starboard propeller speed versus time ***t***.

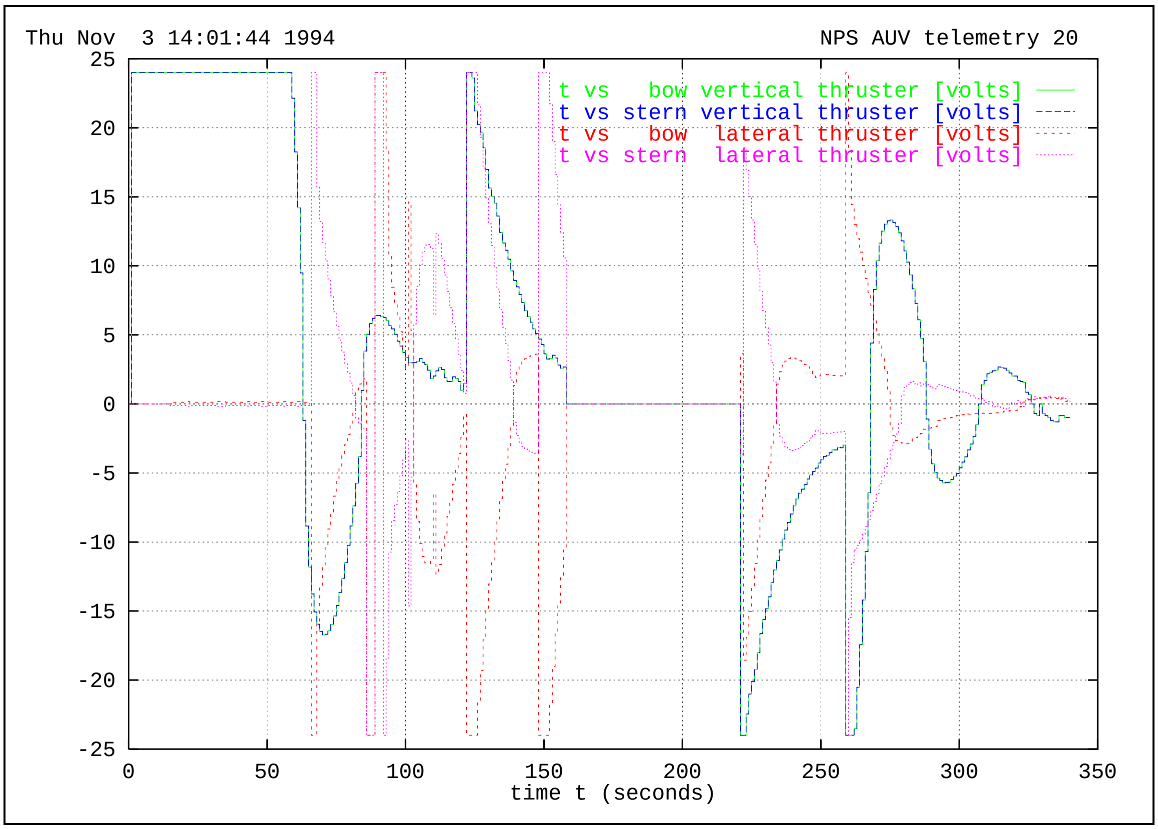

Figure 9.22. AUV vertical and lateral thruster control voltages versus time ***t***.

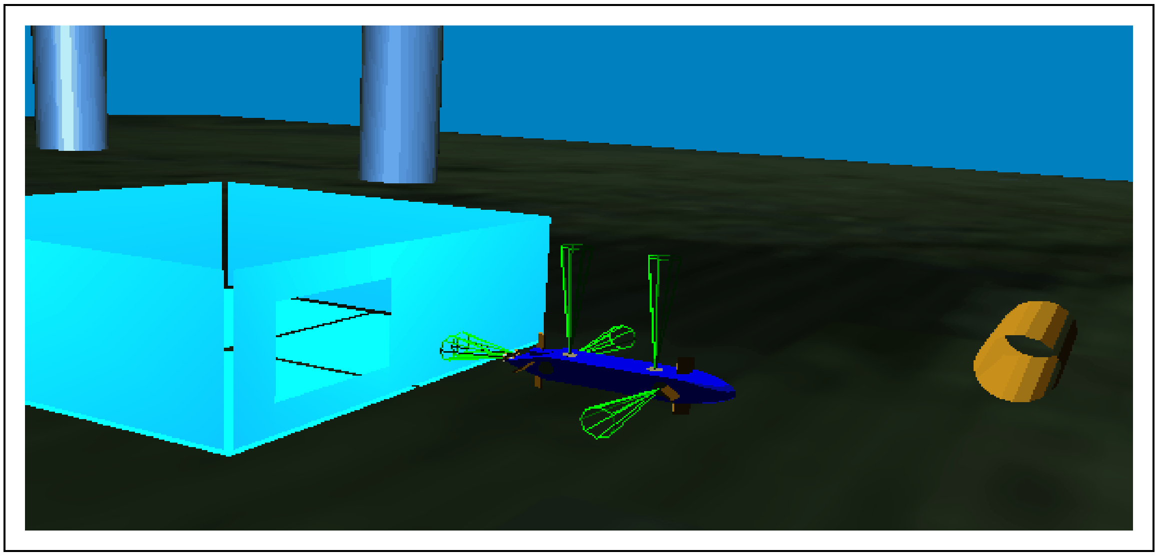

Figure 9.23. AUV initial turn and depth change from the test tank to the torpedo tube using all thrusters, propellers and planes.

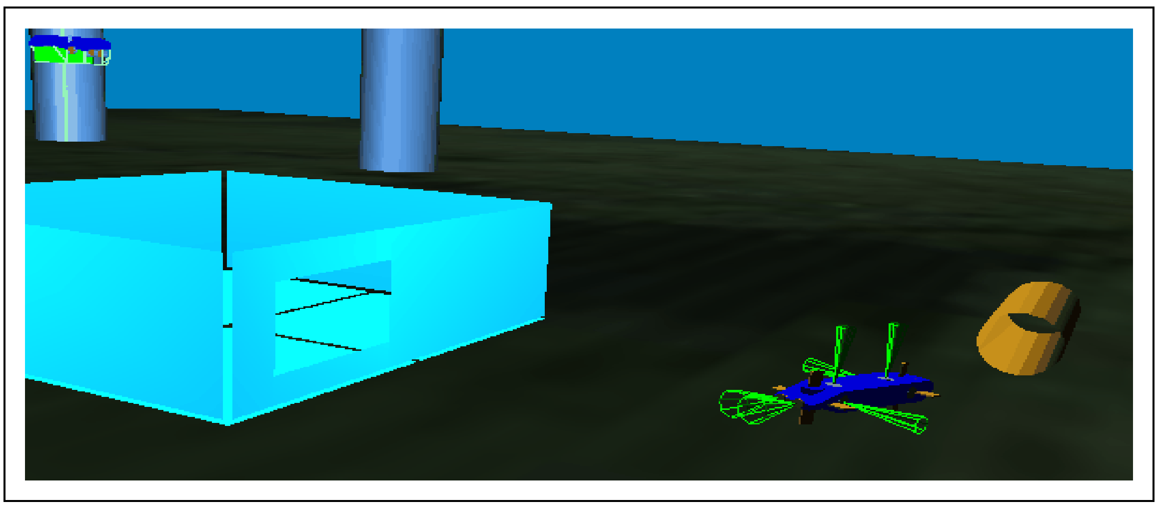

Figure 9.24. AUV nearing entry to torpedo tube. Note counterintuitive (but correct) opposition of lateral thrusters to propeller and rudder control, damping yaw rate ***r*** and preventing overshoot.

D. NETWORK TESTING AT *The Edge*

Distribution of underwater virtual world components enables scalability and real-time response, both for robot world models and for people. The distributed implementation of the underwater virtual world (Brutzman 94e) was tested and demonstrated for six days as part of *The Edge* exhibition at the SIGGRAPH 94 conference, held in Orlando Florida (Brutzman 94b, 94c). We estimate that 2,000 of 32,000 SIGGRAPH attendees stopped at our underwater virtual world exhibit to observe a robot mission and learn about the project. Robot interactions in the virtual world were also multicast over the MBone with worldwide scope (ttl 127) using audio, video and DIS channels.

The forty reviewed exhibits in *The Edge* were representative of leading computer graphics applications in the world. *The Edge* was intended to include shared experiences, simulation, training, education, virtual environments, high-bandwidth networked graphics, telepresence and telerobotics. The underwater virtual world project has components and relevance in each of those areas. Our objective was to inspire and stimulate attendees to consider a myriad of opportunities previously considered infeasible. Feedback comments from visitors, SIGGRAPH organizers and the press (Meyer 94) were uniformly enthusiastic.

One technical goal during this demonstration was to evaluate netwoRk loading. Bandwidth budget plans called for an average bandwidth of 225 kilobits per second (Kbps) is available (i.e. 25% of a 1.5 Mbps T1 Internet connection). This bandwidth budget included 128 Kbps for locally generated video/graphics, 64 Kbps for a shared audio channel and 15 Kbps for sending DIS PDUs. 128 Kbps is the default bandwidth for world-wide multicast video programs and equates to 1-3 frames per second. Lower or higher bandwidths and a corresponding change in frame rate are feasible.

DIS Entity State PDU size for the NPS AUV is larger than the nominal DIS PDU default, since three articulated parameters are attached to each AUV PDU for sonar, plane surface, propeller and thruster values. DIS protocol bandwidth for the

SIGGRAPH demonstrations were based on these PDUs being multicast at full virtual world update frequency of 10 Hz.

$$\begin{aligned} DIS\ \ bandwidth &= (multicast\ PDU\ update\ rate)\ [PDU\ size] \\ &= (10\text{ Hz }\ update\ rate) \left[\frac{1152\text{ bits}}{PDU} + 3\left(\frac{128\text{ bits}}{\begin{matrix} articulated \\ parameter \end{matrix}} \right) \right] \qquad \textbf{(9.1)} \\ &= 15.36\text{ Kbps} \end{aligned}$$

This full update rate of 10 Hz was used to relay every possible nuance related to physical motion of the highly dynamic autonomous underwater vehicle. By way of contrast, a standard Entity State PDU with no articulated parameters being relayed at the maximum standard interval of 5 seconds produces only a 0.23 Kbps bandwidth load.

Another important way of making virtual worlds widely available is developing an information infrastructure where potential virtual world participants have the network capabilities to participate. Toward this end we have utilized MBone in a number of scholarly conferences. Objectives are usually twofold: learn how to use global videoconferencing more effectively, and assist potential collaborators in learning more and connecting. We have achieved a steady series of successes at a variety of sites including 1993 U.S. Geological Survey Menlo Park scientific visualization workshop, the International Advanced Robotics Programme (IARP) Mobile Robotics for Subsea Environments 94 (Brutzman 94a), IEEE Autonomous Underwater Vehicles 94 (Brutzman 94d), GLOSAS Global Lecture Hall of July 94 (McLeod 94), and SIGGRAPH 94 (Brutzman 94b, 94c). Effectiveness of these techniques has been formally evaluated (Gambrino 94) with typically positive and enthusiastic results (Macedonia, Brutzman 94). It is our belief that use of MBone in a variety of media will continue to grow at a slow but exponential rate, and it is our experience that familiarity and practice overcomes limitations associated with bandwidth restrictions.

The combined use of socket connectivity, MBone audio/video/graphics/PDUs, the DIS protocol and World-Wide Web (WWW) functionality means that the underwater virtual world is an excellent application to take advantage of a high-bandwidth information superhighway, further extending the capabilities of multiple researchers. The network approach allows many individuals dynamic remote access, and distributing components minimizes dependence on unique (or hard-to-replace) hardware and software. The DIS protocol permits compatible interaction with other virtual worlds over the Internet. Providing hypermedia access via publicly available WWW browsers such as *Mosaic* makes a complete variety of pertinent archived information available to anyone. Such information media include images, papers, datasets, software, sound clips, text and any other computer-storable media. This supports another long term objective of the project, which is to continue extending the scope of virtual world entities and simplifying virtual world interfaces in order to become useful as an exemplar application for education. Thus an infrastructure is evolving whereby virtual worlds can support remote scientific collaboration and education, both regionally and globally.

E. SUMMARY AND FUTURE WORK

This chapter showed experimental results in hydrodynamics, Internet-wide network loading and remote collaboration. Hydrodynamics behavior of an underwater vehicle is shown to be highly complex and dependent on a large number of interacting variables. Temporal plots permit precise analysis of results, but real-time 3D graphics playback is required for overall evaluation and insight. From a network perspective, the Internet is currently capable of supporting the variety of high-bandwidth information streams needed for full virtual world connectivity. Tested streams include point-to-point telemetry sockets, audio, video, graphics, DIS PDUs and archived multimedia. Addition of arbitrarily large numbers of virtual world viewers is shown possible through use of the MBone for time-sensitive information such as audio, video and DIS position updates.

Future work on experimental results is extensive because use of the underwater virtual enables many new capabilities. Top priority is to reintegrate execution level software in the actual vehicle and reproduce virtual world results in the real world.

Regrettably, the long break in 1994 AUV testing due to hydrogen explosion repairs has precluded running any of these missions in the water. This lack of validating test data duplicating virtual world missions in the real world has precluded performing a "Turing Test" of virtual world operations. A top priority for 1995 is to stabilize the equipment rebuild and upgrade the execution level robot control program to use new hardware interfaces. At that point virtual world test results are expected to be completely validated against identical missions run in the test tank.

Collaboration with other underwater robotics and virtual world researchers is highly desirable in order to scale up the scope of the underwater virtual world. A formal validation and verification confidence assessment can improve project implementation and may help formally clarify the fundamental requirements needed for global internetworked large-scale virtual worlds. Exciting future possibilities include use of the underwater virtual world as an educational tool, as a testbed for new AUVs, and as a means for providing context amidst the gigantic mass of information content which is being connected via the Internet.

X. CONCLUSIONS AND RECOMMENDATIONS

A. PRINCIPAL DISSERTATION CONCLUSIONS

Construction of an underwater virtual world is feasible. Using 3D real-time computer graphics in an underwater virtual world enables effective AUV development. Visualization of robot interactions in an underwater virtual world improves our perceptual ability to evaluate robot performance. A networked robot and virtual world enables researchers to "scale up" to extremely high degrees of complexity, and makes robotics research and collaboration accessible worldwide.

A comprehensive software-hardware architecture for a general AUV underwater virtual world demonstrates the feasibility of these concepts. Several components make up this virtual world. A complete underwater vehicle dynamics model suitable for physics-based real-time simulation is developed. Real-time visualization of AUV and sensors interacting with a realistic environment is achieved by decoupling robot-virtual world interactions from graphics by the use of the Distributed Interactive Simulation (DIS) protocol. World-wide accessibility is provided using the World-Wide Web (WWW) for software archive retrieval and the Multicast Backbone (MBone) for live streams. Convenient and comprehensive network connectivity enables effective scientific collaboration among researchers and robots.

B. SPECIFIC CONCLUSIONS, RESULTS AND RECOMMENDATIONS FOR FUTURE WORK

Additional detailed conclusions appear at the end of each chapter.

1. Underwater Robotics

A large gap between theory and practice has resulted in relatively few underwater robots. The difficulties of the underwater environment are as severe as any other, making any AUV failure potentially fatal. AUV software and hardware solutions are available but many problems remain due to difficulty integrating component solutions. Real-time control under complex dynamic and temporal constraints is essential. Stability and reliability are paramount but testing is difficult. Virtual worlds may help break the AUV development bottleneck.

A large number of open tasks await underwater robot designers. Insistence on in-water testing following laboratory simulation is needed to ensure that progress is cumulative and grounded in reality. AUVs are beginning to perform important tasks that cannot be done effectively by other platforms. Both NPS and the AUV community is poised for significant real world accomplishments such as effective minefield search.

2. Object-Oriented Real-Time Graphics

Interactive 3D graphics are used as our window into virtual worlds. Graphics are completely decoupled from robot-virtual world interaction in order to permit real-time performance for robot and world models. Graphics viewers can be effectively networked to use the global Internet as an input/output device, i.e. as a source or target for any information stream desired. Object-oriented graphics models supported by scene description languages are essential for scaling up virtual worlds. Rendering and network compatibility across multiple platforms is highly desirable. *Open Inventor* and the proposed Virtual Reality Modeling Language (VRML) are well suited for virtual world construction and rendering. Future work includes increasing the number of objects populating the virtual world, adding hooks to objects on remote servers via URL definitions, and porting to other architectures to encourage widespread use of the underwater virtual world. Since the physical size and scope of an underwater virtual world is very large, it is a good candidate for a large immersive environment such as a CAVE (Cruz-Neira 93).

3. Underwater Vehicle Hydrodynamics Models

No hydrodynamics model was previously available which was suitable for real-time simulation performance in hover and cruise modes. A general standardized and parameterized hydrodynamics model is presented based on numerous preceding models, rigorous physical derivations and empirical testing. The model is computationally efficient, capable of running at short intervals (10 Hz) and networked using the DIS protocol. Future work includes implementation of the hydrodynamics model for other vehicles, extension to include special effects such as tethers and wave motion, connection to a large-scale world collision detection model, and possible porting into vehicle software as a predictive "world in the loop" for intelligent control and machine learning supervision.

4. Networking

The key considerations in networking virtual worlds concern compatibility, bandwidth and scalability. The Internet Protocol (IP) suite is essential for global compatibility. Point-to-point sockets are capable of supporting tightly-coupled world models interacting with a robot. The IEEE DIS protocol permits entities to compatibly communicate and interact at the entity/application level. Multicast protocols allow bandwidth reduction for both transmission and reception, a necessary step for scaling beyond limited numbers (hundreds) of interacting entities. Scalability is also supported through the use of World-Wide Web compatibility which provides flexible and well-defined communications mechanisms for information retrieval. Future work includes continuing to scale up using DIS, multicast and the World-Wide Web, and also building a persistent underwater virtual world server for continuous availability to humans and robots using the Internet.

5. Sonar Modeling and Visualization

Many sonar models are available but most are highly specialized and none appear to be suited for general use. Based on initial testing and validation, the Recursive Ray Acoustics (RRA) Algorithm sonar model (Ziomek 93) holds exceptional promise due to computational efficiency, frequency constraint independence, sound speed profile (SSP) constraint independence and bathymetry constraint independence. Sonar visualization is the application of scientific visualization techniques to enable effective analysis of high-dimensionality sonar data.

Sonar visualization also holds great promise as a means of effectively interpreting robot sensor interactions in a counterintuitive environment. Future work starts with porting the RRA ***C***-translation offline model to become an online model, possibly by adding hooks to the original RRA FORTRAN source code. Visualization tools will be used to experiment with effective matches between the high dimensionality of sonar parameters and the high dimensionality of graphics rendering techniques. The most promising visualization results will be used for real-time rendering of sonar transmissions in the underwater virtual world.

C. NEXT STEP: BUILDING A LARGE-SCALE UNDERWATER VIRTUAL WORLD

Results in this dissertation have stressed the possibilities of scalability to arbitrarily large levels. Having shown that comprehensive real-time robot operation is possible in a virtual world, it is now appears possible to scale up while including numerous independently acting human and robot entities. Scientific rigor can be provided by examining inputs, outputs and constraints on various world models and defining ways for these models to interact in theoretically correct ways. The exponential and open growth of the World-Wide Web has been possible due to the open capabilities of the Hypertext Markup Language, providing an excellent exemplar for sustained exponential growth. It is likely that VRML or related efforts will enable similar growth of virtual worlds and virtual world components.

Virtual worlds have the potential to completely change the current paradigms of how people use information. This work points out promising directions for connecting information, robots and people in ways that provide context, meaning and substance. The real world is the best model for a virtual world. When our virtual constructs cumulatively approach realistic levels of depth and sophistication, our understanding of the real world will deepen correspondingly.

APPENDIX A. ACRONYMS

3D	three dimensional
AI	artificial intelligence
ALAN	Acoustic local area network
ANSI	American National Standards Institute
AOSN	Autonomous Oceanographic Sampling Network
API	application program interface
ARPA	Advanced Research Projects Agency (formerly DARPA, originally ARPA)
atan2 (y, x)	arctangent function, returns angle to a point in proper quadrant
ATM	Asynchronous Transfer Mode
AUV	autonomous underwater vehicle
BSD	Berkeley Software Distribution (a Unix variant)
CB	center of buoyancy
CFD	computational fluid dynamics
CG	center of gravity
dB	decibel (logarithmic unit of measure of sound intensity)
DES	Data Encryption Standard
DIS	IEEE Distributed Interactive Simulation Protocol
DoD	U.S. Department of Defense
DOF	degrees of freedom
dynamics	AUV Underwater Virtual World component: hydrodynamics, network connections and other real-time models (e.g. sonar)
execution	AUV Underwater Virtual World component: robot execution level
FEC	forward error compression - encoding scheme that includes redundant information to reduce probability of data loss

ftp	File Transfer Protocol, refers both to host servers providing information and clients connecting for information retrieval
GPS	Global Positioning System
html	Hypertext Markup Language
http	Hypertext Transfer Protocol, refers both to host servers providing information and clients connecting for information retrieval
IEEE	Institute of Electrical and Electronics Engineers
I^3LA	Initiative for Information Infrastructure and Linkage Applications, a Monterey Bay regional K-16 educational network collaboration
IP	Internet Protocol
IPC	inter-process communications
Irix	Unix as implemented on SGI workstations
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
.iv	Open Inventor scene description language filename extension
Kbps	Kilobits per second
LAN	local area network
lynx	WWW browser with line mode interface, usable within application programs as a client to retrieve files from WWW
MAPS	MBARI-NASA Ames-Naval Postgraduate School-Stanford Aerospace Robotics Lab effort to build next-generation AUV
MBone	Multicast Backbone
Mbps	Megabits per second
MIDI	Musical Instrument Digital Interface
MIME	Multipurpose Internet Mail Extensions (RFC 1341)
MOE	measure of effectiveness
Mosaic	WWW browser with graphical user interface
mrouted	multicast router daemon
NPS	Naval Postgraduate School

nv	*network video* MBone application tool for video and graphics
OOSPICs	Object-Oriented Simulation PICtures
OS-9	Real-time operating system produced by Microware Inc.
OSI	Open Systems Interconnection reference model
PDU	protocol data unit
RBM	Rational Behavior Model
RFC	Request For Comments, draft Internet documents provided for information or standards development
ROV	remotely operated vehicle
rpm	revolutions per minute
RRA	Recursive Ray Acoustics sonar algorithm
SAF	semi-automated forces
sd	*session directory* MBone application tool for session advertisement and selection
SDV-9	Swimmer Delivery Vehicle, hull 9
SGI	Silicon Graphics Inc.
sonar	SOund Navigation And Ranging
SPL	sound pressure level
SSP	sound speed profile (measured versus depth)
SVP	sound velocity profile (typically a misnomer for SSP)
Tcl/Tk	Tool control language/Tool kit
TCP	Transmission Control Protocol, part of IP transport layer
telnet	virtual terminal protocol permitting remote system login
ttl	time-to-live packet hop counter
UDP	User Datagram Protocol, part of IP transport layer
URL	Universal Resource Locator
USN	U.S. Navy
UUV	unmanned underwater vehicle, may be controlled remotely
vat	*visual audio tool* MBone application tool for audio

viewer	AUV Underwater Virtual World component: networked 3D real-time graphics to remotely view robot operating in virtual world
vrml	Virtual Reality Modeling Language
VV & A	validation verification and accreditation
wb	*whiteboard* MBone application tool for shared drawing and images
WWW	World-Wide Web

APPENDIX B. VIDEO DEMONSTRATION

A. INTRODUCTION

This section briefly describes the contents and objectives of the video appendix.

B. NPS AUV OPERATING IN THE UNDERWATER VIRTUAL WORLD: THE SIGGRAPH MISSION

This video segment shows a six minute mission in the underwater virtual world. It first appeared in the Video Proceedings of the AUV 94 conference (Brutzman 94a). The original abstract follows:

> A critical bottleneck exists in Autonomous Underwater Vehicle (AUV) design and development. It is tremendously difficult to observe, communicate with and test underwater robots, because they operate in a remote and hazardous environment where physical dynamics and sensing modalities are counterintuitive.
>
> An underwater virtual world can comprehensively model all salient functional characteristics of the real world in real time. This virtual world is designed from the perspective of the robot, enabling realistic AUV evaluation and testing in the laboratory. Three-dimensional real-time graphics are our window into that virtual world. Visualization of robot interactions within a virtual world permits sophisticated analyses of robot performance that are otherwise unavailable. Sonar visualization permits researchers to accurately "look over the robot's shoulder" or even "see through the robot's eyes" to intuitively understand sensor-environment interactions.
>
> Distribution of underwater virtual world components enables scalability and real-time response. The IEEE Distributed Interactive Simulation (DIS) protocol is used for compatible live interaction with other virtual worlds. Network access allows individuals remote observation. *Mosaic* and the World-Wide Web provides open access to archived images, papers, datasets, software, sound clips, text and any other computer-storable media. This project presents the frontier of 3D real-time graphics for underwater robotics, ocean exploration, sonar visualization and worldwide scientific collaboration. (Brutzman 94a)

C. NPS AUTONOMOUS UNDERWATER VEHICLE

This video segment shows the basic functionality of the NPS AUV. It first appeared in the Video Proceedings of the IEEE International Conference on Robotics and Automation 1992, and the Video Proceedings of the Eighth International Symposium on Unmanned Untethered Submersible Technology (Brutzman 92a, 92b, 93a). The original abstract follows:

> The Naval Postgraduate School (NPS) Autonomous Underwater Vehicle (AUV) is an eight foot long, 387-pound untethered robot submarine designed for research in adaptive control, mission planning, navigation, mission execution, and post-mission data analysis. Neutral buoyancy, eight plane surfaces and twin propellers allow precise maneuverability.
>
> Simulation programs running on Iris three-dimensional graphics workstations are used to evaluate NPS AUV software and predict system performance prior to each mission. Graphics simulations can replay in real time actual data collected in the pool. The taped playback demonstrates reconstruction and visualization of vehicle track, control systems dynamic response, logic and state changes, plotted locations of individual sonar returns, and expert system classification of detected objects.
>
> Ongoing NPS AUV research is investigating linear and nonlinear control techniques, advanced sonar classification, failure mode analysis using neural networks, dynamic path and search planning, use of cross-body thrusters for hovering control, alternate AUV operating architectures, incorporation of Global Positioning System (GPS) receiver navigation, and construction of an underwater virtual world to permit complete and realistic testing of every aspect of AUV operation in the laboratory. (Brutzman 92a, 92b, 93a)

D. LIVE EXHIBIT AND WORLDWIDE MULTICAST AT *The Edge*, SIGGRAPH 94

This segment shows the NPS AUV Underwater Virtual World exhibit at *The Edge*, SIGGRAPH 94 (Brutzman 94b, 94c). Video photographer is Michael J. Zyda.

E. NPS AUV WORLD-WIDE WEB HOME PAGE

This video segment shows how to connect to the NPS AUV home page and retrieve the underwater virtual world distribution. Installation is also demonstrated.

F. EXTENDED NPS AUV MISSION REPLAYS

Additional NPS AUV missions are run from the underwater virtual world archive distribution, evaluating a variety of hydrodynamic and sonar responses.

G. NPS AUV POSTURE CONTROL

This video segment demonstrates in-water test tank results from early 1994. It first appeared in the Video Proceedings of the AUV 94 conference (Brutzman 94a) by authors A.J. Healey, D.B. Marco, R.B. McGhee, D.P. Brutzman, R. Cristi and F.A. Papoulias. The original abstract follows:

> Recent work with the NPS AUV II demonstrates further development of the execution level software to incorporate hover control behavior in the NPS hover tank. Of particular interest is the use of the ST 100 and ST 725 high frequency sonars to provide data about the environment. Thus positioning can be accomplished without the use of beacons.
>
> Motion behaviors may be instituted that include diving and pitch control under thruster power, heading control at zero speed, lateral and longitudinal positioning, as well as the automatic initiation of filters as needed when a new target is found. A simple task level language is developed that will be used to direct tactical level output to a port which communicates with execution level software. (Healey abstract, Brutzman 94a)

H. MBone: AUDIO/VIDEO INTERNET TOOLS FOR INTERNATIONAL COLLABORATION

This video segment describes and demonstrates use of the MBone. It was recorded from a worldwide MBone broadcast of the May 1994 International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments, hosted by the Monterey Bay Aquarium Research Institute (MBARI). It originally appeared at in the Video Proceedings of the AUV 94 conference (Brutzman 94a). MBone media strengths and limitations videoconference are formally evaluated in (Gambrino 94) using this videoconference. The original abstract follows:

> Recently it has become possible to broadcast live audio and video over the Internet using the Multicast Backbone (MBone). This development holds great promise as an enabling technology for collaborative work among underwater vehicle researchers separated by long distances.
>
> This talk describes technical considerations related to use of the MBone, which is the virtual network used for these Internet sessions. Anyone with direct Internet connections, adequate bandwidth and a workstation can receive multicast. We hope to demonstrate that worldwide collaboration among underwater robotics researchers is not only feasible but even convenient.
>
> For more information on how to connect your lab to MBone, refer to "MBone Provides Audio and Video Across the Internet" in the April 94 issue of *IEEE COMPUTER*, pp. 30-36. This article is also available for electronic retrieval in PostScript, text, and hypertext versions:
> ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.ps
> ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.txt
> ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.html (Brutzman 94a)

REFERENCES

Abel, Jacob M., "Cable Interactions in a Depth Controlled Submersible," *Journal of Hydronautics,* vol. 6 no. 2, July 1972, pp. 83-89.

Alexander, J.C. and Maddocks, J.H., "On the Kinematics of Wheeled Robots," *Autonomous Robot Vehicles*, Cox, I.J. and Wilfong, G.T., editors, Springer-Verlag, New York, 1990, pp. 5-24.

Allen, Joseph P., "Physics at the edge of the Earth," *Pictorial communication in virtual and real environments*, Stephen R. Ellis, editor, Taylor and Francis, London, 1991, pp. 12-21.

Amyot, J.R., editor, *Hovercraft Technology, Economics and Applications*, Elsevier, Amsterdam, 1989.

Armitage, Grenville, "IP Multicast over UNI 3.0 based ATM Networks," Internet draft, *ftp://ds.internic.net/internet-drafts/draft-armitage-ipatm-ipmc-01.txt*, October 3 1994.

Auran, Per Gunnar and Silven, Olli, "Ideas for Underwater 3D Sonar Range Sensing and Environmental Modelling," *International Federation of Automatic Control (IFAC) Workshop on Control Applications in Marine Systems (CAMS) 95*, Trondheim Norway, May 10-12 1995, in review.

Babb, Ivar G. and Scheifele, Peter, "Marine Science Education Using Low-Cost ROV Technology," *Marine Technology Society Journal*, vol. 26 no. 4, Winter 1992-1993, pp. 73-78.

Badr, Salah M., Byrnes, Ronald B. Jr., Brutzman, Donald P. and Nelson, Michael L., *Real-Time Systems*, technical report NPS-CS-92-004, Naval Postgraduate School, Monterey California, February 1992.

Bahrke, Fredric G., *On-Line Identification of the Speed, Steering and Diving Response Parameters of an Autonomous Underwater Vehicle from Experimental Data*, Master’s Thesis, Naval Postgraduate School, Monterey California, March 1992.

Baiardi, Paolo, Bono, Riccardo, Cannata, Giorgio, Casalino, Guiseppe and Veruggio, Gianmarco, "Dynamic Simulation of Underwater Vehicles," *Fourth International Advanced Robotics Programme (IARP) Workshop on Underwater Robotics*, Genoa Italy, November 4-5 1992, 7 pages.

Bailey, Michael P., "Object-Oriented Simulation Pictures (OOSPICs) for Design and Testing," *Proceedings of the 1994 Summer Computer Simulation Conference*, San Diego California, July 18-20 1994, pp. 67-76.

Bales, James W. and Bellingham, James G., editors, "Report on the Anglo-US Meeting on Cooperation for Cleaner Seas: Brighton United Kingdom, 6-7 March 1994," MIT Sea Grant report 94-21, Cambridge Massachusetts, 26 pages.

Bales, James W. and Levine, Edward R., "Sensors for Oceanographic Applications of Autonomous Underwater Vehicles," *Proceedings of the 21st Annual Symposium of the Association for Unmanned Vehicle Systems,* Detroit Michigan, May 23-25 1994, pp. 439-446.

Ballard, Robert D., "The MEDEA/JASON remotely operated vehicle system," *Deep-Sea Research*, Pergamon Press, vol. 40 no. 8, pp. 1673-1687, 1993.

Barr, Alan H., "Teleological Modeling," *Making Them Move: Mechanics, Control and Animation of Articulated Figures*, Badler, Norman I., Barsky, Brian A., and Zeltzer, David A., editors, Morgan Kauffmann Publishers Inc., San Mateo California, 1991, pp. 315-321.

Barzel, Ronen, *Physically-Based Modeling for Computer Graphics - A Structured Approach*, Academic Press, Boston, 1992.

Bate, Roger R., Mueller, Donald D. and White, Jerry E., *Fundamentals of Astrodynamics*, Dover Publications Inc., New York, 1971.

Bekker, M.G, *Theory of Land Locomotion: The Mechanics of Vehicle Mobility,* University of Michigan Press, Ann Arbor Michigan, 1956.

Bekker, M.G, *Introduction to Terrain-Vehicle Systems*, University of Michigan Press, Ann Arbor Michigan, 1969.

Belinfante, Axel, "Say..." text to speech sound server, University of Twente, Netherlands, *http://www_tios.cs.utwente.nl/say/?*, 1994.

Bell, Gavin, Parisi, Anthony and Pesce, Mark, "The Virtual Reality Modeling Language (VRML) Version 1.0 Specification," draft, *http://www.eit.com/vrml/vrmlspec.html*, 3 November 1994.

Bellingham, J.G. and Consi, T.R., "State Configured Layered Control," *Proceedings of the First IARP Workshop on Mobile Robots for Subsea Environments,* Monterey California, October 23-26 1990, pp. 75-80.

Bellingham,, J.G., Goudey, C.A., Consi, T.R., Bales, J.W., Atwood, D.K., Leonard, J.J. and Chryssostomidis, C., "A Second Generation Survey AUV," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 148-155.

Belton, Jerry T., *Component Level Simulation for Autonomous Underwater Vehicles*, Master's Thesis, Naval Postgraduate School, Monterey California, December 1993.

Berners-Lee, Tim, Cailliau, Luotonen, Ari, Nielsen, Henrik Frystyk, and Secret, Arthur, "The World-Wide Web," *Communications of the ACM*, vol. 37 no. 8, August 1994, pp. 76-82.

Berners-Lee, Tim and Frystyck, Henrik, *www* World-Wide Web hypertext browser, *http://info.cern.ch/hypertext/WWW/TheProject.html*, European Particle Physics Laboratory (CERN), Geneva Switzerland, 1994.

Blidberg, D.R., Chappell, S., Jalbert, J., Turner, R., Sedor, G. and Eaton, P., "The EAVE AUV Program at the Marine Systems Engineering Laboratory," *Proceedings of the First IARP Workshop on Mobile Robots for Subsea Environments,* Monterey California, October 23-26 1990, pp. 33-42.

Blumberg, Alan F., Signell, Richard P. and Jentner, Harry L., "Modeling Transport Processes in the Coastal Ocean," *Environmental Science in the Coastal Zone: Issues for Further Research,* National Research Council Commission on Geosciences, Environment and Resources, National Academy Press, Washington DC, 1994.

Bonasso, R. Peter, Yoerger, Dana R. and Stewart, W. Kenneth, "Semi-Autonomous Underwater Vehicles for Shallow Water Mine Clearing," *IEEE Oceanic Engineering Society Symposium on Autonomous Underwater Vehicles*, Washington DC, June 2-3 1992, pp. 22-28.

Booch, Grady, *Object-Oriented Design with Applications*, Benjamin/Cummings Publishing, Redwood City California, 1991.

Borenstein, Nathaniel S. and Freed, Ned, "MIME (Multipurpose Internet Mail Extensions," Request for Comments (RFC) 1341, *ftp://ds.internic.net/rfc/rfc1341.txt,* September 1993.

Brancart, Claude P., "The Evolution of the DARPA/ARPA UUV Program," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, May 3-6 1994, 12 pages.

Bréant, F., Lê Văn Gông, H. and Soto, M., "Towards Open Virtual Environments Interoperability," *Fourth International Conference on Artificial Reality and Tele-Existence (ICAT)*, Otemachi Tokyo Japan, July 14-15 1994, pp. 195-203.

Bricken, William and Coco, Geoffrey, "The VEOS Project," *PRESENCE: Teleoperations and Virtual Environments*, vol. 3 no. 2, Spring 1994, pp. 111-129.

Brooks, Rodney A, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. RA-2 no. 1, March 1986, pp. 14-23.

Brooks, Rodney A., *Sensor-Based Planning and Control of Mobile Robots*, videotaped conference tutorial and slides, IEEE Robotics and Automation Society, New York, 1990.

Brown, James P., *Four Quadrant Dynamic Model of the AUV II Thruster*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1993.

Brown, Peter J., "Diving Into the Future," *U.S. Naval Institute Proceedings*, vol. 119 no. 7, July 93, pp. 76-80.

Brutzman, Donald P. and Compton, Mark A., "AUV Research at the Naval Postgraduate School," *Sea Technology*, vol. 32 no. 12, December 1991, pp. 35-40.

Brutzman, Donald P., *NPS AUV Integrated Simulator*, Master's Thesis, Naval Postgraduate School, Monterey California, March 1992.

Brutzman, Donald P., Floyd, Charles A. and Whalen, Russell, "Naval Postgraduate School Autonomous Underwater Vehicle," *Video Proceedings of the IEEE International Conference on Robotics and Automation 92*, Nice, France, May 1992.

Brutzman, Donald P., Kanayama, Yutaka, and Zyda, Michael J., "Integrated Simulation for Rapid Development of Autonomous Underwater Vehicles," *Proceedings of the IEEE Oceanic Engineering Society Conference AUV 92*, Washington DC, June 2-3 1992, pp. 3-10.

Brutzman, Donald P., "From virtual world to reality: designing an autonomous underwater robot," *AAAI Fall Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots*, Cambridge Massachusetts, October 23-25 1992, pp. 18-22.

Brutzman, Donald P., Compton, Mark A. and Kanayama, Yutaka, "Autonomous Sonar Classification using Expert Systems," *Proceedings of the IEEE Oceanic Engineering Society Conference OCEANS 92*, Newport Rhode Island, October 26-29 1992, pp. 554-559.

Brutzman, Donald P., editor, *Video Proceedings of the Eighth International Symposium on Unmanned Untethered Submersible Technology*, University of New Hampshire, Durham, New Hampshire, September 27-29 1993.

Brutzman, Donald P., "Beyond intelligent vacuum cleaners," *American Association for Artificial Intelligence (AAAI) Fall Symposium on Applications of Artificial Intelligence for Instantiating Real-World Agents*, Raleigh, North Carolina, October 22-24 1993, pp. 23-25.

Brutzman, Donald P., editor, *Video Proceedings of the IEEE Oceanic Engineering Society 1994 Autonomous Underwater Vehicle Conference*, Charles Stark Draper Laboratories, Cambridge Massachusetts, July 19-20 1994. Includes two self-produced video segments: "A Virtual World for an Autonomous Underwater Vehicle" and "MBone: Multicast Backbone Audio/Video Tools for International Collaboration."

Brutzman, Don, Zyda, Michael J., Barham, Paul T., Falby, John S., Healey, Anthony J., Isakari, Shirley, Luck, Rodney, Macedonia, Michael R., McGhee, Robert B., Pratt, David R. and Ziomek, Lawrence J., "A Virtual World for an Autonomous Underwater Vehicle," exhibit at "The Edge," *SIGGRAPH 94 Visual Proceedings*, Orlando Florida, July 24-29 1994, pp. 204-205.

Brutzman, Don and Paxinos, Gary, "MBone at SIGGRAPH 94," *COMPUTER GRAPHICS*, Association for Computing Machinery (ACM) SIGGRAPH, vol. 28 no. 4, November 1994, pp. 300-301.

Brutzman, Don, "MBone at AUV 94," *IEEE Oceanic Engineering Society Newsletter*, Fall 1994, pp. 8-9.

Brutzman, Donald P., *Software Reference: A Virtual World for an Autonomous Underwater Vehicle*, technical report NPS-CS-010-94, Naval Postgraduate School, Monterey California, December 1994. The accompanying public electronic distribution of this reference includes source code and executable programs. World-Wide Web (WWW) Uniform Resource Locator (URL) is *ftp://taurus.cs.nps.navy.mil/pub/auv/auv.html*

Brutzman, Don, editor, "Regional Network Design Tiger Team White Paper," Initiative for Information Infrastructure & Linkage Applications (I^3LA), Monterey California, 1994. Available electronically via I^3LA WWW home page at *ftp://taurus.cs.nps.navy.mil/pub/i3la/i3la.html*

Bryson, Steve and Levit, Creon, "The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows," *IEEE Visualization 91,* Boston Massachusetts, October 22-25 1991, pp. 17-24.

Byrnes, R.B., MacPherson, D.L., Kwak, S.H., McGhee, R.B. and Nelson, M.L., "An Experimental Comparison of Hierarchical and Subsumption Software Architectures for Control of an Autonomous Underwater Vehicle," *Proceedings of the IEEE Oceanic Engineering Society Conference AUV 92,* Washington DC, June 2-3, 1992, pp. 135-141.

Byrnes, Ronald Benton Jr., *The Rational Behavior Model: A Multi-Paradigm, Tri-Level Software Architecture for the Control of Autonomous Vehicles,* Ph.D. Dissertation, Naval Postgraduate School, Monterey California, March 1993.

Byrnes, R.B., Mcghee, R.B., Kwak, S.H., Healey, A.J., and Brutzman, D.P., "The Rational Behavior Model: A Top-Down Approach to Autonomous Underwater Vehicle Control," in preparation for *IEEE Transactions on Systems*, *Man and Cybernetics*, 1995.

Calder, Nigel, "The Great Battery Debate: Gel Cell or Wet Cell?," *Cruising World*, March 1994, pp. 85-91.

Calvin, James, Dickens, Alan, Gaines, Bob, Metzger, Paul, Miller, Dale, and Owen, Dan, "The SIMNET Virtual World Architecture," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle Washington, September 18-22 1993, pp. 450-455.

Carlsson, Christer and Hagsand, Olof, "DIVE - a Multi-User Virtual Reality System," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle Washington, September 18-22 1993, pp. 394-400.

Carriero, Nicholas and Gelernter, David, *How To Write Parallel Programs: A First Course*, MIT Press, Cambridge Massachusetts, 1990.

Catipovic, Josko, Brady, David and Etchemendy, Steven, "Development of Underwater Acoustic Modems and Networks," *Oceanography*, vol. 6 no. 3, December 1993, pp. 112-119.

Chappell, Steven G., Jalbert, James C., Pietryka, Paul and Duchesney, John, "Acoustic Communication between Two Autonomous Underwater Vehicles," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 462-469.

Chen, ChuXin, Trivedi, Mohan M. and Bidlack, Clint R., "Simulation and Animation of Sensor-Driven Robots," *IEEE Transactions on Robotics and Automation,* vol. 10 no. 5, October 1994, pp. 684-704.

Choi, S.K., Takashige, G.Y. and Yuh, J., "Experimental Study on an Underwater Robotic Vehicle: ODIN," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 79-84.

Chou, Jack C.K., "Quaternion Kinematic and Dynamic Differential Equations," *IEEE Transactions on Robotics and Automation*, vol. 8 no. 1, February 1992, pp. 53-64.

Codella, Christopher F., Jalili, Reza, Koved, Lawrence, and Lewis, J. Bryan, "A Toolkit for Developing Multi-User, Distributed Virtual Environments," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle Washington, September 18-22 1993, pp. 401-407.

Cody, Steven E., *An Experimental Study of the Response of Small Tunnel Thrusters to Triangular and Square Wave Inputs*, Master’s Thesis, Naval Postgraduate School, Monterey California, December 1992.

Comer, Douglas E., *Internetworking with TCP/IP Volume I: Principles, Protocols and Architecture*, second edition, Prentice Hall, Englewood Cliffs New Jersey, 1991.

Compton, Mark A., *Minefield Search and Object Recognition for Autonomous Underwater Vehicles*, Master’s Thesis, Naval Postgraduate School, March 1992.

Cooke, Joseph C., *NPSNET: Flight Simulation Dynamic Modeling using Quaternions,* Master’s Thesis, Naval Postgraduate School, March 1992.

Cooke, J.C., Zyda, M.J., Pratt, D.R. and McGhee, R.B., "NPSNET: Flight Simulation Dynamic Modeling using Quaternions," *PRESENCE: Teleoperations and Virtual Environments*, vol. 1 no. 4, Fall 1992, pp. 404-420.

Cottle, Dean J., *Mine Avoidance and Localization for Underwater Vehicles using Continuous Curvature Path Generation and Nonlinear Tracking Control*, Master’s Thesis, Naval Postgraduate School, Monterey California, September 1993.

Covington, James Harvey Jr., *Implementing an Ocean Theater in NPSNET*, Master's Thesis, Naval Postgraduate School, Monterey California, March 1994.

Cristi, R. and Healey, A.J., "Adaptive Identification and Control of an Autonomous Underwater Vehicle," *Sixth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, IEEE Ocean Engineering Society and University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, June 12-14 1989, pp. 563-572.

Cruz-Neira, Carolina, Leigh, Jason, Papka, Michael, Barnes, Craig, Cohen, Steven M., Das, Sumit, Engelmann, Roger, Hudson, Randy, Roy, Trina, Siegel, Lewis, Vasilakis, Christina, DeFanti, Thomas A. and Sandin, Daniel J., "Scientists in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment," *IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, San Jose California, October 25-26 1993, pp. 59-66 and CP-3.

Curtin, Thomas B., Bellingham, James G., Catipovic, Josko and Webb, Doug, "Autonomous Oceanographic Sampling Networks," *Oceanography*, vol. 6 no. 3, December 1993, pp. 86-94.

Dayan, Paul S., *The OS-9 Guru: The Facts*, Galactic Industrial Limited, Durham, United Kingdom, 1992.

D'Azzo, John J. and Houpis, Constantine H., *Linear Control System Analysis and Design*, third edition, McGraw-Hill, New York, 1988.

Deering, Steve, "Host Extensions for IP Multicasting," Request for Comments (RFC) 1112, *ftp://ds.internic.net/rfc/rfc1112.txt*, August 1989.

Dorato, Peter, editor, *Robust Control*, IEEE Press, New York, 1987.

Draper, Richard N., "Supercomputing 93," IEEE *Computational Science and Engineering*, vol. 1 no. 1, Spring 1994, pp. 85-86.

Durlach, Nathaniel I. and Mavor, Anne S., editors, *Virtual Reality: Scientific and Technological Challenges*, National Research Council report, National Academy Press, Washington DC, 1994.

Elfes, Alberto, "A distributed control architecture for an autonomous mobile robot," *Artificial Intelligence*, vol. 1 no. 2, 1986.

Emerson, Toni, "On The Net: Internet Resources in Virtual Reality," World-Wide Web home page, *ftp://ftp.u.washington.edu/public/VirtualReality/HITL/HITLMosaic/onthenet.html*, November 1994.

Engelson, Sean P. and Bertani, Niklas, "Ars Magna: The Abstract Robot Simulator Manual," version 1.0, Department of Computer Science technical report, Yale University, New Haven Connecticut, October 1992.

Etter, Paul C., *Underwater Acoustic Modeling: Principles, Techniques and Applications*, Elsevier Applied Science, London, 1991.

Etter, Paul C., "Underwater Acoustic Modeling," tutorial, *IEEE Oceanic Engineering Society Conference OCEANS 92*, Newport Rhode Island, October 26-29 1992.

Feldman, Gene, Haleblian, Ray, Maffei, Andrew, Marra, Marty, Pape, Dave and Stewart, Ken, "JASON Interactive Mapper," Virtual Reality Room (VROOM) CAVE exhibit, *SIGGRAPH 94 Visual Proceedings*, Orlando Florida, July 24-29 1994, p. 256.

Feldman, J., *DTNSRDC Revised Standard Submarine Equations of Motion*, technical report DTNSRDC/SPD-0393-09, David W. Taylor Naval Ship Research and Development Center (DTNSRDC), Bethesda Maryland, June 1979.

Foley, James D, van Dam, Andries, Feiner, Steven K. and Hughes, John F., *Computer Graphics: Principles and Practice*, second edition, Addison-Wesley, Reading Massachusetts, 1990.

Fortner, Brand, *The Data Handbook: A Guide to Understanding the Organization and Visualization of Technical Data*, Spyglass Inc., Champaign Illinois, 1992.

Fossen, Thor I., *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, Chichester England, 1994.

Frakes, William B., Fox, Christopher J. and Nejmeh, Brian A., *Software Engineering in the UNIX®/C Environment*, Prentice Hall, Englewood Cliffs New Jersey, 1991.

Frank, A.A. and McGhee, R.B., "Some Considerations Relating to the Design of Autopilots for Legged Vehicles," *Journal of Terramechanics*, vol. 6 no. 1, Pergamon Press, Great Britain, 1969, pp. 23-35.

Frank, Thomas A. and Sheridan, Alan L., "Smart Communications System (SMARTCOMMS)," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 7-12.

Fricke, J. Robert, "Down to the Sea in Robots," *Technology Review*, Association of Alumni and Alumnae of MIT, Cambridge Massachusetts, October 1994, pp. 46-55.

Fu, K.S., Gonzalez, R.C. and Lee, C.S.G., *Robotics: Control, Sensing, Vision and Intelligence*, McGraw-Hill, New York, 1987.

Funda, Janez, Taylor, Russell and Paul, Richard P., "On Homogenous Transforms, Quaternions, and Computational Efficiency," *IEEE Transactions on Robotics and Automation*, vol. 6 no. 3, June 1990, pp. 382-388.

Gambrino, John Robert, *An Analysis of Internet's MBone: A Media Choice Perspective*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1994.

Gargano, J. and Wasley, D., "K-12 Internetworking Guidelines," Request for Comments (RFC) 1709, *ftp://ds.internic.net/rfc/rfc1709.txt*, November 1994.

Gelernter, David, *Mirror Worlds - or the Day Software Puts the Universe in a Shoebox... How It Will Happen and What It Will Mean*, Oxford University Press, New York, 1992.

Gelernter, David and Carriero, Nicholas, "Coordination Languages and their Significance," *Communications of the ACM*, vol. 35 no. 2, February 1992, pp. 96-107.

Gertler, Morton and Hagen, Grant R., *Standard Equations of Motion for Submarine Simulation*, Naval Ship Research and Development Center (NSRDC) Research and Development Report 2510, Washington DC, June 1967.

Goheen, K.R., Jefferys, E.R. and Broome, D.R., "Robust Self-Designing Controllers for Underwater Vehicles," *Transactions of the ASME*, vol. 109, May 1987, pp. 170-178.

Green, Mark, "Using Dynamics in Computer Animation: Control and Solution Issues," *Making Them Move: Mechanics, Control and Animation of Articulated Figures*, Badler, Norman I., Barsky, Brian A., and Zeltzer, David A., editors, Morgan Kauffmann Publishers Inc., San Mateo California, 1991, pp. 281-314.

Greenwood, Donald T., *Principles of Dynamics*, second edition, Prentice Hall, Englewood Cliffs New Jersey, 1988.

Halang, Wolfgang A. and Stoyenko, Alexander D., *Constructing Predictable Real Time Systems*, Kluwer Academic Publishers, Boston, 1991.

Hamming, Richard W., *Numerical Methods for Scientists and Engineers*, second edition, Dover Publications Inc., Mineola New York, 1986.

Haynes, Keith, *Computer Graphics Tools for the Visualization of Spacecraft Dynamics*, Master's Thesis, Naval Postgraduate School, December 1993.

Healey, A.J., Papoulias, F.A. and Cristi, R., "Design and Experimental Verification of a Model-Based Compensator for Rapid AUV Depth Control," *Sixth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, IEEE Ocean Engineering Society and University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, June 12-14 1989, pp. 458-474.

Healey, A.J., McGhee, R.B., Cristi, F., Papoulias, F.A., Kwak, S.H., Kanayama, Y., Lee, Y., Shukla, S. and Zaky, A., "Research on Autonomous Underwater Vehicles at the Naval Postgraduate School," *Naval Research Reviews*, Office of Naval Research, Washington DC, vol. XLIV no. 1, Spring 1992.

Healey, A.J. and Marco, D.B., "Experimental Verification of Mission Planning by Autonomous Mission Execution and Data Visualization using the NPS AUV II," *Proceedings of the IEEE Oceanic Engineering Society Conference AUV 92*, Washington DC, June 2-3 1992, pp. 65-72.

Healey, Anthony J., "Dynamics of Marine Vehicles," course notes, Naval Postgraduate School, Monterey California, Fall 1992.

Healey, A.J. and Lienard, D., "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, vol. 18 no. 3, July 1993, pp. 327-339.

Healey, A.J., Marco, D.B., McGhee, R.B., Brutzman, D.P., Cristi, R., Papoulias, F.A. and Kwak, S.H., "Tactical/Execution Level Coordination for Hover Control of the NPS AUV II using Onboard Sonar Servoing," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 129-138.

Healey, A.J., Rock, S.M., Cody, S., Miles, D. and Brown, J.P., "Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 340-352.

Hine, Butler P. III, Stoker, Carol, Sims, Michael, Rasmussen, Daryl, Hontalas, Phil, Fong, Terrence W., Steele, Jay, Barch, Don, Anderson, Dale, Miles, Eric and Nygren, Erik, "The Application of Telepresence and Virtual Reality to Subsea Exploration," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, May 3-6 1994, 10 pages.

Hover, Franz S., Grosenbaugh, Mark A. and Triantafyllou, Michael S., "Calculation of Dynamic Motions and Tensions in Towed Underwater Cables," *IEEE Journal of Oceanic Engineering*, vol. 19 no. 3, July 1994, pp. 449-457.

Huang, Thomas T. and Feldman, Jerome P., *Development of Mathematical Models by Experimental and Analytical Methods for Predicting the Stability, Control and Maneuvering Characteristics of Submarines*, technical report, David Taylor Research Center, Bethesda Maryland, June 1988.

Hughes, Kevin, "Entering the World-Wide Web: A Guide to Cyberspace," *ftp://ftp.eit.com/pub/web.guide/guide.61/guide.61.ps.Z*, May 1994.

Humphreys, Douglas E., "The Impact of Speed-Varying Hydrodynamics Coefficients on UUV Maneuvering Performance," *Sixth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, IEEE Ocean Engineering Society and University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, June 12-14 1989, pp. 548-553.

Humphreys, Douglas E. and Smith, Neill S., "Hydrodynamics and Control of a Streamlined UUV Operating at 180-Degree Angle of Attack," *Seventh International Symposium on Unmanned Untethered Submersible Technology (UUST)*, University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, September 23-25 1991, pp. 171-190.

Hyland, John C. and Taylor, Fred J., "Mine Avoidance Techniques for Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, vol. 18 no. 3, July 1993, pp. 340-350.

IEEE Standard for Information Technology - Protocols for Distributed Interactive Simulation (DIS) Applications, version 2.0, Institute for Simulation and Training report IST-CR-93-15, University of Central Florida, Orlando Florida, May 28 1993.

IEEE Standard for Distributed Interactive Simulation -- Application Protocols, Standards Proposal P1278.1 ballot draft, IEEE Standards Department, Piscataway New Jersey, November 1994.

IEEE Standard for Distributed Interactive Simulation -- Communication Architecture Requirements, Standards Proposal P1278.2 ballot draft, IEEE Standards Department, Piscataway New Jersey, November 1994.

Internet Network Information Center (NIC), *Request For Comments (RFC)*, network archive, *ftp://ds.internic.net*, 1994.

Iyengar, S. Sitharama and Elfes, Alberto, editors, *Autonomous Mobile Robots: Perception, Mapping and Navigation*, volume 1, IEEE Computer Society Press, Los Alamitos, California, 1990.

Iyengar, S. Sitharama and Elfes, Alberto, editors, *Autonomous Mobile Robots: Control, Planning and Architecture*, volume 2, IEEE Computer Society Press, Los Alamitos, California, 1990.

Jackson, Harry A., "Fundamentals of Submarine Concept Design," *Society of Naval Architects and Marine Engineers (SNAME) Annual Conference*, New York, October 28-31 1992, pp. 15-1 through 15-22.

Jurewicz, T.A., *A Real Time Autonomous Underwater Vehicle Dynamic Simulator*, Master's Thesis, Naval Postgraduate School, Monterey California, December 1990.

Kamgar-Parsi, Behzad, Rosenblum, Lawrence J. and Belcher, Edward O., "Underwater Acoustic Imaging," *IEEE Computer Graphics and Applications*, vol. 12 no. 4, July 1992, pp. 11-13.

Karahalios, Margarida A., *Underwater Source Localization using Scientific Data Visualization*, Master's Thesis, University of South Florida, July 1991.

Kazman, Rick, "Making WAVES: On the Design of Architectures for Low-end Distributed Virtual Environments," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle Washington, September 18-22 1993, pp. 443-449.

Keller, Peter R. and Keller, Mary M., *Visual Cues: Practical Data Visualization*, IEEE Computer Society Press, Los Alamitos California, 1993.

Kneppell, Peter L. and Arangno, Deborah C., *Simulation Validation: A Confidence Assessment Methodology*, IEEE Computer Society Press, Los Alamitos California, 1993.

Kolve, D.I., "Describing an Attitude," *16th Annual American Astronautical Society (AAS) Guidance and Control Conference,* Keystone Colorado, February 6-10, 15 pages.

Kuroda, Yoji, Ura, Tamaki and Aramaki, Koji, "Vehicle Control Architecture for Operating Multiple Vehicles," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 323-329.

Kwak, S.H., McGhee, R.B. and Bihari, T.E., "Rational Behavior Model: A Tri-level Multiple Paradigm Architecture for Robot Vehicle Control Software," technical report NPS-CS-92-003, Naval Postgraduate School, Monterey California, March 1992.

Laplante, Phillip, *Real-Time Systems Design and Analysis: An Engineer's Handbook*, IEEE Press, New York, 1993.

Larson, Wiley J. and Wertz, James R., editors, *Space Mission Analysis and Design*, second edition, Microcosm Inc. and Kluwer Academic Publishers, Torrance California, 1992.

Latombe, Jean-Claude, *Robot Motion Planning*, Kluwer Academic Publishers, Boston 1991.

Law, Averill M. and Kelton, W. David, *Simulation Modeling and Analysis*, second edition, McGraw-Hill, New York, 1991.

Leonard, John. J. and Durrant-Whyte, Hugh F., *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, Boston, 1992.

Lewis, Edward V., editor, *Principles of Naval Architecture*, volume III, second revision, Society of Naval Architects and Marine Engineers (SNAME), Jersey City New Jersey, 1988, pp. 188-190 and 418-423.

Li, Zexiang and Canny, J.F., editors, *Nonholonomic Motion Planning*, Kluwer Academic Publishers, Boston, 1993.

Lin, Kuo-Chi and Ng, Huat, "Coordinate Transformations in Distributed Interactive Simulation (DIS)," *SIMULATION*, vol. 61 no. 5, November 1993, pp. 326-331.

Lin, Kuo-Chi and Schab, Daniel E., "The Performance Assessment of the Dead Reckoning Algorithms in DIS," *SIMULATION*, vol. 63 no. 5, November 1994, pp. 318-325.

Macedonia, Michael R. and Brutzman, Donald P., "MBone Provides Audio and Video Across the Internet," *IEEE COMPUTER*, vol. 27 no. 4, April 1994, pp. 30-36.

Macedonia, Michael R., Zyda, Michael J., Pratt, David R., and Barham, Paul T., "Exploiting Reality with Multicast Groups: A Network Architecture for Large Scale Virtual Environments," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Research Park North Carolina, 11-15 March 1995, to appear.

Macedonia, Michael R., Brutzman, Donald P., Zyda, Michael J., Pratt, David R., Barham, Paul T. and Falby, John, "NPSNET: Demonstration of a Multi-Player Virtual Environment over the Internet," *SIGGRAPH Symposium on Interactive 3D Graphics*, Monterey California, April 9-12 1995, to appear.

Macedonia, Michael R., *A Network Software Architecture for Large Scale Virtual Environments*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, June 1995.

MacPherson, David Leonard Jr., *Automated Cartography by an Autonomous Mobile Robot using Ultrasonic Range Finders*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, September 1993.

Maillot, Patrick-Giles, "Using Quaternions for Coding 3D Transformations," *Graphics Gems*, Academic Press Inc., Harcourt Brace Jovanovich, Boston, 1990, pp. 498-515 and 775-777.

Maloney, Elbert S., *Dutton's Navigation and Piloting*, Naval Institute Press, Annapolis Maryland, 1985.

Marco, David B., *Slow Speed Control and Dynamic Positioning of an Autonomous Vehicle*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, March 1995.

Marks, Richard L., McLain, Timothy, Miles, David W., Rock, Stephen M., Sapilewski, Glen A., Wang, Howard H., Lee, Michael J. and Burton, Richard C., *Monterey Bay Aquarium Research Institute/Stanford Aerospace Robotics Laboratory Joint Research Program Summer Report 1992*, Monterey Bay Aquarium Research Institute, Pacific Grove California, Summer 1992.

Marks, Richard L., Lee, Michael J. and Rock, Stephen M., "Using Visual Sensing for Control of an Underwater Robotic Vehicle," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, May 3-6 1994, 13 pages.

Marks, Richard L., Rock, Stephen M. and Lee, Michael J., "Real-Time Video Mosaicking of the Ocean Floor," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 21-27.

McGhee, R.B. and Pai, A.L., "An Approach to Computer Control for Legged Vehicles," *Journal of Terramechanics*, vol. 11 no. 1, Pergamon Press, 1974, pp. 9-27.

McLeod, John and McLeod, Suzette, "A Doubly Useful and Important Infrastructure: Progress, Problems and Evaluation of Technologies Involved," *SIMULATION*, vol. 63 no. 5, November 1994, pp. 346-350.

Meyer, Kenny, "VR Gets Some Respect: SIGGRAPH '94 Presentations," *CyberEdge Journal*, issue 22, vol. 4 no. 4, July/August 1994, pp. 6 and 22.

Moravec, Hans, "The Stanford Cart and the CMU Rover," *Proceedings of the IEEE,* vol. 71 no. 7, July 1983, pp. 872-884.

Moravec, Hans, *Mind Children*, Harvard University Press, Cambridge Massachusetts, 1988, p. 48.

Morrison, John, "The VR-Link Networked Virtual Environment Software Infrastructure," *PRESENCE: Teleoperations and Virtual Environments,* to appear 1995.

Musker, A.J., Loader, P.R. and Butcher, M.C., "Simulation of a Submarine Under Waves," *International Shipbuilding Programme*, vol. 35 no. 404, 1988, pp. 389-410.

Newman, James B. and Robison, Bruce H., "Development of a Dedicated ROV for Ocean Science," *Marine Technology Society Journal*, vol. 26 no. 4, Winter 1992-1993, pp. 46-53.

Nielsen, Gregory M., Shriver, Bruce and Rosenblum, Lawrence J., editors, *Visualization in Scientific Computing*, IEEE Computer Society Press, Los Alamitos California, 1990.

Offenbeck, H.G. and Dahl, H.J., "Helicopter Simulation for Design," *Flight Simulation of Helicopters: Status and Prospects*, Royal Aeronautical Society, London, 1985.

Ong, Meng Seow, *A Mission Planning Expert System with Three-Dimensional Path Optimization for the NPS Model 2 Autonomous Underwater Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey California, June 1990.

Osterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley, Reading Massachusetts, 1994.

Pantelidis, Veronica S., "Virtual Reality and Education: Information Sources," World-Wide Web home page, *ftp://ftp.u.washington.edu/pub/user-supported/VirtualReality/misc/papers/VR-Education-Bibl.txt*, November 1994.

Pape, Dave, Molyneaux, Chuck and Stewart, Ken, "*Jason IV* Real-Time Visualization," *SIGGRAPH 93 Video Proceedings: Small Animation Theater Science Reel*, issue 93, August 1993.

Papoulias, Fotis A., Cristi, R., Marco, D. and Healey, A.J., "Modeling, Sliding Mode Control Design, and Visual Simulation of AUV Dive Plane Dynamic Response," *Sixth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, IEEE Ocean Engineering Society and University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, June 12-14 1989, pp. 536-547.

Papoulias, Fotis A., "Stability Considerations of Guidance and Control Laws for Autonomous Underwater Vehicles in the Horizontal Plane," *Seventh International Symposium on Unmanned Untethered Submersible Technology (UUST)*, University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, September 23-25 1991, pp. 140-158.

Pappas, George, Shotts, William, O'Brien, Mack and Wyman, William, "The DARPA/Navy Unmanned Undersea Vehicle Program," *Unmanned Systems*, vol. 9 no. 2, Spring 1991, pp. 24-30.

Payton, David W. and Bihari, Thomas E., "Intelligent Real-Time Control of Robotic Vehicles," *Communications of the ACM*, vol. 34 no. 8, August 1991, pp. 49-63.

Pesce, Mark and Behlendorf, Brian, moderators, "Virtual Reality Modeling Language (VRML)," working group home page, *http://www.wired.com.vrml*, 1994.

Pohl, Ira, *Object-Oriented Programming Using C++*, Benjamin/Cummings Publishing, Redwood City California, 1993.

Porter, Michael B., "Acoustic Models and Sonar Systems," *IEEE Journal of Oceanic Engineering*, vol. 18 no. 4, October 1993, pp. 425-437.

Pratt, David R., *A Software Architecture for the Construction and Management of Real Time Virtual Environments*, Ph.D. Dissertation, Naval Postgraduate School, Monterey California, June 1993.

Pratt, David R., Locke, John, Macedonia, Michael, Zeswitz, Steven, Young, Roy David and Zyda, Mike, "Ensuring World Consistency in a Shared Networked Virtual Environment," *First International Workshop on Networked Reality in Telecommunication (NR 94)*, Tokyo Japan, May 13-14 1994, section 2-3, 11 pages.

Pratt, David R., Zyda, Michael J. and Kelleher, Kristen, "1994 Annual Report for the NPSNET Research Group," Naval Postgraduate School, Monterey California, September 1994, 20 pages. Available electronically as *ftp://taurus.cs.nps.navy.mil/pub/NPSNET_MOSAIC/1994.NPSNET.Annual.Report.ps*

Press, William H., Teukolsky, Saul A., Vetterling, William T. and Flannery, Brian P., *Numerical Recipes in C*, second edition, Cambridge University Press, Cambridge England, 1992.

Raibert, Marc H., "Running with Symmetry," *International Journal of Robotics Research*, vol. 5 no. 4, Winter 1986, also appearing in *Autonomous Robot Vehicles*, Cox, I.J. and Wilfong, G.T., editors, Springer-Verlag, New York, 1990, pp. 45-61.

Ren, Guang and Balchen, Jens G., "Finite element modeling of the hydrodynamic environment of a small ROV," *Modeling, Identification and Control*, vol. 14 no. 3, Royal Norwegian Council for Scientific and Industrial Research, July 1993, pp. 145-159.

Rhyne, Theresa, Bolstad, Mark, Rheingans, Penny, Petterson, Lynn and Shackelford, Walter, "Visualizing Environmental Data at the EPA," *IEEE Computer Graphics and Applications*, vol. 13 no. 2, March 1993, pp. 34-38.

Rhyne, Theresa Marie, Treinish, Lloyd, Hibbard, Bill and Landreth, Chris, "The Process of Visualizing Environmental Data Sets," *IEEE Visualization 93*, San Jose California, October 25-29 1993, tutorial.

Rhyne, Theresa Marie, Brett, George, Brutzman, Don, Cox, Donna J. and Santos, Adelino, "Exploiting Networks for Visualization and Collaboration: No Network Roadblocks?," discussion panel, *Association for Computing Machinery (ACM) Special Interest Group on Computer Graphics (SIGGRAPH) 94*, Orlando Florida, July 24-29, 1994, pp. 481-482.

Rolfe, J.M. and Staples, K.J., editors, *Flight Simulation*, Cambridge University Press, Cambridge England, 1986.

Rosenblum, Lawrence J. and Kamgar-Parsi, Behzad, "3D Reconstruction of Small Underwater Objects using High-Resolution Sonar Data," *Proceedings of the IEEE Oceanic Engineering Society Conference AUV 92*, Washington DC, June 2-3 1992, pp. 228-235.

Rosenblum, Lawrence J., Stewart, W. Kenneth and Kamgar-Parsi, Behzad, "Undersea visualization: a tool for scientific and engineering progress," chapter 12 in *Animation and Scientific Visualization: Tools and Applications*, Earnshaw, R.A. and Watson, D., editors, Academic Press, London, 1993, pp. 205-223.

Rosenfeld, Leslie K., Schramm, Richard E., Paduan, Jennifer B., Hatcher, Gerald A. Jr. and Anderson, Todd, "Hydrographic Data Collected in Monterey Bay During 1 September 1988 to 16 December 1992," technical report 94-15, Monterey Bay Aquarium Research Institute (MBARI), Pacific Grove California, January 27 1994.

Ruth, Michael J. and Humphreys, Douglas E., "A Robust Depth and Speed Control System for a Low-Speed Undersea Vehicle," *Proceedings of the Symposium on Autonomous Underwater Vehicle (AUV) Technology 1990*, IEEE Oceanic Engineering Society, Washington DC, June 5-6 1990, pp. 51-58.

Sagatun, Svein I. and Fossen, Thor I., "The Norwegian Experimental Remotely Operated Vehicle Hydrodynamic Properties," *Seventh International Symposium on Unmanned Untethered Submersible Technology (UUST)*, University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, September 23-25 1991, pp. 10-19.

Saunders, George H., *Dynamics of Helicopter Flight*, John Wiley & Sons, New York, 1975.

Scholz, Thomas, *The State Transition Diagram with Path Priority and Its Applications*, Master’s Thesis, Naval Postgraduate School, Monterey California, September 1993.

Schulzrinne, Henning and Casner, Stephen, "RTP: A Transport Protocol for Real-Time Applications," Audio-Video Transport Working Group, Internet Engineering Task Force Draft, Oct. 20, 1993, available as *ftp://nic.ddn.mil/internet-drafts/draft-ietf-avt-rtp-04.ps*

Shank, Roger C., "Where’s the AI?" AI Magazine, vol. 12 no. 4, Winter 1991, pp. 38-49.

Shaw, Chris and Green, Mark, "The MR Toolkit Peers Package and Experiment," *IEEE Virtual Reality Annual International Symposium (VRAIS)*, Seattle Washington, September 18-22 1993, pp. 463-469.

Sheridan, Thomas B., *Telerobotics, Automation, and Human Supervisory Control*, MIT Press, Cambridge Massachusetts, 1992.

Shoemake, Ken, "Animating Rotation with Quaternion Curves," *Computer Graphics Proceedings (SIGGRAPH)*, vol. 19 no. 3, Association for Computing Machinery (ACM), New York, July 22-26 1985, pp. 245-254.

Singh, Gurminder, Serra, Luis, Png, Willie and Ng, Hern, "BrickNet: A Software Toolkit for Network-Based Virtual Worlds," *PRESENCE: Teleoperations and Virtual Environments*, vol. 3 no. 1, Winter 1994, pp. 19-34.

Smith, N.S., Crane J.W. and Summey, D.C., *SDV Simulator Hydrodynamic Coefficients*, Naval Coastal Systems Center (NCSC), Panama City Florida, June 1978 (contents declassified 31 DEC 85).

Smith, Samuel M. and Dunn, Stanley E., "The Ocean Voyager II: An AUV Designed for Coastal Oceanography," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 139-147.

Spector, Lee and Hendler, James, "The Supervenience Architecture," AAAI Fall Symposium on Sensory Aspects of Robotic Intelligence, Asilomar Conference Center, Pacific Grove, California, November 15-17, 1991, pp. 93-100.

Stallings, William, *Data and Computer Communications*, fourth edition, Macmillan Publishing, New York, 1994.

Stevens, Richard W., *UNIX® Network Programming*, PTR Prentice Hall, Englewood Cliffs New Jersey, 1990.

Stewart, W. Kenneth Jr., *Multisensor Modeling Underwater with Uncertain Information*, Ph.D. Dissertation, MIT Artificial Intelligence Laboratory Technical Report 1143, Massachusetts Institute of Technology, Cambridge Massachusetts, July 5 1988.

Stewart, W. Kenneth, "Three-Dimensional Modeling of Seafloor Backscatter from Sidescan Sonar for Autonomous Classification and Navigation," *Sixth International Symposium on Unmanned Untethered Submersible Technology (UUST)*, IEEE Ocean Engineering Society and University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, June 12-14 1989, pp. 372-392.

Stewart, W. Kenneth, "Multisensor Visualization for Underwater Archaeology," *IEEE Computer Graphics and Applications*, vol. 11 no. 2, March 1991, pp. 13-18.

Stewart, W. Kenneth, "Visualization resources and strategies for remote subsea exploration," *The Visual Computer*, Springer-Verlag, vol. 8 no. 5-6, June 1992, pp. 361-379.

Strauss, Paul S. and Carey, Rikk, "An Object-Oriented 3D Graphics Toolkit," *COMPUTER GRAPHICS*, vol. 26 no. 2, July 1992, pp. 341-349.

Stroustrup, Bjarne, *The C++ Programming Language*, second edition, Addison-Wesley Publishing, Reading Massachusetts, 1991.

Thalmann, Daniel, editor, *Scientific Visualization and Graphics Simulation*, John Wiley & Sons, Chichester Great Britain, 1990.

Thornton, Frederick Perry Boynton Jr., *A Concurrent, Object-Based Implementation for the Tactical Level of the Rational Behavior Model*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1993.

Torsiello, Kevin, *Acoustic Positioning of the NPS Autonomous Underwater Vehicle (AUV II) During Hover Conditions*, Engineers Thesis, Naval Postgraduate School, Monterey California, March 1994.

Towers, John and Hines, Jack, *Highly Dynamic Vehicles in a Real/Simulated Virtual Environment - HyDy - Equations of Motion of the DIS 2.0.3 Dead Reckoning Algorithms,* NCCOSC RDT&E, San Diego California, 7 February 1994.

Trimble, G. M., "Effecting Heuristic Constraint and Adaptive Mission Execution in a Directed-Task UUV Control Architecture," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, May 3-6 1994, 9 pages.

Trimble, G. M. and Markett, R.E., "Distribution of Guidance, Navigation and Control Functionality in the Explosive Ordnance Disposal Autonomous Underwater Vehicle Robotic Work Packages Program," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 1-6.

Tuohy, Séamus T., "A Simulation Model for AUV Navigation," *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, July 19-20 1994, pp. 470-478.

Urick, Robert J., *Principles of Underwater Sound*, third edition, McGraw-Hill, New York, 1983.

U.S. Congress Office of Technology Assessment (OTA), "Virtual Reality and Technologies for Combat Simulation," background paper OTA-BP-ISS-136, U.S. Government Printing Office, Washington DC, September 1994.

Vetter, Ronald J., Spell, Chris and Ward, Charles, "Mosaic and the World-Wide Web," *IEEE COMPUTER*, vol. 27 no. 10, October 1994, pp. 49-57.

Walsh, Don, "Undersea Satellites: The Commercialization of AUVs," *Marine Technology Society Journal*, vol. 27 no. 4, Winter 1993-1994, pp. 54-63.

Warner, David C., *Design, Simulation and Experimental Verification of a Computer Model and Enhanced Position Estimator for the NPS AUV II*, Master's Thesis, Naval Postgraduate School, Monterey California, December 1991.

Watkinson, Kenneth W., Smith, Neil S. and Henn, Douglas S., *User's Manual for TRJUUV: A Six Degree-of-Freedom Simulation Program for Unmanned Underwater Vehicles*, A.R.A.P. Report No. 635, California Research & Technology Inc., McLean Virginia, June 1989.

Wernecke, Josie, *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor™, Release 2*, Addison-Wesley Publishing, Reading Massachusetts, 1994.

Wernecke, Josie, *The Inventor Toolmaker: Extending Open Inventor™, Release 2,* Addison-Wesley Publishing, Reading Massachusetts, 1994.

Wernecke, Josie, Bassler, Eleanor, Carey, Rikk and Strauss, Paul, "Open Inventor: How to Write an Open Inventor File Translator," technical report, Silicon Graphics Inc. (SGI), Mountain View California, 1994.

Wilhelms, Jane, "Dynamics Experiences," *Making Them Move: Mechanics, Control and Animation of Articulated Figures*, Badler, Norman I., Barsky, Brian A., and Zeltzer, David A., editors, Morgan Kauffmann Publishers Inc., San Mateo California, 1991, pp. 265-279.

Wilkinson, Wilfrid P., *Mission Executor for an Autonomous Underwater Vehicle*, Master's Thesis, Naval Postgraduate School, Monterey California, September 1991.

Williams, L., "Real Time Blade Element Rotor Modeling," *Flight Simulation of Helicopters: Status and Prospects*, Royal Aeronautical Society, London, 1985.

Witkin, Andrew, Kass, Michael, Baraff, David, and Barr, Alan, *SIGGRAPH 93 Course Notes: Physically-Based Modeling*, Association for Computing Machinery (ACM), New York, August 1993.

WTEC: World Technology Evaluation Center, "Research Submersibles and Undersea Technologies in Finland, France, Russia, Ukraine, and the United Kingdom," workshop symposium report, Loyola College, Baltimore Maryland, July 29 1993.

Yoerger, Dana R. and Slotine, Jean-Jacques E., "Robust Trajectory Control of Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, vol. OE-10 no. 4, October 1985, pp. 462-470.

Yoerger, Dana R., "Precise Control of Underwater Robots: Why and How," *Proceedings of the First IARP Workshop on Mobile Robots for Subsea Environments*, Monterey California, October 23-26 1990, pp. 113-117.

Yoerger, Dana R., Bradley, Albert M. and Walden, Barrie, "The Autonomous Benthic Explorer (ABE): An AUV Optimized for Deep Seafloor Studies," *Seventh International Symposium on Unmanned Untethered Submersible Technology (UUST)*, University of New Hampshire (UNH) Marine Systems Engineering Laboratory (MSEL), Durham New Hampshire, September 23-25 1991, pp. 60-70.

Yoerger, Dana R., Bradley, Albert M. and Walden, Barrie, "System Testing of the Autonomous Benthic Explorer," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, May 3-6 1994, 11 pages.

Yuh, J., "Modeling and Control of Underwater Robotic Vehicles," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20 no. 6, November/December 1990, pp. 1475-1483.

Zehner, Stanley N., *Modeling and Simulation of a Deep Submergence Rescue Vehicle (DSRV) and Its Networked Application*, Master's Thesis, Naval Postgraduate School, Monterey California, June 1993.

Zeswitz, Steven, *NPSNET: Integration of Distributed Interactive Simulation (DIS) Protocol for Communication Architecture and Information Interchange*, Master's Thesis, Naval Postgraduate School, Monterey California, 28 May 1993.

Ziomek, L.J. and Polnicky, F. Wynn, "The RRA Algorithm: Recursive Ray Acoustics for Three-Dimensional Speeds of Sound," *IEEE Journal of Oceanic Engineering*, vol. 18 no. 1, January 1993, pp. 25-30.

Ziomek, L.J., "Sound-Pressure Level Calculations Using the RRA Algorithm for Depth-Dependent Speeds of Sound Valid at Turning Points and Focal Points," *IEEE Journal of Oceanic Engineering*, vol. 19 no. 2, April 1994, pp. 242-248.

Zorpette, Glenn, "Underwater vehicles make a splash: autopilots of the deep," *IEEE Spectrum*, vol. 31 no. 8, August 1994, pp. 38-44.

Zyda, M.J., McGhee, R.B., Kwak, S.H., Nordman, D.B., Rogers, R.C. and Marco, D., "Three-Dimensional Visualization of Mission Planning and Control for the NPS Autonomous Underwater Vehicle," *IEEE Journal of Oceanic Engineering*, vol. 15 no. 3, July 1990, pp. 217-221.

Zyda, Michael J., Monahan, James G and Pratt, David R., "NPSNET: Physically-Based Modeling Enhancements to an Object File Format," *Creating and Animating the Virtual World*, Thalmann, Nadia Magnenat and Thalmann, Daniel, editors, Springer-Verlag, Tokyo Japan, 1992, pp. 35-52.

Zyda, Michael J., "Virtual Worlds and Simulation Systems," course notes, Naval Postgraduate School, Monterey California, 1992.

Zyda, Michael J., Pratt, David R., Falby, John S., Barham, Paul T., Lombardo, Chuck and Kelleher, Kristen M., "The Software Required for the Computer Generation of Virtual Environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 2 no. 2, MIT Press, Cambridge Massachusetts, Spring 1993, pp. 130-140.

Zyda, Michael, Pratt, David, Falby, John, Barham, Paul and Kelleher, Kristen, "NPSNET and the Naval Postgraduate School Graphics and Video Laboratory," *PRESENCE: Teleoperators and Virtual Environments*, vol. 2 no. 3, MIT Press, Cambridge Massachusetts, Summer 1993, pp. 244-258.

Zyda, Michael, Pratt, David, and Macedonia, Michael, guest editors, Special Issue on Networked Virtual Reality, *PRESENCE: Teleoperators and Virtual Environments*, MIT Press, Cambridge Massachusetts, to appear 1995.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3.	Computer Technology Programs, Code CS Naval Postgraduate School Monterey, California 93943-5000	1
4.	Dr. Michael P. Bailey, Code OR/Ba Operations Research Department Naval Postgraduate School Monterey, California 93943-5000	1
5.	Don Brutzman, Code UW/Br Undersea Warfare Academic Group Naval Postgraduate School Monterey, California 93943-5000	10
6.	Dr. James Eagle, Code UW Chair, Undersea Warfare Academic Group Naval Postgraduate School Monterey, California 93943-5000	1
7.	CDR Michael J. Holden, USN Computer Science Department Naval Postgraduate School Monterey, California 93943-5000	1
8.	Dr. Anthony J. Healey, Code ME/Hy Mechanical Engineering Department Naval Postgraduate School Monterey, California 93943-5000	1

9. Dr. SeHung Kwak, Code CS/Kw 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

10. Dr. Ted Lewis, Code CS/Lw 1
Chair, Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

11. MAJ Michael R. Macedonia USA, Code CS/Ma 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

12. Dr. David Marco, Code ME/Ma 1
Mechanical Engineering Department
Naval Postgraduate School
Monterey, California 93943-5000

13. Dr. Scott McMillan, Code CS/Mc 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

14. Dr. Robert B. McGhee, Code CS/Mz 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

15. Dr. David R. Pratt, Code CS/Pr 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

16. Dr. Peter Purdue, Code OR/Pd 1
Chair, Operations Research Department
Naval Postgraduate School
Monterey, California 93943-5000

17. Dr. Man-Tak Shing, Code CS/Sh 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

18. Dr. Harrison Shull, Code 01 1
Provost/Academic Dean
Naval Postgraduate School
Monterey, California 93943-5000

19. Dr. Xiaoping Yun, Code EE/Yu 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

20. Dr. Lawrence Ziomek, Code EE/Zm 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943-5000

21. Dr. Michael J. Zyda, Code CS/Zk 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

22. Chief of Naval Research 1
800 North Quincy Street
Arlington Virginia 22217-5000

23. Commander, Naval Undersea Warfare Center Division 1
1176 Howell Street
Attn: Erik Chaum, Code 2251, Building 1171-3
Combat Systems Engineering and Analysis Laboratory (CSEAL)
Newport, Rhode Island 02841-1708

24. Dr. James Bales 1
Underwater Vehicles Laboratory, MIT Sea Grant College Program
292 Main Street
Massachusetts Institute of Technology
Cambridge Massachusetts 02142

25. Dr. Stephen G. Batsell 1
Network Research and Simulation - Code 5521
Information Technology Division
Naval Research Laboratory
Washington DC 20375-5000

26. Dr. James Bellingham 1
Underwater Vehicles Laboratory, MIT Sea Grant College Program
292 Main Street
Massachusetts Institute of Technology
Cambridge Massachusetts 02142

27. D. Richard Blidberg, Director 1
Marine Systems Engineering Laboratory, Marine Science Center
Northeastern University
East Point, Nahant, Massachusetts 01908

28. Dr. Brian S. Bourgeois, Code 7442 1
Electronics/System Engineer, Advanced Sensor and Survey
Naval Research Laboratory, Mapping Charting and Geodesy Branch
Stennis Space Center, Mississippi 39529-5004

29. MAJ Ronald B. Byrnes, Jr. 1
Software Technology Branch, Army Research Lab (ARL)
115 O'Keefe Building
Georgia Institute of Technology
Atlanta Georgia 30332-0800

30. Norman Caplan 1
National Science Foundation
4201 Wilson Boulevard
Arlington Virginia 22230

31. Rikk Carey 1
OpenInventor Design Team
Silicon Graphics Inc.
2011 N. Shoreline Boulevard
Mountain View, California 94039-7311

32. Tom Curtin 1
Office of Naval Research (ONR)
800 North Quincy Street
Arlington, Virginia 22217

33. Caroline Delthiel 1
DGA-DCN, Centre Technique des Systemes Navals
Dissuasion Lutte Sous Marine
DCN Toulon, BP 28, 83800 Toulon-Naval
France

34. Kirk Dye 1
Naval Surface Warfare Center (NSWC), Code 4310
Coastal Systems Division
Panama City, Florida 32407-5000

35. Dr. Stanley Dunn 1
Advanced Marine Systems Group, Ocean Engineering Department
Florida Atlantic University
Boca Raton, Florida 33431

36. CAPT Bruce Dyer USN 1
c/o Vail Research & Technology Corp.
4301 North Fairfax Drive
Arlington Virginia 22203

37. Professor Thor I. Fossen 1
University of Trondheim
The Norwegian Institute of Technology
Department of Engineering Cybernetics
N-7034 Trondheim, Norway

38. Bruce Gritton 1
Data Administrator, I^3LA Chair
Monterey Bay Aquarium Research Institute (MBARI)
160 Central Avenue
Pacific Grove, California 93950

39. Dr. Harold Hawkins 1
Manager, Perceptual Science
Office of Naval Research (ONR), Code 342PS
800 North Quincy Street
Arlington, Virginia 22217

40. Butler Hine 1
NASA Ames Research Center, Information Sciences
Mail Stop 269-3
Moffett Field, California 94035

41. Clinton N. Keith 1
Applied Remote Technology
9950 Scripps Lake Drive, Suite 106
San Diego California 92131

42. Dr. Yoji Kuroda 1
Institute of Industrial Science
University of Tokyo
7-22-1, Roppongi, Minato-ku, Tokyo 106 Japan

43. CAPT Chris Johnson USN 1
Advanced Research Projects Agency (ARPA)
Maritime Systems Technology Office (MSTO)
4301 North Fairfax Drive
Arlington Virginia 22203

44. Michael Lee 1
Senior Research Engineer
Monterey Bay Aquarium Research Institute (MBARI)
160 Central Avenue
Pacific Grove, California 93950

45. Dr. John Leonard 1
Underwater Vehicles Laboratory, MIT Sea Grant College Program
292 Main Street
Massachusetts Institute of Technology
Cambridge Massachusetts 02142

46. Dr. Didier Leandri 1
DGA-DCN, Centre Technique des Systemes Navals
Dissuasion Lutte Sous Marine
DCN Toulon, BP 28, 83800 Toulon-Naval
France

47. Dr. Pat Mantey 1
Computer Science and Engineering Department
University of California Santa Cruz
Santa Cruz, California 95064

48. Dr. Teresa McMullen 1
Office of Naval Research (ONR)
800 North Quincy Street
Arlington, Virginia 22217

49. Dr. Joseph J. Molitoris 1
Center for Naval Analyses
4401 Ford Avenue
Alexandria Virginia 22302

50. Chris O'Donnel, Disposal Technology Division 1
Commander, Naval Explosive Ordnance Disposal
Indian Head, Maryland 20640-5070

51. Dr. Veronica Pantelidis, Codirector 1
Virtual Reality and Education Laboratory
School of Education, East Carolina University
Greenville, North Carolina 27858-4353

52. Dr. Antonio Pasqual 1
Institut of Systems and Robotics
Instituto Superior Technico (ISR/IST)
Av. Rovisco Pais, 1
Lisbon, 1096, Codex Portugal

53. Theresa Marie Rhyne 1
Martin Marietta Technical Services
U.S. EPA Scientific Visualization Center
4201 Building - 2nd Floor, 79 Alexander Drive
Research Triangle Park, North Carolina 27709

54. Dr. Larry Rosenblum 1
Director, VR Systems and Research
Code 5580
Naval Research Laboratory
Washington DC 20375-5000

55. Dr. Alan Schultz, Code 5514 1
Naval Center for Applied Research in Artificial Intelligence (NCAR AI)
4555 Overlook Avenue S.W.
Naval Research Laboratory
Washington DC 20375-5000

56. Dr. Horst D. Simon 1
Supercomputer Systems Division
2011 North Shoreline Boulevard, Mail Stop 580
Silicon Graphics Inc.
Mountain View, California 94043-1389

57. Art Spero 1
Naval Sea Systems Command, Code PEO - Submarines
2531 Jefferson Davis Highway
Arlington, Virginia 22242-5160

58. Dr. W. Kenneth Stewart 1
Deep Submergence Laboratory
Woods Hole Oceanographic Institution (WHOI)
Woods Hole, Massachusetts 02543

59. Gary Trimble, Program Manager 1
Explosive Ordnance Disposal Robotics Work Packages (EODRWP)
Lockheed Martin Inc.
Organization 8K-01, Building 586E
1111 Lockheed Way
Sunnyvale California 94089-3504

60. Paul van Gool 1
Faculty of Aerospace Engineering
Delft University of Technology
Kluyverweg 1, 2629HS Delft
The Netherlands

61. Gianmarco Veruggio 1
Consiglio Nazionale delle Richerche
Istituto Automazione Navale
Via De Marini, 6
16149 Genova - Italia

62. Dr. Glen H. Wheless 1
Center for Coastal Physical Oceanography
Crittendon Hall, Old Dominion University
Norfolk Virginia 23529

63. Dr. Glen Williams 1
Computer Science Department
Texas A&M University
College Station, Texas 77843

64. Dr. Dana Yoerger 1
Deep Submergence Laboratory
Woods Hole Oceanographic Institution (WHOI)
Woods Hole, Massachusetts 02543

65. Dr. Junku Yuh 1
Autonomous Systems Laboratory
Department of Mechanical Engineering
University of Hawaii at Manoa
Honolulu Hawaii 96822

66. Dr. David Zeltzer 1
Sensory Communication Group
Research Laboratory of Electronics
Massachusetts Institute of Technology
50 Vassar Street, Room 36-763
Cambridge Massachusetts 02139