Virtual-Environment-Based Navigation and Control of Underwater Vehicles

Design and implementation of a virtual-reality-based testbed for navigation and control of underwater vehicles, with the ultimate goal of improved coastal ecosystem management

by D. GRACANIN, K.P. VALAVANIS,
N.C. TSOURVELOUDIS, and M. MATIJASEVIC

A remotely operated underwater vehicle (ROV), the Phantom S2, is being converted to an autonomous underwater vehicle (AUV) for shallow-water and coastal environments. As part of this project, a three-layered fuzzy-logic controller was developed and combined with a path-planning algorithm to navigate the vehicle in dynamic environments. Simulated experiments have been created in a virtual testbed to test the developed controller prior to and during real pool tests. The testbed utilizes a virtual model of the Phantom S2, which imitates the dynamic behavior of the actual vehicle in various environmental conditions. The vehicle's motion and behavior can be observed, recorded, and used to fine-tune the control variables.

Models of the environment and underwater vehicles are being developed based on the High-Level Architecture (HLA) and some existing models in the HLA Object Model Library. Using the Virtual Reality Modeling Language (VRML), a virtual-reality-based user interface is under development that will become a fully functional automated monitoring system for data collection, fusion, and interpretation from aerial photography, satellite images, and ground-truth data, and will also include a complete virtual-environment-based 3-D model of a coastal ecosystem, all contributing to ecosystem balance, preservation, and management.

In this article, after an overview of issues associated with designing and implementing an automated monitoring system, we provide justification and motivation for our research and define ecosystems and ecosystem management. We then discusses how the ROV to AUV conversion is being done by using mostly commercially available off-the-shelf hardware and software technologies. We then present the fundamentals of the three-layer fuzzy-logic navigation scheme for the underwater vehicle, as well as details pertaining to the virtual-reality-based environment modeling approach.

Automated Monitoring Systems

This research has been motivated by the long-term goal and challenge to design and implement a fully functional 3-D (autonomous) automated monitoring system for environment mapping and field visualization, and to apply enabling technologies to preserve, manage, and balance ecosystems. An integral part of the automated monitoring system is the real-time and real-life collection, fusion, and interpretation of data as frequently as possible from aerial photography, satellite images, and ground-truth data (ground-based and (shallow) underwater data) as shown in Fig. 1.

Key issues associated with the 3-D monitoring system functionality are collection, processing, and interpretation of data for monitoring habitats, uplands, wetlands, and submerged land by using satellite remote sensor data, aerial photography, and ground-truth data. Important aspects of the overall data collection, processing, and interpretation system

1070-9932/99/$10.00©1999IEEE

include temporal, spatial, radiometric resolution, atmospheric, and soil moisture conditions; effects of tidal stage on image classification; selecting appropriate classification algorithms; and training and verification samples for supervised and unsupervised classification.

All components shown in Fig. 1 comprise a comprehensive monitoring system that is the prerequisite for the development of a virtual-reality-based 3-D environment model to monitor, predict, and evaluate ecosystem changes. Satellite data provide a framework where aerial photography (a sequence of images covering the study area) is assembled. From the aerial photography, experts create a map of submerged aquatic vegetation. Ground-truth data collected through an underwater vehicle add the missing "third dimension" and improve detection and measurement of habitats. Data interpretation (including 3-D modeling and visualization) results in accurate 3-D description and characterization of the shallow-water environment and coastal ecosystem.

The automated monitoring system of Fig. 1, when fully functional, will be implemented to monitor the northern Gulf of Mexico coastal ecosystem (herein defined as the "environment" and the "problem domain"). This ecosystem stretches from the Rio Grande in Texas to Florida Bay (a distance of 2,483 km) and encompasses 3,193,799 ha of estuarine areas [1]. It is a complex community of organisms, and the environment functions as an ecological unit in nature within the tidally influenced areas spanning from the coastal wetlands to the marine ecosystem, including barrier islands, bays, bayous, estuaries, wetlands, seagrasses, and shellfish beds.

Currently, raw aerial and satellite data are available from the National Wetlands Research Center. Thus, our research efforts focus on:

- Automating the process of ground-truth data collection (both continuous and episodic to reflect closely the situation on the ground) using the Phantom S2 ROV, currently being converted into an AUV and
- Creating a 3-D simulation testbed utilizing a virtual-reality interface that may provide all necessary functionality for ecosystem management.

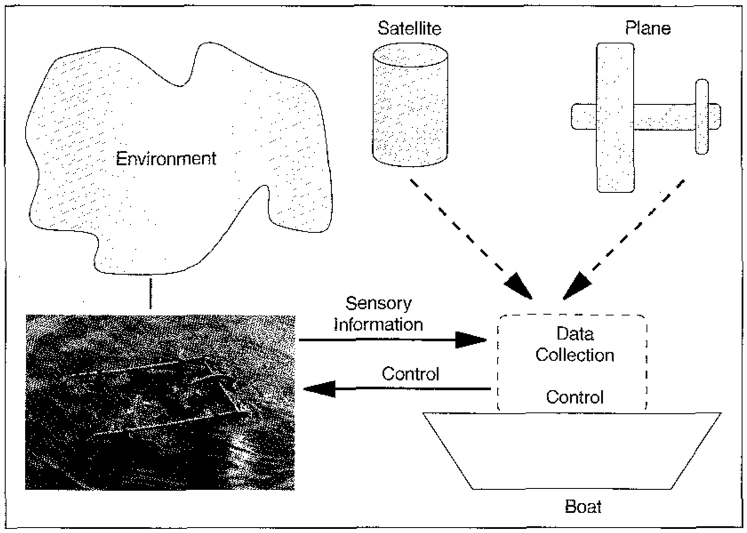


Figure 1. *Real-life data collection.*

Motivation and Current Approaches

Motivation

The reported research draws upon the recommendations made in published reports by the US Environmental Protection Agency (EPA) Environmental Monitoring and Assessment Program (EMAP) [2], the National Oceanic and Atmospheric Administration (NOAA) Coastal Change Analysis Program (C-CAP) [3], and in the Proceedings of the February 1998 IARP Workshop [4], all dictating that "our ability to preserve and restore ecosystems will be advanced most effectively through the integration of mapping, monitoring, and research across a range of spatial and temporal scales." The reports clearly identify:

- The need to gain a better understanding of a generic shallow-water coastal environment, coastal ecosystems, and their related specific problems (including oyster reefs, seagrass beds, plume edge, wetlands, etc.);
- The need for accurate mapping from aerial photographs combined with extensive concurrent ground-truthing (including submerged aquatic vegetation);
- The need to develop a nationally standardized database on land cover and habitat change in the coastal regions of the United States to monitor coastal submersed habitats, wetland habitats, and adjacent uplands and changes in these habitats.

Coastal Ecosystems and Ecosystem Management

A "coastal ecosystem" includes estuaries and coastal waters and lands located at the lower end of drainage basins, where stream and river systems meet the sea and are mixed by tides. The coastal ecosystem includes saline, brackish (mixed saline and fresh water) and freshwaters, marshes, submerged aquatic vegetation, mangroves, shellfish reefs, animals and plants, as well as coastlines and the adjacent lands, acting as integrated ecological units. Shorelands, dunes, offshore islands, barrier islands, headlands, and wetlands within estuarine drainages are included in the ecosystem definition since these interrelated features are crucial to coastal fish and wildlife and their habitats. A variety of animals and plants complete the ecological system.

Coastal ecosystems are vulnerable because of the tremendous amount of human activity that takes place along the coast, in addition to natural disturbances; e.g., storms and sea-level rise. Coastal wetlands were among the first places to be converted and developed for human activities.

Ecosystem monitoring, preservation, and restoration go beyond habitat protection because of the importance of the coastal area for commercial, industrial, and recreational uses. As such, ecosystem management refers to damage reduction maximization, based on the conflicting objectives of managing the environment for multiuse purposes, while attempting to restore or stabilize man-induced changes [1].

Therefore, it is necessary to develop specific technologies to balance coastal ecosystems and to determine the status and trends of coastal habitats in a effective way. These technologies include an automated monitoring system for measurements, data collection (remote sensing, geographic information systems, ground-truth data collected with the aid of an AUV), data interpretation, and 3-D computer models that simulate accurately the influence of environmental changes and management activities.

> Specific technologies are needed to balance coastal ecosystems and to determine the status and trends of coastal habitats in an effective way.

Current Approach to Ecosystem Management, and Limitations

Currently, data is initially derived from aerial photography and videography, satellites, radar, automated data loggers, and then verified locally ("ground verification"). This complicates and delays the overall process. In addition, small-scale changes can not be detected, and dynamic phenomena (e.g., edge of plume) cannot be monitored from aerial/satellite imagery, or moored probes. Further, since collected data is processed and stored using a geographic information system (GIS), there are two main issues limiting their use [5]:

- Coastal/marine data are time-dependent and 3-D while existing commercial GIS software were developed based on 2-D spatial topology;
- Most GIS applications involve only database generation, map production, and simple map analysis, without analytical modeling or 3-D simulation and visualization.

AUV NAVIGATION

An AUV is an unmanned, untethered, underwater vehicle that carries its own power source and relies on an on-board computer and built-in machine intelligence to execute a mission consisting of a series of preprogrammed instructions, (potentially) modifiable by data or information gathered by the vehicle sensors [6]. Unlike an ROV, an AUV operates independently, away from a support ship or platform, and does not require a human operator for navigation. The restrictions imposed on AUV design are probably the strictest of any type of robot. A consequence is that power, navigation, and mission management are the three technologies critical for the future use of AUVs. Advances in these technologies will enable AUV designers to meet the following objectives: flexible communication, efficient solution to temporal planning and resource allocation, information integration and recognition in the process of multisensor operation, planning for a given task, and adaptation to system and environment changes [7]. Navigation and mission management problems require design and development of new control architectures capable of dealing with the increasing complexity of AUV missions [8].

Control Architecture

Having reviewed the existing technology, it has been decided to use the real-time QNX operating system for software development and single-board computers (SBCs) with the STD 32 standard as a hardware platform to allow the system to be easily expandable, upgradable, modular, reliable, and cost-effective.

The STD 32 backplane design incorporates several important features, including increased backplane signal impedance. Higher backplane signal impedance means that "cleaner" signals are sent across the backplane. That is, ringing and reflections are minimized. This is especially important during signal transitions between the TTL threshold regions of 0.8 V and 2.0 V.

Figure 2 shows a typical STD-32 STAR system, available from Ziatech Corporation. It includes multiple PC-compatible CPU cards in a single card cage (backplane). Each CPU has its own memory and operating system, but shares backplane memory, disks, video, and I/O with other CPUs in the system. The sensor interface is implemented using Echelon Lonworks technology. Implementation of a real-time, sensor-based, control vehicle architecture is shown in Fig. 3.

This architecture offers the following advantages as a whole, compared to other reviewed architectures [8]:

- It utilizes the shared memory module as a communication medium between the various modules, thus eliminating other more complex means of communication between the modules. This also eliminates any overhead generated by the communication protocols that would have been used otherwise.
- It uses state diagrams to specify missions, thus making it easy to change missions on the fly, and reprogram the operation.
- The functionality of the architecture is based on software functions that run on the various modules.

Since modules operate independently, the software of a specific module can easily be modified, improved, and changed all together without affecting the functionality of the other modules. All modules can directly access the shared memory without having to go through the supervisor or an intermediate level, minimizing data transfer latency between modules and across levels.

The supervisory control component and its internal structure are shown in Fig. 4. The differences between this design and the one used by Bellingham and Consi [9] are:

- The communication between the various modules is accomplished by using shared memory variables.

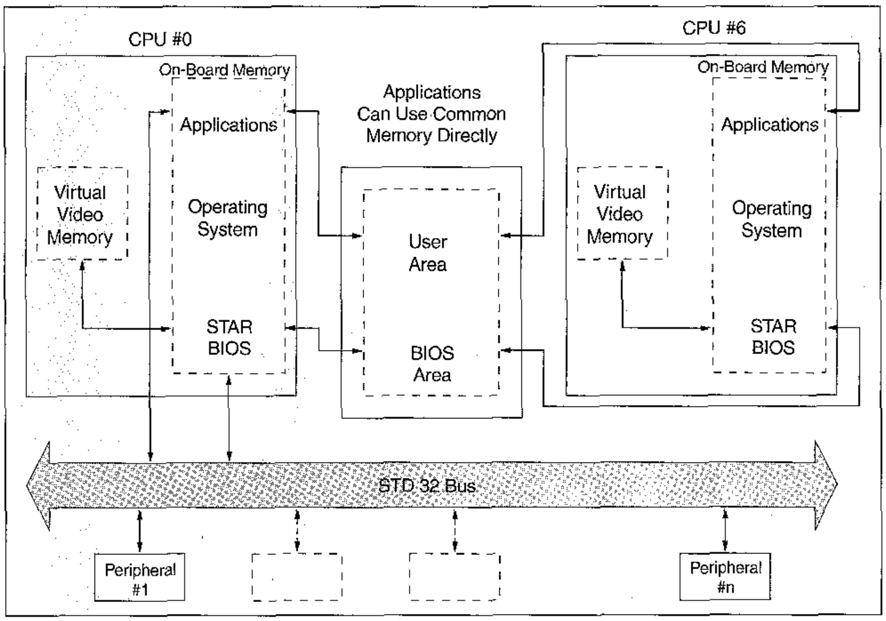


Figure 2. *STAR system architecture.*

- The overall operation of the system is coordinated by the master controller (MC).

Access to shared memory variables is controlled using semaphores and automatic logging of variables needing time stamps. Every variable is accompanied by a semaphore variable and a time-stamp variable.

Every module has its own local memory and keeps a local copy of any shared memory variable that it needs to access. Before accessing a shared memory variable, a module must set its semaphore and reset it after the variable data is transferred to the local copy of the variable. Copying the shared variables to the local memory before proceeding with the execution of the module functions, rather than manipulating the shared variables in the shared memory, minimizes the time a module accesses the shared memory, thus allowing other modules to proceed with their execution and accessing of the shared memory.

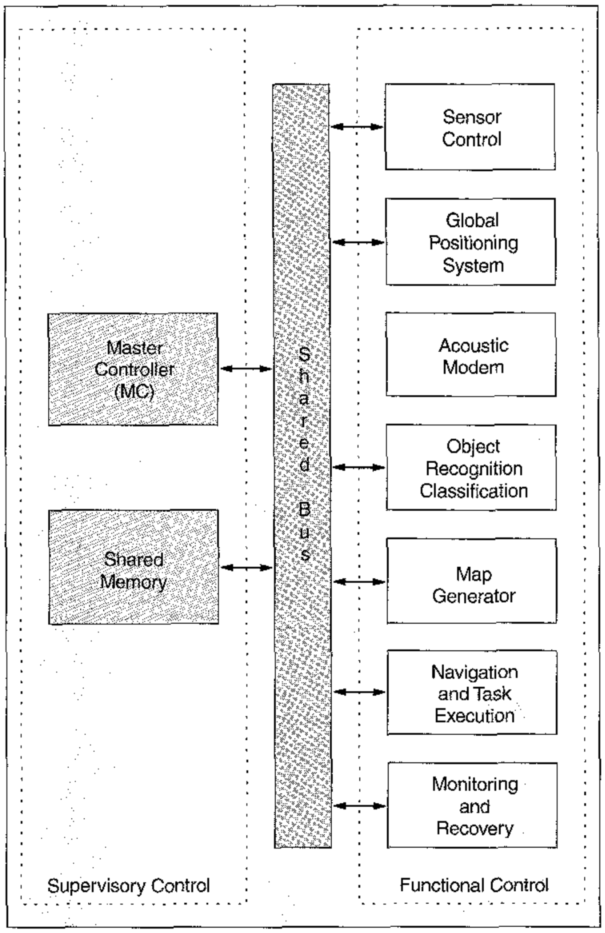


Figure 3. *Proposed control architecture block diagram.*

The three processes of the MC shown in Fig. 4 also access the shared memory. A set of flags, associated with the modules of the system, will be kept in the shared memory. Every time the state diagram reaches a new mission phase, it will set the flags associated with the modules that need to be activated. The modules periodically "look" at these flags and, according to their status, become active or will remain inactive. After a phase is completed, the MC resets all of the flags. The backup function, in the MC, backs up all information stored in the shared memory variables, after a specific phase has completed its operation. This is necessary for recovery purposes and for reviewing the overall mission performance of the AUV. The system can retrieve information and make decisions for recovery techniques based on previously stored information.

The functional component is composed of functionally independent modules that perform all pertinent functions, as shown in Fig. 3.

Fuzzy-Logic-Based Navigation

Fuzzy-logic-based navigation solutions incorporate heuristic control knowledge in the form of if-then rules and are a convenient choice when an explicit model of the system is not known [10]. They have also shown a good degree of robustness, which is crucial in the area of underwater robotics where:

- sonar data is unreliable,
- mathematical models about the environment and the vehicle are usually not available, and

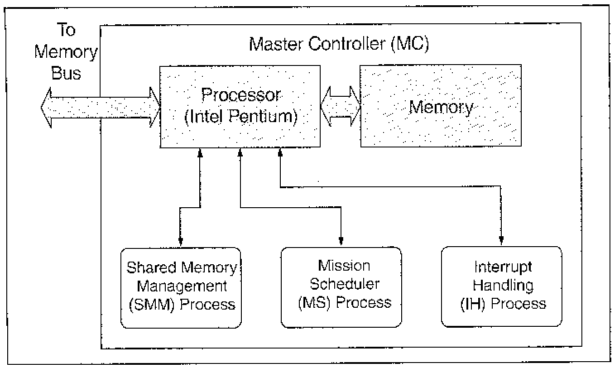


Figure 4. *Master controller block diagram.*

- the available navigation expertise is rather poor.

The overall navigation architecture is shown in Fig. 5. Three layers of fuzzy controllers are responsible for accurate and collision-free navigation. The well-known advantage of fuzzy controllers [10,11] is that they don't require any modeling of the system dynamics. Generally, a good understanding of vehicle behavior is usually enough to create a control or navigation strategy [12,13].

The first layer implements sensor fusion and it is responsible for position monitoring. The rule base of the first controller contains rules that incorporate sonar and vision information. Since the major objective is the operation of the vehicle in shallow and muddy waters, the data given by the vision system is of less importance. The controller outputs the position and the degree of possibility with which a particular collision may occur. Each sensor reading is represented as two different fuzzy variables: *sensor_direction* and *sensor_distance*. *Sensor_direction* has five different values that describe the sensor's membership in cardinal relative directions: *front_collision*, *left_collision*, *down_collision*, *right_collision*, and *up_collision*. Each of the collision values is represented as a fuzzy variable with values: *not_possible*, *possible*, and *high*. Thus, the complete local environment of the vehicle is represented in the type-2 fuzzy variable collision. Any obstacle that is close to the vehicle in any direction is represented in the collision variable. The rule base of the obstacle-detection controller contains rules of the following form:

$$\text{IF } d_i \text{ is } LD^{(k)} \text{ THEN } c_j \text{ is } LC^{(k)}$$

where, k is the rule number, d_i represents the readings of sensor i, $LD^{(k)}$ is a linguistic value of the term set $D = \{Close, Near, Far\}$, c_j is the collision of type j ($j \in \{Front, Left, Right, Up, Down\}$), and $LC^{(k)}$ is a linguistic value of the term set $C = \{Not - Possible, Possible, High\}$. The mathematical meaning of the k-th rule is given as a fuzzy relation $R_{(k)}$ on $D \times C$, which in the membership function domain is

$$\mu_{R^{(k)}}(d_i, c_j) = \min[\mu_{LD^{(k)}}(d_i), \mu_{LC^{k}}(c_j)] \,. \qquad (1)$$

The membership function of an obstacle's position and distance from the vehicle is computed by the max-min composition between the fuzzified readings and the fuzzy relation that represents the rule base, as described by Eq. (1); that is

$$\mu_C^*(c_j) = \max_{d_i}\min[\mu_D^*(d_i), \mu_{R^{(k)}}(d_i, c_j)] \,. \qquad (2)$$

This information is combined with the output of the path planner in the second layer, and if an obstacle is within the desired path a collision-avoidance decision that includes heading, depth, and speed changes is taken. The input variables of the second controller are *collision*, *heading_error*, *depth_error*, and *speed*, and generate the control variables *change-of-speed*, *depth*, and *heading*. The heading relative to the current heading of the vehicle is given as the *heading_error*. This angle represents not the angle to the goal point, but the angle in which the vehicle should point in order to follow the desired course to the goal point. Since the desired heading is given by an external course planner, the second-layer controller should implement the change- of-heading necessary to reduce the *heading_error* to zero, only changing the desired heading if collision in the vicinity of the desired path is possible or high. Thus, two kinds of navigation behaviors are implemented: goal- and reaction-directed behaviors. Goal-directed behaviors are controlled through the *heading_error depth_error* variables, incorporating any behaviors such as edge following, head to goal, etc. Reaction-directed behaviors involve possible collisions, and they are controlled through analysis of the current situation. For example:

$$\text{IF } \textit{collision} \text{ is } \textit{front possible} \text{ AND } \textit{heading\_error} \text{ is } \textit{zero}$$
$$\text{THEN } \textit{change-of-heading} \text{ is } \textit{turn\_right}.$$

Similar rules, which examine the *speed* and *depth_error* as well as all values of collision, generate the controls

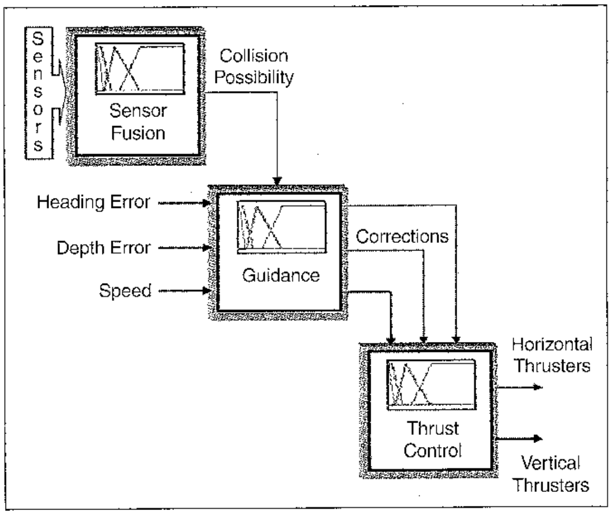
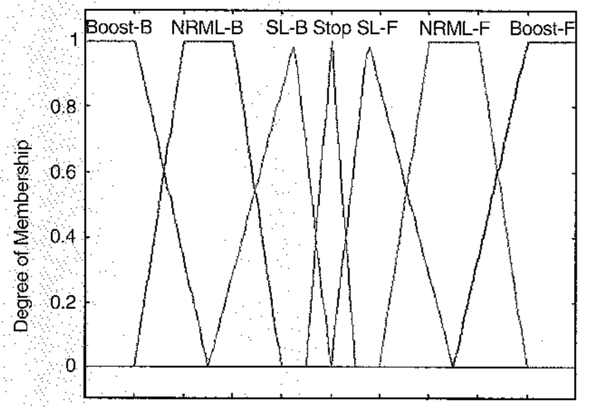
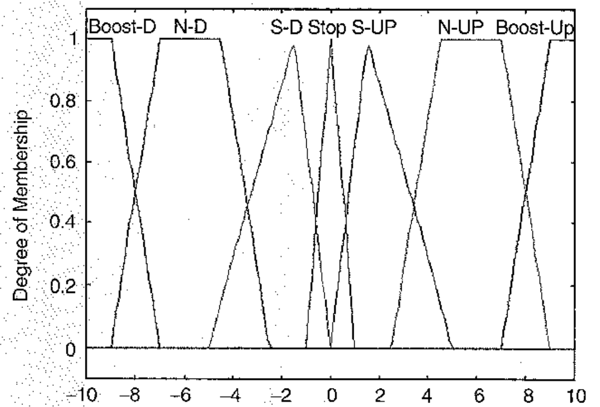


Figure 5. *Navigation architecture.*



(a) Horizontal Thruster Voltage



(b) Vertical Thruster Voltage

Figure 6. *Membership functions of the thruster controller output variables (Boost = full power; N, NRML = normal power; S, SL = low power; B = back; f = forward; UP = up; D = down).*

change-of-speed and *depth*. The rules of the second fuzzy controller can be described as follows:

$$\text{IF } c_j \text{ is } LC \text{ AND } \psi \text{ is } L\Psi \text{ AND } v \text{ is } LV \text{ AND } h \text{ is } LH$$
$$\text{THEN } d\psi \text{ is } LD\Psi \text{ AND } dh \text{ is } LDH \text{ AND } dv \text{ is } LDV$$

where, c_j is collision of type j; ψ is the heading error; v is the speed of the vehicle; $d\psi$ is the heading angle correction; dh is the change of depth; dv is the change of speed; and LC, $L\Psi$, LV, $LD\Psi$, LDH, LDV are the linguistic values of c_j, ψ, h, v, $d\psi$, dh, and dv, respectively. All variable errors are computed by continuously comparing the desired values (i.e., the values that guide the vehicle to the target point given) with the actual value of each of the variables ψ, h, and v. The heading error is important when the vehicle detects no collision on the target path. In case of possible collision, the information about the heading error ψ becomes less important, and the rules that contain such information are firing with smaller strength than the rules that perform collision avoidance (reaction-directed behavior). The generic mathematical expression of the navigation output on $N = \Psi \times H \times V$ is

$$\mu^*_N(d\psi, dh, dv) = \max \min_{c_j, \psi, v, h} [\mu^*_{AND}(c_j, \psi, h, v),$$
$$\mu_R(c_j, \psi, h, v, d\psi, dh, dv)] \quad (3)$$

where $\mu^*_{AND}(c_j, \psi, h, v)$ is the membership function of the combined effect of the 4 inputs, and $\mu_R(c_j, \psi, h, v, d\psi, dh, dv)$ is the union of all individual rule meanings.

The third layer performs low-level control by adjusting the thrusters' (two horizontal and two vertical) rotational speed (the original vehicle design does not include any fins) in order to achieve the new course. This can be done by regulating the voltage applied on the motors of the vehicle. By assuming that yaw and depth dynamics are approximately decoupled, we consider the vertical thrusters as responsible only for depth/altitude regulation, while the horizontal thrusters are responsible for heading control. In the same manner, speed in the X-Y plane is controlled by the horizontal thrusters, while the two vertical thrusters control the speed in the X-Z plane. The lateral motion of the vehicle is not considered. The rules of the third controller are of the following form:

$$\text{IF } d\psi \text{ is } LD\Psi \text{ AND } dh \text{ is } LDH \text{ AND } dv \text{ is } LDV$$
$$\text{THEN } r_t \text{ is } LR_t \text{ AND } l_t \text{ is } LL_t \text{ AND } v_t \text{ is } LV_t$$

where r_t, l_t represent the voltage applied to right and left horizontal thruster, respectively; v_t is the voltage of the vertical thrusters; and LR_t, LL_t, and LV_t represent sets of values for these variables. The membership function of the voltage values are shown in Fig. 6.

Prior to the controller's design, experimental pool tests were conducted to acquire the necessary information for the derivation of the fuzzy rule base. This information is the core of knowledge that is represented by the fuzzy rules, and it includes the vehicle's movement capability and the sensors' sensitivity and reliability.

Simulation and Visualization

The virtual environment utilizes one of the largest collections of environmental and other related data available through the Master Environment Library (MEL) Project. The MEL is a joint project involving the US DOD military services and other US Federal Government agencies. It provides a uniform interface to various data sources connecting several current and candidate resource sites in the US and in Europe (http://mel.dmso.mil/). Terrain (elevation data), hydrography, weather data, etc., are provided; they can be searched in both spatial and temporal domains. The simulation testbed follows the DOD Modeling and Simulation Master Plan and it conforms to the HLA specifications, conceptual models of the mission space (CMMS) to provide a basis for the development of consistent and authoritative simulation representations, and to data standards (DS) to support common representations of data across models, simulations, and C^{4}I systems. The testbed, when completed, will be used to simulate, measure, and evaluate the coastal ecosystem and its parameters.

Standards

Emerging standards for distributed simulation and virtual reality are used to implement a distributed simulation testbed with the virtual-reality-based interface. The two core standards used (HLA and VRML) are described below.

> AUV-related research has become one of the most important research topics in underwater robotics.

HLA facilitates reuse of simulations and their components by specifying the general structure of the interfaces between simulations without making specific demands on the implementation of each simulation. The whole distributed application (exercise, set of simulations) is called a federation. The federation consists of individual simulations and interfaces (federates). Federates share the same federation object model, which is a description of all shared information (objects, attributes, associations, and interactions) essential to a particular federation. In addition, a particular simulation (federate) uses a simulation object model to describe objects, attributes, and interactions that can be used externally in a federation. All object models are based on a common framework specified by the object model template. The basic concepts include:

- Run-time infrastructure (RTI): Implementation of a distributed operating system as a base software layer that manages communication between all simulation models;
- Rules: A set of 10 technical rules with which an HLA participant has to comply;
- Object model templates: Standardized formats used to specify the functionality of simulation models and interaction among them before the actual simulation takes place.

The immense popularity that the Internet instantly gained after introduction of the World Wide Web [14] and the market demand for Web content expanded from initial hypertext and images to audio, video, and 3-D graphics. The idea of expanding the 2-D Web to 3-D "worlds" resulted in the VRML standard [15]. VRML has introduced 3-D graphical visualization and interactivity to the Internet and the WWW. The VRML Consortium initiated the forming of technical committees, called working groups (WGs), that research and propose solutions to specific technical problems related to VRML. These enhancements may be included in the future versions of VRML. The current international standard for VRML, ISO/IEC 14772-1:1997, VRML97, provides means for creating static and animated 3-D objects/worlds, user interaction, animation, scripting, and prototyping.

Testbed

The development of the virtual-environment testbed at the Virtual Reality and Multimedia Laboratory of the University of Southwestern Louisiana started in 1995. The objective was to create an underwater virtual world to be used for testing of methods and algorithms developed as a part of the project efforts. Issues involved in a design of such a system include:

- designing 3-D models of the environment, AUVs, and other objects;
- integrating designed models in a single simulation;
- establishing communication among simulation, AUVs, and users;
- synchronizing the real AUVs with the simulation.

Figure 7 shows the structure of the testbed. Based on the HLA RTI interface, a set of federates (simulations) is being developed. The user interface is based on the VRML and provides necessary visualization functions.

The core of the simulation part of the testbed is the coastal ecosystem simulation. The user interface is virtual-reality-based and structured as a virtual environment where the user is able to explore the ecosystem model and visualize assessment, real-life, and simulation data. Possible approaches for advanced visualization of coastal-related data in the virtual environment will be evaluated and compared to provide for the effective user interface.

The use of VRML as a standard for specification and presentation of the virtual environment facilitates presentation of data on the WWW. The interface enables the user to navigate in the "virtual ecosystem," explore individual components, and modify the state of the system. Resulting effects are displayed accordingly.

Mathematical Model for AUV Navigation

In order to provide a mathematical model for the AUV's navigation, the approach from [7] and [16] is used. A moving coordinate reference frame $X_0Y_0Z_0$, called the *body-fixed* reference frame, is defined, with origin in the vehicle's center

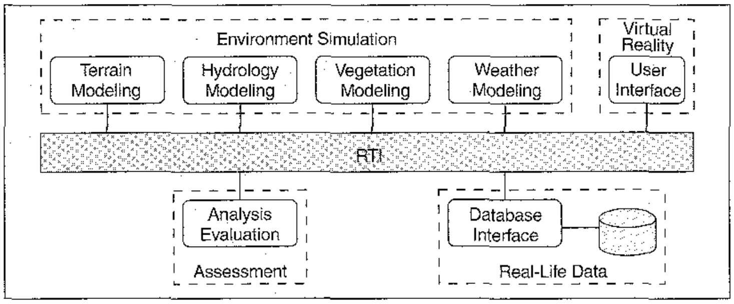
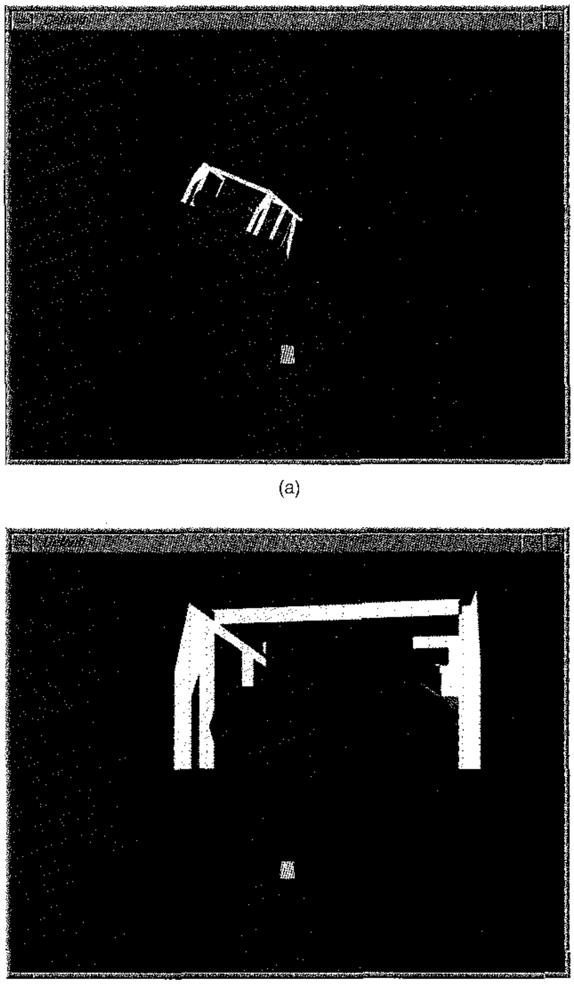


Figure 7. *Virtual-reality testbed structure.*



(a)

(b)

Figure 8. *Starting and ending point of navigation.*

of gravity and with X_0, Y_0, Z_0 serving as the AUV's principal axes of inertia. A fixed/world coordinate frame XYZ, called the *earth-fixed* reference frame, is defined. The AUV's position and orientation in the 3-D space is determined as a function of six degrees-of-freedom (DOFs).

The first three degrees-of-freedom (DOFs) (surge, sway, heave) and their time derivatives determine the vehicle's position and translational motion along the x-, y-, z- axes, while the last three DOFs (roll, pitch, yaw) and their time derivatives determine the vehicle's orientation and rotational motion. The vector $\vec{\eta} = (x\ y\ z\ \phi\theta\psi)^T$ defines the AUV's position and orientation with the respect to the earth-fixed reference frame; the vector $\vec{v} = (u\ v\ w\ p\ q\ r)^T$ defines the AUV's linear and angular velocity with respect to the body-fixed reference frame; and the vector $\vec{\tau} = (X\ Y\ Z\ K\ M\ N)^T$ defines the forces and torques with respect to the body reference frame.

The body-fixed frame, 6-DOF nonlinear dynamic equations of motion are represented in compact form (including vehicle thruster forces, hydrodynamics damping, and lift and restoring forces) as follows:

$$\vec{\tau} = \vec{M}\dot{\vec{v}} + \vec{C}(\vec{v})\vec{v} + \vec{D}(\vec{v})v + \vec{g}(\vec{\eta}) \tag{4}$$

where $\bar{\tau}$ is the vector of control inputs, $\vec{M}$ is the inertia matrix (with added mass), $\vec{C}(\vec{v})$ is the matrix of Coriolis and centripetal terms (with added mass), $\vec{D}(\vec{v})$ is the damping matrix, and $\vec{g}(\vec{\eta})$ is the vector of gravitational forces and moments. The earth-fixed vector representation is:

$$\vec{\tau}_{\eta} = \vec{M}_{\eta}(\vec{\eta})\ddot{\vec{\eta}} + \vec{C}_{\eta}(\vec{v}, \vec{\eta})\dot{\vec{\eta}} + \vec{D}_{\eta}(\vec{v}, \vec{\eta})\dot{\vec{\eta}} + \vec{g}_{\eta}(\vec{\eta}) \tag{5}$$

AUV Navigation in the Virtual Environment

The AUV kinematics and dynamics equations of motion provide a framework within which a path-planning or a navigation algorithm can be implemented in the virtual-environment testbed. An algorithm based on Eq. (5) consists of the following steps:

- Determine the AUV's environment including terrain and obstacles.
- Set a goal, or a location to be reached (in the environment).
- Produce a path.
- Follow the path by applying thruster controls of the virtual AUV and using Eq. (5).

The environment can be determined in two ways: either by using the database module and the virtual environment's geometry data, or by using virtual sensors to probe the virtual environment and construct the map of the environment. The first approach provides complete knowledge of the environment, something that is never the case in the actual mission. The second approach uses virtual sensors (sonar) to create an incomplete map of the environment. The first approach is used during the path-planning algorithm development phase, while the second approach is used during the verification and validation of the developed algorithm.

The goal or a specific location in the environment can be set either manually, or by some higher level planning algorithm. The automatic navigation of the AUV can be implemented using an algorithm such as the fuzzy-logic-based navigation described in the previous section to produce a path between the starting point and the ending point (Fig. 8).

Although Eq. (5) has four components of the vector of control inputs $\vec{\tau}_{\eta}$, only the inertia matrix $\vec{M}$ and the vector of gravitational forces and moments $\vec{g}(\vec{\eta})$ are utilized at this point.

Conclusion

AUV-related research has become one of the most important research topics in underwater robotics. Although there are many research results, the number of completed AUVs is small and their capabilities are limited. A costly, difficult, and potentially hazardous development process is one of the main reasons for this. The ability to design and test control algorithms without extensive real-life testing is very attractive, and a virtual-environment-based testbed makes it feasible. The virtual-environment testbed currently under development at the Virtual Reality and Multimedia Laboratory of the University of Southwestern Louisiana uses some new trends in simulation and virtual reality to provide a portable and reusable tool for development and evaluation of control algorithms for AUVs. It is envisioned to be a functional automated monitoring system with a clearly defined approach for data collection; fusion; and interpretation from aerial photography, satellite images, and ground-truth data. The expected contributions include an operational underwater vehicle, design specifications for a new dedicated underwater vehicle, and a complete virtual-environment-based 3-D model of a coastal ecosystem, all contributing to ecosystem balance, preservation, and management.

Acknowledgments

This work has been partially supported by NSF Grants BES-9506771 and BES-9712565.

Keywords

Remotely operated vehicle, autonomous underwater vehicle, fuzzy logic, virtual reality

References

[1] R.H. Kilgen and R. J. Dugas, "The ecology of oyster reefs of the northern Gulf of Mexico," NWRC Open File Report 89-02, National Wetlands Research Center, U.S. Fish and Wildlife Service, Sept. 1989.

[2] H.A. Neckles, ed., "Indicator development: Seagrass monitoring and research in the Gulf of Mexico," Tech. Rep. EPA/620/R-94/029, U.S. Environmental Protection Agency, Dec. 1994.

[3] J.E. Dobson, E.A. Bright, R.L. Ferguson, D.W. Field, L.L. Wood, K.D. Hadda, H. Iredale III, J.R. Jensen, V.V. Klemas, R.J. Orth, and J.P. Thomas, "NOAA coastwatch change analysis project - guidance for regional implementation - version 1.0," Tech. Rep., National Oceanic and Atmospheric Administration, Jan. 20 1994.

[4] K.P. Valavanis, D. Gracanin, M. Matijasevic, and R. Kolluru, eds., *Proceedings of the IARP First International Workshop on: Autonomous Underwater Vehicles for Shallow Water and Coastal Environments* (Lafayette, LA), University of Southwestern Louisiana, Feb. 17-19 1998.

[5] W. Ji and J. B. Johnston, "Coastal ecosystem decision-support GIS: Functions and methodology," *Marine Geodesy*, vol. 18, pp. 229-241, 1995.

[6] S.W. Kandebo, *AW&ST and AUVSI 1997-98 International Guide to Unmanned Vehicles*. New York: McGraw-Hill, 1997.

[7] J. Yuh, ed., Underwater Robotic Vehicles: Design and Control. Albuquerque, New Mexico: TSI Press, 1995.

[8] K.P. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G.A. Demetriou, "Control architectures for autonomous underwater vehicles," *IEEE Control Systems Magazine*, vol. 17, no. 6, pp. 48-64, Dec. 1997.

[9] J.G. Bellingham and T.R. Consi, "State configured layered control," in *Proceedings of the IARP 1st Workshop on Mobile Robots for Subsea Environments* (Monterey, California), pp. 75-80, Oct. 23-26 1990.

[10] N. Kato, "Applications of fuzzy algorithm to guidance and control of underwater vehicles," in *Underwater Robotic Vehicles: Design and Control* (J. Yuh, ed.), ch. 6, pp. 145-183, Albuquerque, NM: TSI Press, 1995.

[11] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*. Berlin: Springer (second, revised ed.), 1996.

[12] P.A. DeBitteto, "Fuzzy logic for depth control of unmanned undersea vehicles," *IEEE J. Oceanic Eng.*, vol. 20, no. 3, pp. 242-248, 1995.

[13] Z. Vukic, E. Omerdic, and L. Kuljaca, "Fuzzy autopilot for ships experiencing shallow water effect in manouvering," in *Proc. 4th IFAC Conference on Manouvering and Control of Marine Craft* (Brijuni, Croatia), pp. 69-74, 10-12 Sept. 1997.

[14] T. Berners-Lee, R. Callieu, A. Luotonen, H.F. Nielsen, and A. Secret, "The World-Wide Web," *Communications of the ACM*, vol. 37, pp. 76-82, Aug. 1994.

[15] —, *Information Technology - Computer graphics and image processing - The Virtual Reality Modeling Language (VRML) - Part 1: Functional specification and UTF-8 encoding. ISO/IEC 14772-1*. International Standards Organization, 1997.

[16] A.J. Healey and D.P. Brutzman, "Underwater robotics workshop," July 6, 1997. 8th International Conference on Advanced Robotics, Monterey, CA.

Denis Gracanin received the B.Sc. and M.Sc. in electrical engineering from the University of Zagreb, Croatia, in 1985 and 1988, respectively. He also received the M.Sc. and Ph.D. degrees in computer science from the University of Southwestern Louisiana, Lafayette, LA, in 1992 and 1994, respectively. Between 1985 and 1991 he was a lecturer assistant at the Telecommunications Department of the University of Zagreb. He is currently a research scientist at the A-CIM Center and an adjunct assistant professor at the Center for Advanced Computer Studies (CACS) at the University of Southwestern Louisiana, LA. His research interests are in the fields of virtual reality, intelligent robotic systems, Petri nets, and automated manufacturing systems. Dr. Gracanin is a member of IEEE, ACM, AAAI, APS, and SIAM.

Kimon Valavanis is a professor of computer engineering at the University of Southwestern Louisiana Center for Advanced Computer Studies and the director of the Robotics and Automation Laboratory, and he is also the A-CIM/BORSF Regents Professor of Manufacturing. His research interests are in the areas of robotics, automation, and distributed intelligence systems. He has published more than 180 transactions, journal, and conference papers; book chapters; and technical reports. He has been the general chair (GC) and program chair (PC) of IEEE conferences and symposia and has served on several committees in different capacities, including the GC (with Z. Vukic) of the 2001 IEEE Mediterranean Conference in Control and Automation. In June 1998 he was elected Vice President for Administration of the Mediterranean Control Association (MCA). He is a Senior Member of IEEE.

Nikos C. Tsourveloudis received the diploma and Ph.D. degrees in production engineering and management from the Technical University of Crete, Chania, Greece, in 1990 and 1995, respectively. From 1992 to 1995, he was the president of the Hellenic Association of Industrial Engineers. From 1996 to 1998, he was a visiting research scientist at the Robotics and Automation Laboratory, A-CIM Center, University of Southwestern Louisiana, Lafayette, LA, USA. Currently he is an adjunct professor at the Department of Production Engineering and Management of the Technical University of Crete. His research interests lie in the areas of fuzzy control, modeling of flexible manufacturing systems, planning, scheduling, integration of production networks, and autonomous operation/navigation of unmanned vehicles. Dr. Tsourveloudis received the Best Student Paper Award of the manufacturing division of the 3rd World Automation Congress, held in Anchorage, Alaska, in 1998.

Maja Matijasevic received the B.S. (1990), M.S. (1994), and Ph.D. (1998) degrees in electrical engineering from the University of Zagreb, Croatia, and the M.S. degree in computer engineering (1997) from the University of Southwestern Louisiana, Lafayette, LA, USA. Her Ph.D. thesis was in the area of distributed multiuser multimedia and virtual-reality applications. Since 1991 she has been affiliated with the Telecommunications Department of the Faculty of Electrical Engineering and Computing at University of Zagreb, where she currently holds a research and lecturing assistant position. From 1996-1998 she worked as a research associate at the Center for Advanced Computer Studies of the University of Southwestern Louisiana, where she completed most of her Ph.D. thesis-related work. Her main research interests include computer and communication networks, multimedia, and virtual reality. Ms. Matijasevic is a member of IEEE and Upsilon Pi Epsilon.

Address for Correspondence: Kimon Valavanis, Center for Advanced Computer Studies, Robotics and Automation Lab, 2 Rex Street, CC 4th Floor, Lafayette, LA 70504-4330 USA. E-mail: kimon@cacs.usl.edu

---

Robust Nonlinear Motion Control for AUVs

(continued from page 38)

been a lecturer and associate professor at the University of Genova, and, since 1990, he has been Professor of Systems Theory at the University of Ancona, Italy. Professor Conte was a Fulbright Scholar in 1980 and 1987. He was Visiting Professor at the Ohio State University in Columbus, Ohio, USA, in 1987 and at the Ecole Normale Superieure de Mecanique de Nantes, France in 1988. His research interests concern algebraic methods in systems and control theory and robotics. Professor Conte is a member of IEEE, AMS, and UMI.

Andrea Serrani graduated with a degree in electrical engineering from the University of Ancona, Italy, in 1993. He received an M.S. in systems science and mathematics from Washington University in St. Louis in 1995, and a Ph.D. in artificial intelligence systems from the University of Ancona in 1997. Since 1994 he has been a Fulbright Student at the Department of Systems Science and Mathematics at Washington University in St. Louis, where he is currently working toward his D.Sc. degree. His research interests include nonlinear control theory and underwater robotics.

Address for Correspondence: Andrea Serrani, Department of Systems Science and Mathematics, Campus Box 1040, Washington University, One Brookings Drive, St. Louis, MO, 63130 USA. E-mail: andrea@zach.wustl.edu.